Parallel Split-Join Networks for Shared-account Cross-domain Sequential Recommendations

Ren, P.; Lin, Y.; Ma, M.; Chen, Zhumin; Ren, Z.; Ma, J.; de Rijke, M.

Publication date
2019

Document Version
Submitted manuscript

License
Unspecified

Parallel Segregation-Integration Networks for Shared-account Cross-domain Sequential Recommendations*

PENGJIE REN, Informatics Institute, University of Amsterdam, The Netherlands
YUJIE LIN, School of Computer Science and Technology, Shandong University, China
MUYANG MA, School of Computer Science and Technology, Shandong University, China
ZHUMIN CHEN, School of Computer Science and Technology, Shandong University, China
ZHAOCHUN REN, School of Computer Science and Technology, Shandong University, China
JUN MA, School of Computer Science and Technology, Shandong University, China
MAARTEN DE RIJKE, Informatics Institute, University of Amsterdam, The Netherlands

Sequential Recommendation (SR) has been attracting a growing attention for the superiority in modeling sequential information of user behaviors. We study SR in a particularly challenging context, in which multiple individual users share a single account (shared-account) and in which user behaviors are available in multiple domains (cross-domain). These characteristics bring new challenges on top of those of the traditional SR task. On the one hand, we need to identify the behaviors by different user roles under the same account in order to recommend the right item to the right user role at the right time. On the other hand, we need to discriminate the behaviors from one domain that might be helpful to improve recommendations in the other domains.
In this work, we formulate *Shared-account Cross-domain Sequential Recommendation* (SCSR) and propose a parallel modeling network to address the two challenges above, namely **P**arallel **S**egregation-**I**ntegration **Net**work (ψ-Net). "Segregation" is used to address the challenge raised by shared accounts; it learns role-specific representations, and uses a gating mechanism to filter out information of some user roles that might be useful for another domain from the mixed user behaviors. "Integration" is used to address the challenge raised by the cross-domain setting; it learns cross-domain representations by combining the information from "Segregation" and then transforms it to another domain. We present two variants of ψ-Net, ψ-Net-I and ψ-Net-II. ψ-Net-I is a "Segregation-by-Integration" framework where it segregates to get role-specific

*This paper is a substantially extended version of [44]. The additions are three-fold. First, we unify the parallel modeling framework introduced in [44] into the ψ-Net in this paper and propose a new model ψ-Net-II which further improves the performance (π-Net in the original paper corresponds to ψ-Net-I in this paper). Second, we build a new dataset for Shared-account Cross-domain Sequential Recommendations (SCSRs) by simulating the shared-account characteristic on a public dataset. Third, we carry out more experiments to test ψ-Net-I and ψ-Net-II. More than half of the experiments reported in the paper were not in Ma et al. [44] and all relevant result tables and figures are either new additions to the article or report new results.

Authors' addresses: Pengjie Ren, Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands, p.ren@uva.nl; Yujie Lin, School of Computer Science and Technology, Shandong University, Jinan, China, yu.jie.lin@outlook.com; Muyang Ma, School of Computer Science and Technology, Shandong University, Jinan, China, muyang0331@gmail.com; Zhumin Chen, School of Computer Science and Technology, Shandong University, Jinan, China, chenzhumin@sdu.edu.cn; Zhaochun Ren, School of Computer Science and Technology, Shandong University, Jinan, China, renzhaochun@sdu.edu.cn; Jun Ma, School of Computer Science and Technology, Shandong University, Jinan, China, majun@sdu.edu.cn; Maarten de Rijke, Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands, derijke@uva.nl.

1046-8188/2019/1-ART $15.00
https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx

representations and integrates to get cross-domain representations at each timestamp simultaneously. ψ-Net-II is a "Segregation-and-Integration" framework where it first segregates role-specific representations at each timestamp, and then the representations from all timestamps and all roles are integrated to get cross-domain representations. We concatenate the in-domain and cross-domain representations to compute the recommendation score for each item. Both ψ-Net-I and ψ-Net-II can simultaneously generate recommendations for two domains where user behaviors on two domains are synchronously shared at each timestamp.

We use two datasets to assess the effectiveness of ψ-Net. The first dataset is a simulated SCSR dataset obtained by randomly merging the Amazon logs from different users in movie and book domains. The second dataset is a real-world SCSR dataset built from smart TV watching logs of a commercial company. Our experimental results demonstrate that ψ-Net outperforms state-of-the-art baselines in terms of MRR and Recall.

CCS Concepts: • **Information systems** → **Recommender systems**.

Additional Key Words and Phrases: Parallel modeling, Shared-account recommendation, Cross-domain recommendation, Sequential recommendation

ACM Reference Format:
Pengjie Ren, Yujie Lin, Muyang Ma, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. Parallel Segregation-Integration Networks for Shared-account Cross-domain Sequential Recommendations. *Submitted to ACM Transactions on Information Systems* 1, 1 (January 2019), 28 pages. https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx

1 INTRODUCTION

Traditional recommendation methods, e.g., Collaborative Filtering (CF) based methods [55] or Matrix Factorization (MF) based models [35], are hard to apply in some recommendation scenarios due to the absence of the user profiles (e.g., user-ids). This happens when the users are not logged in or the websites do not track the user-ids at all. For this reason, *Sequential Recommendation* (SR) is proposed and attracting more and more attention recently. Compared with traditional recommendations, Sequential Recommendation (SR) also has natural advantages when it comes to sequential dynamics [21], i.e., generating different recommendation lists at different timestamps. The goal of SR is to promote recommendations based on a sequence of user behaviors (e.g., a sequence of listened musics, watched videos or clicked products), where interactions are usually grouped by virtue of taking place within the same time frame [12, 38, 40, 52]. Users usually have specific goals during the process, such as finding a good restaurant in a city, or listening to music of a certain style or mood [51]. SRs have a wide range of applications in many domains such as e-commerce, job websites, music and video recommendations [56].

Early studies for SR are mostly based on Markov Chains (MC) [77] or Markov Decision Processes (MDP) [56] where they consider the sequences of items as states and try to learn a state-transition matrix or function to promote recommendations. In this way, the dynamic characteristics of SR are taken into consideration. However, because the states in MC or MDP based methods correspond to sequences of items, the state-transition matrix or function quickly becomes unmanageable in realistic scenarios [50]. Recurrent Neural Networks (RNNs) have demonstrated their effectiveness in sequence modeling in the field of natural language processing. Motivated by this, recent studies introduce RNNs into SR [25] and now RNN-based models are in the majority in the context of SR. Researchers have proposed various RNN architectures to enhance SR from different perspectives, e.g., context-aware SR [41], personalized SR [51], repeated SR [52], etc. However, they all focus on a single domain and none simultaneously considers the shared account and cross-domain characteristics.

In this paper, we study SR in a particularly challenging context, *Shared-account Cross-domain Sequential Recommendation* (SCSR), in which multiple individual users share a single account

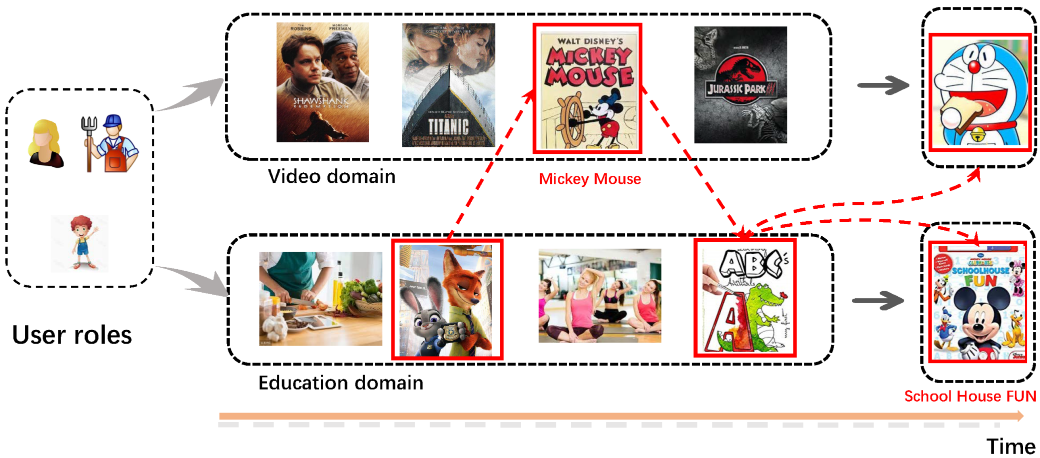

Fig. 1. The smart TV scenario provides a natural example of *Shared-account Cross-domain Sequential Recommendation* (SCSR). Here, the *video* domain contains various movies, TV series, cartoons, etc. The *education* domain contains educational programs and technical tutorials, etc. *Boxed* items reflect similar user interests. Red lines show the interactions and connections between the user behaviors in the two domains.

(shared account) and in which user behavior is recorded in multiple domains (cross-domain). The shared account characteristic is considered because in some applications, people tend to share a single account. For example, in the smart TV recommendation scenario depicted in Figure 1, members of a family share a single account to watch videos. The existence of shared accounts makes it harder to generate relevant recommendations, because multiple user behaviors are mixed together. We consider the cross-domain task because it is a common phenomenon in practice. Users use different platforms to access domain-specific services in daily life. We can get user behavior data from different domains during the same time period. For example, many smart TV platforms use different channels to provide different services, e.g., video channel (domain) which offers movie or TV series watching service and education channel which offers adult or children education materials, as depicted in Figure 1. User behaviors in one domain might be helpful for improving recommendations in another domain [15, 27, 28, 57, 66], the idea being that user behavior in different domains might reflect similar user interests. For example, as shown in Figure 1, videos like "Mickey Mouse" in the video domain might help to predict the next item "School House Fun" in the education domain because they reflect the same interest in Disney cartoon character "Mickey Mouse" presumably by a child in this family. Although leveraging user behavior information from another domain will incorporate useful information which could probably improve recommendation performances, it is nontrivial because user behaviors are mixed and this might introduce noisy information at the same time. This raises another challenge, namely how to identify behavior from one domain that might be helpful to improve recommendations in the other domains while minimizing the impact of noisy information.

In prior work that focuses on shared accounts, a common approach is to capture user preferences by extracting latent features from high-dimensional spaces that describe the relationships among users under the same account [2, 3, 13, 63, 67, 74]. And in prior work on the cross-domain task, one common solution is to aggregate information from two domains [15, 24, 27, 66], while another is to transfer knowledge from the source domain to target domain [13, 76]. None of these methods can be directly applied to SCSR for at least two reasons. Either important sequential characteristics of SR are largely ignored or they rely on explicit user ratings, which are usually unavailable in SR.

To address the above issues, we propose a novel parallel modeling scheme for SCSR, namely **P**arallel **S**egregation-**I**ntegration **Net**work (ψ-Net). The cores of ψ-Net are "segregation" and "integration" which are achieved in parallel modeling settings. To address shared-account, "Segregation" is used to identify different user behaviors where we employ a gating mechanism to extract role-specific representations containing information of some user roles that might be useful for another domain from the mixed user behaviors. To address cross-domain, "Integration" is used to discriminate and combine useful user behaviors where we learn cross-domain representations by combining the information from "Segregation" and then adopt it to another domain. Specifically, ψ-Net is organized in four main modules, namely a *sequence encoder*, a *segregation unit*, an *integration unit* and a *hybrid recommendation decoder*. The *sequence encoder* module encodes the current sequence of mixed user behaviors from each domain into a sequence of in-domain representations. Then, depending on how "segregation" and "integration" are implemented, we propose two ψ-Net variants, i.e., ψ-Net-I and ψ-Net-II. ψ-Net-I employs a "Segregation-by-Integration" scheme where it segregates to get role-specific representations and integrates to get cross-domain representations at each timestamp simultaneously. ψ-Net-II employs a "Segregation-and-Integration" scheme where it first segregates role-specific representations at each timestamp, and then the representations from all timestamps and all roles are integrated to get cross-domain representations. For both variants, "segregation" and "integration" are operated in a parallel recurrent way, which means that they can synchronously share information across both domains at each timestamp. Finally, the *hybrid recommendation decoder* module estimates the recommendation scores for each item based on the information from both domains, i.e., the in-domain representations from the target domain and the cross-domain representations from the complementary domain. During learning, ψ-Net is jointly trained on two domains in an end-to-end fashion.

To assess the effectiveness of ψ-Net, we need datasets that exhibit both share-account and cross-domain characteristics. To the best of our knowledge, there is no such real-world dataset that is publicly available. we construct two datasets for SCSR. The first dataset is a simulated SCSR dataset by randomly merging the Amazon logs from different users in movie and book domains.[1] Although the dataset can satisfy experimental requirements, the merged user behaviors are unnatural and cannot completely reflect realistic scenarios. To this end, we build the second dataset from smart TV watching logs of a commercial company which is a real-world SCSR scenario. We release both datasets to facilitate future research. We carry out extensive experiments on both datasets. The experimental results show that ψ-Net outperforms state-of-the-art baselines in terms of MRR and Recall. We also conduct ablation studies to show that the proposed parallel "segregation" and "integration" schemes are effective and useful for SCSR.

Our contributions can be summarized as follows:

- We introduce the task of *Shared-account Cross-domain Sequential Recommendation* (SCSR), which has little attention in existing studies. We release two shared account, smart TV recommendation datasets to facilitate future research in this space.
- We present a novel ψ-Net framework which introduces the parallel modeling scheme for SCSR and simultaneously yields recommendations for two domains.
- We propose two ψ-Net variants which address share-account and cross-domain with different "segregation" and "integration" strategies.
- We conduct extensive experiments and ablation studies to show the effectiveness of two ψ-Net variants.

[1]http://jmcauley.ucsd.edu/data/amazon/

2 RELATED WORK

We consider three types of related work: sequential, shared account, and cross-domain recommendation.

2.1 Sequential recommendations

The classical recommendation methods, e.g., MF or CF based methods, are hard to capture the sequential dynamics, which leads to SR or next basket recommendation.

Traditional methods. The traditional approaches for SR are mostly based on Markov Chains (MC) [77] or Markov Decision Processes (MDP) [56] to predict users' next action given the last action [65]. Zimdars et al. [77] are the first to propose MC for web page recommendation. They investigate how to extract sequential patterns to learn the next state using probabilistic decision-tree models. To further improve the performance, Shani et al. [56] propose a MDP based recommendation method, where the next recommendation can be computed through the transition probabilities among items. To combine the advantages of matrix factorization and MC based methods, Rendle et al. [54] propose a method based on personalized transition graphs over underlying MC. They show that the proposed method subsumes both a common MC and the normal matrix factorization model. Yap et al. [71] introduce a competence score measure in personalized sequential pattern mining for next-item recommendations. Chen et al. [9] take playlists as MC, and propose logistic Markov embeddings to learn the representations of songs for playlists prediction. Wu et al. [68] propose Personalized Markov Embedding (PME) to consider sequential singing behavior for the next song recommendation. They embed users and songs into Euclidean space where the distance between songs and users represent their relationships. Given each user's last song, they can generate personalized recommendations by ranking the candidate songs according to the relationships. Lu et al. [43] argue that source domain data are not always consistent with the observations in the target domain, which may misguide the target domain recommendation. They propose a criterion based on empirical prediction error and its variance to better capture the consistency across domains in CF settings. To address the sparsity long-tailed distribution issues of most recommendation datasets and meanwhile take sequential dynamics into consideration, He and McAuley [21] propose to combine the advantages of MC based methods and CF based methods. They fuse a similarity-based method with MC to learn a personalized weighting scheme over the sequence of items to characterize users in terms of both preferences and the strength of sequential behavior. All above MC or MDP based sequential recommendation methods show improvements by modeling sequential dynamics. However, a major issue of them is that they can only consider a very short sequence (the most recent five items in Shani et al. [56]'s case), because the state space quickly becomes unmanageable when taking a long sequence into account [50].

Deep learning based methods. Recently, RNNs have been devised to model variable-length sequential data [72]. Quadrana et al. [51] have been the first to apply RNNs to sequential recommendation and achieve significant improvements over traditional methods. They utilize session-parallel mini-batch training and employ ranking-based loss functions to train the model. Later, they further propose data augmentation techniques to improve the performance of RNNs [59]. Contextual information has been proved to be very important for behavior modeling in traditional recommendations. Liu et al. [41] incorporate contextual information into SR and propose a context-aware RNN model. Instead of using the constant input matrix and transition matrix in conventional RNN models, CA-RNN employs adaptive They use context-specific input matrices to capture external situations where user behaviors happen, such as time, location, weather and so on. They also use context-specific transition matrices to capture how lengths of time intervals between adjacent behaviors in historical sequences affect the transition of global sequential features. Hidasi et al. [26] 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; they propose alternative training strategies for p-RNNs that suit them better than standard training. Bogina and Kuflik [6] explore user's dwell time based on an existing RNN-based framework by boosting items above a predefined dwell time threshold. Donkers et al. [14] introduce a new gated architecture with additional input layers for gated recurrent unit (GRU) to explicitly represent the individual user in such a network, which are uniquely designed and optimized for the purpose of generating personalized next item recommendations. Quadrana et al. [51] propose a hierarchical RNN model that can be used to generate personalized sequential recommendations. Li et al. [38] explore a hybrid encoder with an attention mechanism to model the user's sequential behavior and intent to capture the user's main purpose in the current sequence. Zhuang et al. [76] 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. Tang and Wang [60] propose a convolutional sequence embedding recommendation model for personalized top-n sequential recommendation to address the more recent items where they argue that more recent items in a sequence have a larger impact on the next item. Ren et al. [52] propose a repeat aware RNN model to address the repeat consumption in SR which is a common phenomenon in many recommendation scenarios where the same item is re-consumed repeatedly over time. They incorporate a new repeat recommendation mechanism into RNN that can choose items from a user's history and recommends them at the right time. Memory enhanced RNN has been well studied for SR recently. Chen et al. [10] argue that existing SR methods usually embed a user's historical records into a single latent representation, which may have lost the per item- or feature-level correlations between a user's historical records and future interests. They introduce the memory mechanism to SR and design a memory-augmented neural network integrated with the insights of collaborative filtering for recommendation. Huang et al. [30] propose knowledge enhanced SR model to capture fine-grained user preference from the interaction sequence. They integrate the RNN-based networks with key-value memory network. They further incorporate knowledge base information to enhance the learned semantic representations. Ma et al. [45] propose a cross-attention memory network to perform the mention recommendation task for multi-modal tweets where they make full use of both textual and visual information. Huang et al. [29] introduce a taxonomy-aware multi-hop reasoning network, which integrates a basic GRU-based sequential recommender with an elaborately designed memory-based multi-hop reasoning architecture. They incorporate taxonomy data as structural knowledge to enhance the reasoning capacity. Wang et al. [64] hypothesize that the collaborative information contained in neighborhood sequence (that have been generated previously by other users and reflect similar user intents as the current sequence) might help to improve recommendation performance for the current sequence. They propose a RNN model with 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 [32]. Although there are many studies for SRs using RNNs, none considers shared accounts and cross-domain simultaneously.

2.2 Shared account recommendations

Most recommender systems assume that every account in data represents a single user. However, multiple users often share a single account. A typical example is a smart TV account for the whole family.

Most previous approaches to shared account recommendations first identify users and then make personalized recommendations [2, 13, 63, 74]. Zhang et al. [73] are the first to study user identification, in which they use linear subspace clustering algorithms; they recommend the union of items that are most likely to be rated highly by each user. Bajaj and Shekhar [3] propose a similarity-based channel clustering method to group similar channels for IPTV accounts, and they use the Apriori algorithm to decompose users under an account. After that, they use personal profiles to recommend additional channels to the account. Wang et al. [67] suppose that different users consume services in different periods. They decompose users based on mining different preferences over different time periods from consumption logs. Finally, they use a standard User-KNN method to make recommendations for each identified user. Yang et al. [70] also analyze the similarity of the proportion of each type of items under a time period to judge whether a sequence is generated by the same user. Then they make recommendations to the specific user individually by recommending personalized genres to the identified users. Lesaege et al. [36] develop a time-aware user identification model based on the Latent Dirichlet Allocation using a hidden variable to represent the user and assume consumption times to be generated by latent time topics. Yang et al. [69] identify users by using a projection based unsupervised method, and then use Factorization Machine techniques to predict a user's preference based on historical information to generate personalized recommendations. Jiang et al. [33] propose an unsupervised learning-based framework to identify users and differentiate the preferences of users and group sessions by users. They construct a heterogeneous graph to represent items and use a normalized random walk to create item-based session embeddings. A hybrid recommender is then proposed that linearly combines the account-level and user-level item recommendation by employing Bayesian personalized ranking matrix factorization (BPRMF) [53].

The differences between our method and above methods are at least three-fold. First, the work described above achieves user identification and recommendation in two separate processes, which means that the proposed models are not end-to-end learnable. Second, they do not consider the cross-domain scenario on which we focus. Third, they are not RNN based and most of them ignore sequential dynamics.

2.3 Cross-domain recommendations

Cross-domain recommendation concerns data from multiple domains, which has proven to be helpful for alleviating the cold start problem [1, 5] and data sparsity issues [37, 48]. There is an assumption that there exists overlap in information between users and/or items across different domains [15, 16].

Traditional methods. There are two main ways in dealing with cross-domain recommendations [17]. One is to aggregate knowledge between two domains. Berkovsky et al. [4] propose four mediation techniques to solve the data sparsity problem by merging user preferences and extracting common attributes of users and items. Tang et al. [61] propose a cross-domain topic learning model to address three challenges in cross-domain collaboration recommendation: sparse connection (cross-domain collaborations are rare); complementary expertise (cross-domain collaborators often have different expertise and interest) and topic skewness (cross-domain collaboration topics are focused on a subset of topics) Shapira et al. [57] compare several collaborative methods to demonstrate the effectiveness of utilizing available preference data from Facebook. Loni et al. [42] model user preference by using Matrix Factorization separately on different domains, and then incorporate user interaction patterns that are specific to particular types of items to generate recommendations on the target domain. Zhuang et al. [75] propose a consensus regularization classifier framework by considering both local data available in source domain and the prediction consensus with the classifiers learned from other source domains. Cao et al. [7] construct a nonparametric Bayesian framework by jointly considering multiple heterogeneous link prediction tasks between users and different types of items. Chen et al. [8] exploit the users and items shared between domains as a bridge to link different domains by embedding all users and items into a low-dimensional latent space between different domains. The other approach to cross-domain recommendation is to transfer knowledge from source domain to target domain. Hu et al. [28] propose tensor-based factorization to share latent features between different domains. Cremonesi and Quadrana [13] propose a code-book-transfer to transfer rating patterns between domains. Kanagawa et al. [34] propose a content-based approach to learn the semantic information between domains. However, compared with deep learning methods, they are all shallow methods and have difficulties in learning complex user-item interactions.

Deep learning based methods. Deep learning is well suited to transfer learning as it can learn high-level abstractions among different domains. Lian et al. [39] first introduce a factorization framework to tie collaborative filtering and content-based filtering together; they use neural networks to build user and item embeddings. Elkahky et al. [15] propose a multi-view deep learning recommendation system by using rich auxiliary features to represent users from different domains based on *Deep Structured Semantic Model* (DSSM) [31]. Fernández-Tobías et al. [18] address the cold-start issue in a target domain by exploiting user preferences from a related auxiliary domain. They show that cross-domain information is useful to provide more accurate and diverse recommendations when user feedback in the target domain is scarce or not available at all. Hu et al. [27] propose a model using a cross-stitch network [47] to learn complex user-item interaction relationships based on neural collaborative filtering [24]. Zhuang et al. [76] propose a graphic novelty-seeking model to fully characterize users' novelty-seeking trait so as to obtain better performances between different domains. Wang et al. [66] are the first to introduce the problem of cross-domain social recommendation, and they combine user-item interactions in information domains and user-user connections in social network services to recommend relevant items of information domains to target users of social domains; user and item attributes are leveraged to strengthen the embedding learning.

Although these studies have been proven effective in many applications, they are designed for static rating data, and cannot be applied to sequential recommendations, unlike the methods that we introduce in this paper. Besides, none of them simultaneously considers shared-account which is also common in reality.

3 METHOD

In this section, we first give a formulation of the SCSR problem. Then, we give a high-level introduction to two ψ-Net variants. Finally, we describe each component of π-Net in detail.

3.1 Shared-account Cross-domain Sequential Recommendation

We represent a cross-domain behavior sequence (e.g., watching videos, reading books) from a shared account as $S = \{A_1, B_1, B_2, \ldots, A_i, \ldots, B_j, \ldots\}$, where $A_i \in \mathbb{A}$ $(1 \leq i \leq N)$ is the index of one consumed item in domain A; $\mathbb{A}$ is the set of all items in domain A; $B_j \in \mathbb{B}$ $(1 \leq j \leq M)$ is the index of one consumed item in domain B; $\mathbb{B}$ is the set of all items in domain B. Given S, SCSR tries to predict the next item by computing the recommendation probabilities for all candidates in two domains respectively, as shown in Eq. 1:

$$\begin{aligned} P(A_{i+1}|S) &\sim f_A(S) \\ P(B_{j+1}|S) &\sim f_B(S), \end{aligned} \tag{1}$$

where $P(A_{i+1}|S)$ denotes the probability of recommending the item A_{i+1} that will be consumed next in domain A. Also, $f_A(S)$ is the model or function to estimate $P(A_{i+1}|S)$. Similar definitions apply to $P(B_{j+1}|S)$ and $f_B(S)$.

The main differences between SCSR and traditional SR are two-fold. First, S is generated by multiple users (e.g., family members) in SCSR while it is usually generated by a single user in SR. Second, SCSR considers information from both domains for the particular recommendations in one domain, i.e., S is a mixture of behaviors from multiple domains. In this paper, we only consider two domains but the ideas easily generalize to multiple domains.

Next, we will describe two π-Net variants in detail. The key idea of π-Net is to learn a recommendation model that can first extract the information of some specific user roles from domain A, and then transfer the information to domain B, and combine it with the original information from domain B to improve recommendation performance for domain B, and vice versa. This process is achieved in a parallel way, which means that the information from both domains are shared recurrently at each timestamp.

3.2 Sequence encoder

Both variants use the same sequence encoder. Like existing studies [25, 51, 59], we use a RNN to encode a sequence S. Here, we employ a GRU as the recurrent unit, with the GRU given as follows:

$$
\begin{aligned}
z_t &= \sigma(W_z[emb(x_t), h_{t-1}]) \\
r_t &= \sigma(W_r[emb(x_t), h_{t-1}]) \\
\widetilde{h_t} &= \tanh(W_{\widetilde{h}}[emb(x_t), r_t \odot h_{t-1}]) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t},
\end{aligned}
\tag{2}
$$

where W_z, W_r, and $W_{\widetilde{h}}$ are weight matrices; $emb(x_t)$ is the item embedding of item x at timestamp t; σ denotes the sigmoid function. The initial state of the GRUs is set to zero vectors, i.e., $h_0 = 0$. Through the *sequence encoder* we obtain $H_A = \{h_{A_1}, h_{A_2}, \ldots, h_{A_i}, \ldots, h_{A_N}\}$ for domain A, and $H_B = \{h_{B_1}, h_{B_2}, \ldots, h_{B_j}, \ldots, h_{B_M}\}$ for domain B. We consider the last state as the in-domain representation, i.e., $h_A = h_{A_N}$ for domain A and $h_B = h_{B_M}$ for domain B. The in-domain representations are combined with the cross-domain representations (i.e., $h_{(A\rightarrow B)}$ or $h_{(B\rightarrow A)}$) to compute the final recommendation score. In the next two subsections, we will describe two π-Net variants which adopt different frameworks to learn the cross-domain representations.

3.3 ψ-Net-I

In this subsection, we will describe ψ-Net-I, our first solution for SCSR, in detail. Figure 2 provides an overview of ψ-Net-I. ψ-Net-I is a "Segregation-by-Integration" framework where it segregates role-specific representations from the mixed user behaviors and simultaneously integrates them at each timestamp. Then the integrated representations are transformed to another domain as cross-domain representations. ψ-Net-I consists of three main components: a *sequence encoder* (See §3.2), a *segregation-by-integration unit* and a *hybrid recommendation decoder* (See §3.5). The *sequence encoder* encodes the behavior sequences of each domain into high-dimensional hidden representations. The *segregation-by-integration unit* takes each domain's representations as input and tries to first segregate the representations of specific user roles, and then integrates and transforms them to another domain at each timestamp t. The *matching decoder* combines the information from both domains and generates a list of recommended items. Please refer to §3.2 and 3.5 to see the details of *sequence encoder* and *hybrid recommendation decoder*. In this subsection, we mainly introduce the core module (i.e., *segregation-by-integration unit*) of ψ-Net-I.

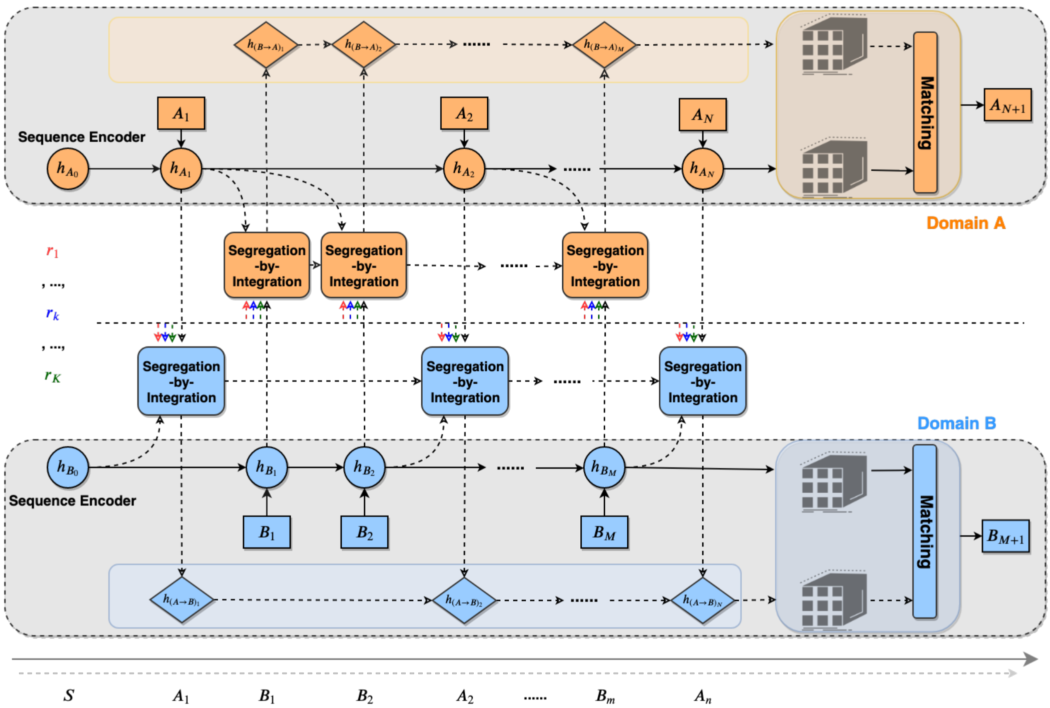

Fig. 2. An overview of ψ-Net-I. Different colors represent different domains. Section 3.3 contains a walkthrough of the model.

***Segregation-by-integration unit*.** Under the shared account scenario, the behavior records under the same account are generated by different users. In practice, not all users that share the account are active in all domains. Besides, even though some users are active in the same domain, they may have different interests. For example, in the smart TV scenario, children may occupy the majority of the educational channel, while adults dominate the video TV channel.

The outputs H_A or H_B of the *sequence encoder* are the mixed representations of all user roles sharing the same account. To learn role-specific representations from the mixed representations, we propose a *segregation-by-integration unit*, as shown in Figure 3. The *segregation-by-integration unit* 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 achieving recommendations in domain B as an example. To learn role-specific representations, we need to know the number of user roles under each account, which is, unfortunately, unavailable in most cases. In this work, we assume that there are K latent roles $(r_1, r_2, \ldots, r_k, \ldots, r_K)$ under each account. For example, in a family account, the user roles correspond to child, male parent, female parent, etc. We first embed each latent role into $emb(r_k)$ $(1 \leq k \leq K)$, which represents and encodes the latent interests of each role. Then, we segregate the specific representation for role r_k at timestamp i in domain A with Eq. 3:

$$h_{A_i}^{r_k} = f_{A_i}^{r_k} \odot \widehat{h}_{A_i}^{r_k} + (1 - f_{A_i}^{r_k}) \odot h_{A_{i-1} \to B}, \tag{3}$$

where $\odot$ represents element-wise multiplication. Mathematically, the representation $h_{A_i}^{r_k}$ is a combination of two representations $\widehat{h}_{A_i}^{r_k}$ and $h_{A_{i-1} \to B}$ balanced by $f_{A_i}^{r_k}$. A higher value of $f_{A_i}^{r_k}$ means that item A_i is more likely generated by r_k and we should pay more attention to r_k's related

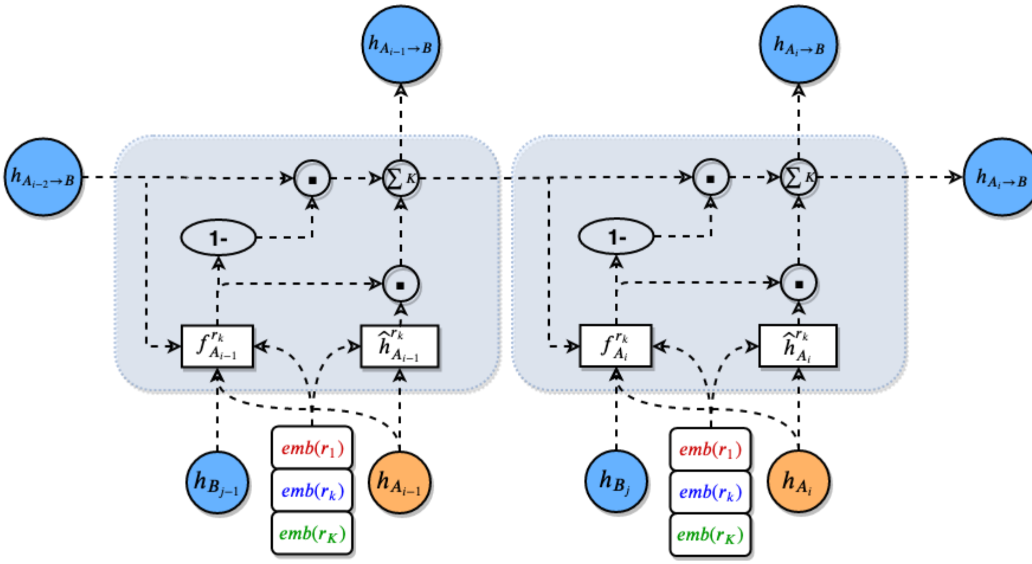

Fig. 3. Domain A to domain B *segregation-by-integration unit.*

representation $\widehat{h}^{r_k}_{A_i}$. A lower value of lower $f^{r_k}_{A_i}$ means that item A_i might not be related to r_k and we should inherit more information from previous time steps.

Next, we introduce the definitions of the three parts of Eq. 3 one by one.

(a) We propose a gating mechanism to implement $f^{r_k}_{A_i}$ in Eq. 4:

$$f^{r_k}_{A_i} = \sigma\left(W_{f_A}\cdot h_{A_i} + W_{f_B}\cdot h_{B_j} + U_f\cdot h_{A_{i-1}\to B} + V_f\cdot emb(r_k) + b_f\right), \tag{4}$$

where $\cdot$ means matrix multiplication. W_{f_A}, W_{f_B}, U_f and V_f are the parameters; b_f is the bias term; h_{A_i} and h_{B_j} are the mixed representations of domain A and B at timestamp i and j, respectively. $h_{A_{i-1}\to B}$ is the previous output, which will be explained later. After the sigmoid function σ, each value of $f^{r_k}_{A_i}$ falls into $(0,1)$. Thus, the gating score $f^{r_k}_{A_i}$ controls the amount of information of r_k to transfer from domain A to domain B. It should be noted that each latent representation $emb(r_k)$ indicates the distribution of user roles with similar preference under each account, and it does not explicitly represents a specific user.

(b) $\widehat{h}^{r_k}_{A_i}$ is the candidate representation for r_k at timestamp i in domain A, which is computed based on the mixed representation h_{A_i}, the filtered previous output $h_{A_{i-1}\to B}$, and the user role r_k's latent interest $emb(r_k)$, as shown in Eq. 5:

$$\widehat{h}^{r_k}_{A_i} = \tanh\left(W_h\cdot h_{A_i} + U_h\cdot h_{A_{i-1}\to B} + V_h\cdot emb(r_k) + b_h\right), \tag{5}$$

where W_h, U_h and V_h are the parameters; b_h is the bias term.

(c) $h_{A_i\to B}$ is the final output at timestamp i in domain A, which is calculated by integrating each role-specific representation $h^{r_k}_{A_i}$:

$$h_{A_i\to B} = \frac{1}{K}\sum_{k=1}^{K}\left(h^{r_k}_{A_i}\right). \tag{6}$$

Note that $h_{A_i\to B}$ is recurrently updated with Eq. 3 and 6.

After Eq. 3 and 6, we get a sequence of integrated representations $[h_{A_1 \to B}, \ldots, h_{A_N \to B}]$. We need to combine and transfer $[h_{A_1 \to B}, \ldots, h_{A_N \to B}]$ to domain B. Specifically, we achieve this by employing another GRU to recurrently encode $h_{A_i \to B}$ at each timestep to obtain a $h_{(A \to B)_i}$, as shown in Eq. 7:

$$h_{(A \to B)_i} = GRU(h_{A_i \to B}, h_{(A \to B)_{i-1}}), \tag{7}$$

where $h_{A_i \to B}$ is the representation filtered from domain A; $h_{(A \to B)_{i-1}}$ is the previous transformed representation at timestamp $i-1$. The initial state is also set to zero vectors, i.e., $h_{(A \to B)_0} = 0$. We set the cross-domain representation from domain A to domain B (i.e., $h_{(A \to B)}$) as the last timestamp representation $h_{(A \to B)_N}$, where N is sequence length of domain A.

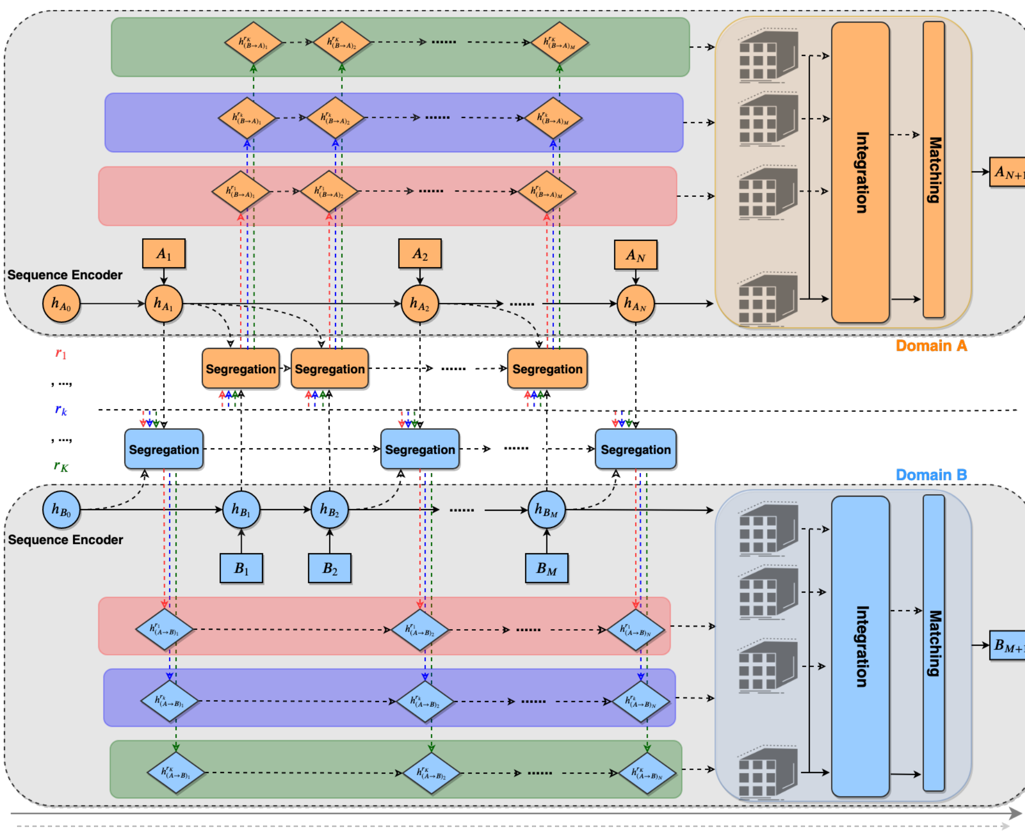

Fig. 4. An overview of ψ-Net-II. Different colors represent different domains. Section 3.4 contains a walk-through of the model.

3.4 ψ-Net-II

In this subsection, we will describe ψ-Net-II, our second solution for SCSR, in detail. Different from ψ-Net-I, ψ-Net-II is a "Segregation-and-Integration" framework which means it first segregates role-specific representations from the mixed user behaviors at each timestamp. Then the role-specific representations are transformed to another domain. Finally, it integrates the role-specific

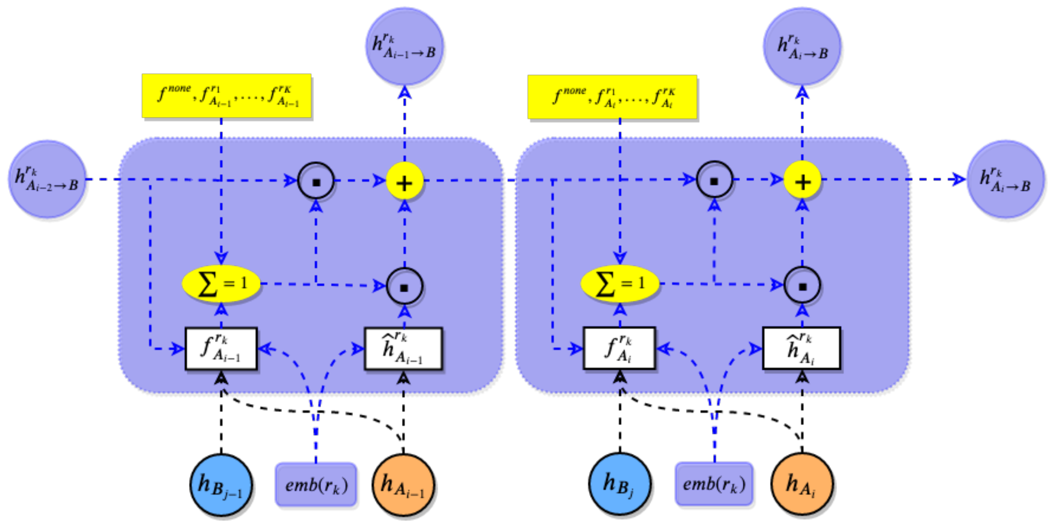

Fig. 5. Domain A to domain B *segregation unit* for r_k.

representations as cross-domain representations. Figure 4 provides an overview of ψ-Net-II. ψ-Net-II consists of four main components: a *sequence encoder* (See §3.2), a *segregation unit*, an *integration unit* and a *hybrid recommendation decoder* (See §3.5). ψ-Net-II uses the same *sequence encoder* and *matching decoder* architectures as ψ-Net-I. Please refer to §3.2 and 3.5 to see the details of *sequence encoder* and *hybrid recommendation decoder*. In this subsection, we mainly introduce the core modules (i.e., *segregation unit* and *integration unit*) of ψ-Net-II.

Segregation unit. The *segregation unit* is shown in Figure 5. The differences with the *segregation-by-integration unit* of ψ-Net-I are marked in yellow. As with ψ-Net-I, ψ-Net-II also assumes that there are K latent roles under each account. We segregate the specific representation for role r_k at timestamp i in domain A with Eq. 8:

$$h^{r_k}_{A_i\to B} = f^{r_k}_{A_i} \odot \widehat{h}^{r_k}_{A_i} + f^{none}_{A_i} \odot h^{r_k}_{A_{i-1}\to B}, \tag{8}$$

where $f^{none}_{A_i}$ is a special gate which handles the case when none of the information from r_k at i (i.e., $\widehat{h}^{r_k}_{A_i}$) is useful and we should inherit more information from previous time steps. We add normalization constraint to force the sum of $f^{r_k}_{A_i}$ and $f^{none}_{A_i}$ to 1.

$$f^{none}_{A_i} + \sum_{k=1}^{K} f^{r_k}_{A_i} = 1. \tag{9}$$

We use similar definitions of $f^{r_k}_{A_i}$ (Eq. 4) and $\widehat{h}^{r_k}_{A_i}$ (Eq. 5) as in ψ-Net-I, except that $h_{A_{i-1}\to B}$ is replaced with $h^{r_k}_{A_{i-1}\to B}$. The differences from *segregation-by-integration unit* are two-fold. First, $h^{r_k}_{A_i\to B}$ is not integrated with respect to all roles. Second, instead of learning independent gates for different roles, we require that all gate values from all roles (and $f^{none}_{A_i}$) are summed to 1.

After Eq. 8, we get a sequence of segregated representations $[h^{r_k}_{A_1\to B}, \dots, h^{r_k}_{A_n\to B}]$ for each user role r_k. We combine and transfer $[h^{r_k}_{A_1\to B}, \dots, h^{r_k}_{A_n\to B}]$ to the domain B by employing another GRU, as shown in Eq. 10:

$$h^{r_k}_{(A\to B)_i} = GRU(h^{r_k}_{A_i\to B}, h^{r_k}_{(A\to B)_{i-1}}), \tag{10}$$

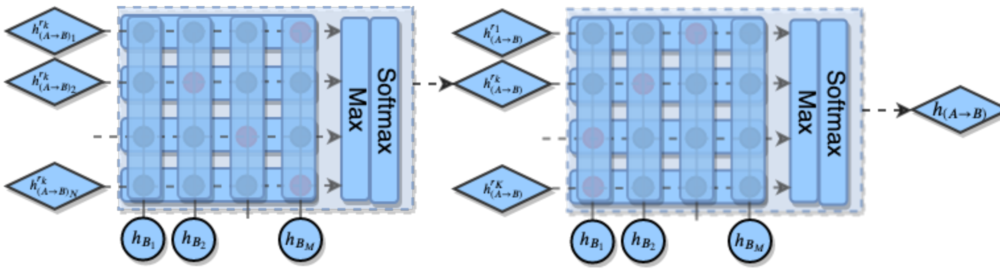

Fig. 6. Domain A to domain B *integration unit*.

where $h^{r_k}_{A_i \to B}$ is the representation filtered from domain A for role r_k.

Integration unit. The *integration unit* is shown in Figure 6. After the *segregation unit*, we get K sequences of transformed representations $[h^{r_k}_{(A\to B)_1}, \ldots, h^{r_k}_{(A\to B)_N}]$ from domain A to domain B. To integrate them, we first compute a similarity matrix $S^I \in \mathbb{R}^{M\times N}$ between the transformed representations and the in-domain representations $[h_{B_1}, \ldots, h_{B_M}]$ from domain B. Each similarity $S^I_{(i,j)}$ is computed with Eq. 11.

$$S^I_{(i,j)} = {v_S}^T \cdot (W_i \cdot h^{r_k}_{(A\to B)_i} + W_j \cdot h_{B_j}), \tag{11}$$

where ${v_S}^T$, W_i and W_j are parameters.

Then we pick the maximum similarity S^I_i between each $h^{r_k}_{(A\to B)_i}$ and all h_{B_j}. S^I_i signifies $h^{r_k}_{(A\to B)_i}$ is more representative for role r_k in domain B because it has the closest similarity to a representation h_{B_j} in domain B.

$$S^I_i = \max_j S^I_{(i,j)}. \tag{12}$$

We normalize S^I_i with softmax afterwards. Then we get the integrated representations for each role r_k in Eq. 13.

$$h^{r_k}_{(A\to B)} = \sum_{i=1}^{N} (S^I_i h^{r_k}_{(A\to B)_i}). \tag{13}$$

Finally, we get the cross-domain representation $h_{(A\to B)}$ by integrating $[h^{r_1}_{(A\to B)}, \ldots, h^{r_K}_{(A\to B)}]$ again with similar operations as in Eq. 12, 12 and 13, but with a different similarity matrix $S^{II} \in \mathbb{R}^{M\times K}$. Note that S^{II} is computed between $[h^{r_1}_{(A\to B)}, \ldots, h^{r_K}_{(A\to B)}]$ and $[h_{B_1}, \ldots, h_{B_M}]$ this time.

3.5 Hybrid recommendation decoder

The *hybrid recommendation decoder* integrates the hybrid information from both domains A and B to evaluate the recommendation probabilities of the candidate items. Specifically, it first gets the hybrid representation by concatenating the representation h_B from domain B and the transformed representation $h_{(A\to B)}$ from domain A to domain B. Then, it evaluates the recommendation probability for each candidate item by calculating the matching of between the hybrid representation and the item-embedding matrix followed by a softmax function, as shown in Eq. 14:

$$P(B_{j+1}|S) = \text{softmax}\left(W_I \cdot \left[h_B, h_{(A\to B)}\right]^\top + b_I\right), \tag{14}$$

where W_I is the embedding matrix of all items of domain B; b_I is the bias term.

3.6 Objective function

We employ the negative log-likelihood loss function to train π-Net in each domain as follows:

$$
\begin{aligned}
L_A(\theta) &= -\frac{1}{|\mathbb{S}|}\sum_{S\in\mathbb{S}}\sum_{A_i\in S}\log P(A_{i+1}|S)\\
L_B(\theta) &= -\frac{1}{|\mathbb{S}|}\sum_{S\in\mathbb{S}}\sum_{B_j\in S}\log P(B_{j+1}|S),
\end{aligned}
\tag{15}
$$

where θ are all the parameters of our model π-Net and $\mathbb{S}$ are the sequence instances in the training set. In the case of joint learning, the final loss is a linear combination of both losses:

$$
L(\theta) = L_A(\theta) + L_B(\theta). \tag{16}
$$

All parameters as well as the item embeddings are learned in an end-to-end back-propagation training way.

4 EXPERIMENTAL SETUP

4.1 Research questions

We seek to answer the following research questions in our experiments:

(RQ1) What is the performance of ψ-Net-I and ψ-Net-II in the SCSR task? Do they outperform the state-of-the-art methods in terms of Recall and MRR? (See Section 5.)

(RQ2) Which ψ-Net variant is more effective in the SCSR task? ψ-Net-I or ψ-Net-II? (See Section 5.)

(RQ3) What are the performances of ψ-Net-I and ψ-Net-II on different domains? Do they improve the performance of both domains? Are the improvements equivalent? (See Section 5.)

(RQ4) Is it helpful to leverage the cross-domain information? How well does the parallel modeling schema improve the performance of recommendations? (See Section 6.1.)

(RQ5) Is it helpful to model the shared-account characteristic? How well do the "integration" and "segregation" units improve the performance of recommendations? (See Section 6.1.)

(RQ6) How does the hyperparameter K (the number of latent user roles) affect the performance of ψ-Net-I and ψ-Net-II? Does the best K accord with reality? (See Section 6.2.)

(RQ7) How does the hyperparameter K change with the two ψ-Net variants and also the two SCSR scenarios? Is the best K the same under all situations? (See Section 6.2.)

4.2 Datasets

We need datasets that exhibit both share-account and cross-domain characteristics to conduct experiments. To demonstrate the effectiveness of the proposed ψ-Net model, we build and release two new datasets, named HAmazon and HVIDEO.

- HAMAZON: He and McAuley [22] release an Amazon product data which contains product reviews (ratings, text, helpfulness votes) and metadata (descriptions, category information, price, brand, and image features) from Amazon, including 142.8 million reviews spanning May 1996 – July 2014. The data contains user behaviors from multiple domains. In this paper, we use data from two Amazon domains. The M-domain contains *movie* watching and rating behaviors of Amazon users. The B-domain covers *book* reading and rating behaviors of Amazon users. We collect user-id, item-id, rating, and timestamp from the data and conduct some preprocessing. We first order the items by time under each account. Then, we merge records of the same item watched/read by the same user with an adjacent timestamp. We only keep the items whose frequency is larger than 5 in M-domain and 10 in B-domain. To satisfy cross-domain characteristics, we first find users who have behaviors in both Amazon movie and book domains and then only keep the users who have more than 10 records. To simulate shared-account characteristics, we first split

Table 1. Statistics of the datasets.

HAMAZON		HVIDEO	
M-domain		*V-domain*	
#Items	67,161	#Items	16,407
#Logs	4,406,924	#Logs	227,390
B-domain		*E-domain*	
#Items	126,547	#Items	3,380
#Logs	4,287,240	#Logs	177,758
#Overlapped-users	13,724	#Overlapped-users	13,714
#Sequences	289,160	#Sequences	134,349
#Training-sequences	204,477	#Training-sequences	102,182
#Validation-sequences	49,814	#Validation-sequences	18,966
#Test-sequences	34,869	#Test-sequences	13,201

time schedule into 6 intervals, which are 1996-2000, 2001-2003, 2004-2006, 2007-2009, 2010-2012, 2013-2015. Then, we combine data from both domains by randomly merging 2, 3, or 4 users and their behaviors in each interval as one shared account. Because each sequence has a lot of user behaviors recorded in a long time, we split the sequences from each account into several small sequences with each containing watching/reading records within a year. We also require that each sequence contains at least 5 items from M-domain and 2 items from B-domain. The length of each sequence is between 4 and 60 with an average length of 31.29. For evaluation, we use the last watched/read item in each sequence for each domain as the ground truth respectively. We randomly select 75% of all data as the training set, 15% as the validation set, and the remaining 10% as the test set. The statistics of the final dataset are shown in Table 1. Note that although HAMAZON can be used for experiments, it is not a SCSR dataset by nature. There are two shortcomings. First, the merged users do not naturally have the shared-account characteristic. Second, the two domains are quite different and not well correlated in content which means the items in different domains have little chance to reflect similar interests.

- HVIDEO: To facilitate future research for SCSR, we release another dataset, HVIDEO, which exhibits shared-account and cross-domain characteristics by nature. HVIDEO is a smart TV dataset that contains 260k users watching logs from October 1st 2016 to June 30th 2017. The logs are collected on two platforms (the V-domain and the E-domain) from a well-known smart TV service provider. The V-domain contains family *video* watching behavior including TV series, movies, cartoons, talent shows and other programs. The E-domain covers online *educational* videos based on textbooks from elementary to high school, as well as instructional videos on sports, food, medical, etc. On the two platforms, we gather user behaviors, including which video is played, when a smart TV starts to play a video, and when it stops playing the video, and how long the video has been watched. Compared with previous datasets, HVIDEO contains rich and natural family behavior data generated in shared-account and cross-domain context. Therefore, it is very suitable for SCSR research. We conduct some preprocessing on the dataset. We first filter out users who have less than 10 watching records and whose watching time is less than 300 seconds. Then, we merge records of the same item watched by the same user with an adjacent time less than 10 minutes. After that, we combine data from different domains together in chronological order which is grouped by the same account. Because each account has a lot of logs recorded in a long time, we split the logs from each account into several small sequences with each containing 30 watching records. And we require that the number of items in both domains must be greater than 5 in each sequence, which can ensure the sequences mix is high enough. For evaluation, we use the last watched item in each sequence for each domain as the ground truth respectively. We randomly select 75% of all data as the training set, 15% as the validation set, and the remaining 10% as the test set. The statistics of the final dataset are shown in Table 1.

4.3 Baseline methods

For our contrastive experiments, we consider baselines from four categories: traditional, sequential, shared account, and cross-domain recommendations.

4.3.1 Traditional recommendations. As traditional recommendation methods, we consider the following:

- POP: This method ranks items in the training set based on their popularity, and always recommends the most popular items. It is a very simple baseline, but it can perform well in certain domains and is frequently used as a baseline because of its simplicity [24].
- Item-KNN: The method computes an item-to-item similarity that is defined as the number of co-occurrences of two items within sequences divided by the square root of the product of the number of sequences in which either item occurs. Items that are similar to the actual item will be recommended by this baseline. Regularization is included to avoid coincidental high similarities between rarely visited items [40].
- BPR-MF: This model is a commonly used matrix factorization method. This model cannot be applied directly to SRs, because the new sequences do not have pre-computed feature vectors. Like [25], we apply it for sequential recommendations by representing a new sequence with the average latent factors of items that appeared in this sequence, i.e., we average the similarities of the feature vectors between a recommendable item and the items of the session so far.

4.3.2 Shared account recommendations. There are some studies that explore shared account recommendations by first achieving user identification [3, 33, 70]. However, they need extra information for user identification, e.g., some explicit ratings for movies or descriptions for some musics, even some textual descriptions for flight tickets, which is not available in our datasets. Therefore, we select a method that works on the IP-TV recommendation task that is similar to ours.

- VUI-KNN: This model encompasses an algorithm to decompose members in a composite account by mining different preferences over different time periods from logs [67]. The method first divides a day into time periods, so the logs of each account fall into the corresponding time period; logs in each time period are assumed to be generated by a virtual user that is represented by a 3-dimensional $\{account \times item \times time\}$ vector. After that, cosine similarity is used to calculate similarity among virtual users, some of which are merged into a latent user. VUI deploys User-KNN method to produce recommendations for these latent users.

4.3.3 Cross-domain recommendations. As cross-domain recommendations, we choose two baseline methods.

- NCF-MLP++: This model uses a deep learning-based process to learn the inner product of the traditional collaborative filtering by using Multilayer perceptron (MLP) [24]. We improve NCF-MLP by sharing the collaborative filtering in different domains. It is too time-consuming to rank all items with this method, because it needs to compute the score for each item one by one. We randomly sample four negative instances for each positive instance in the training process, and sample 3,000 negatives for each predicted target item in the test process, thus simplifying the task for this method.

- Conet: This ia a neural transfer model across domains on the basis of a cross-stitch network [27, 47], where a neural collaborative filtering model [24] is employed to share information between domains.

4.3.4 Sequential recommendations. Recently, a number of sequential recommendations methods have been proposed, among which RNN based neural methods have demonstrated superior performance over traditional MC or MDP based methods. There are many RNN based methods so far. In this paper, we consider two methods with somewhat similar architectures as ψ-Net.

- GRU4REC: This model uses GRU to encode sequential information. It uses a session-parallel mini-batch training process and employs ranking-based loss functions for learning the model [25].
- HGRU4REC: Quadrana et al. [51] propose this model based on RNNs which can deal with two cases: (i) user identifiers are present and propagate information from the previous sequence (user session) to the next, thus improving the recommendation accuracy, and (ii) there are no past sessions (i.e., no user identifiers). The model is based on a hierarchical RNN where the hidden state of a lower-level RNN at the end of one sequence is passed as an input to a higher-level RNN which aims at predicting a good initialization for the hidden state of the lower RNN for the next sequence.

4.4 Evaluation metrics

Recommender systems can only recommend a limited number of items at a time. The item a user might pick should be amongst the first few on the ranked list [11, 23, 51]. Commonly used metrics are MRR@20 and Recall@20 [38, 46, 52]. In this paper, we also report MRR@5, Recall@5 and MRR@10, Recall@10.

- Recall: The primary evaluation metric is Recall, which measures the proportion of cases when the relevant item is amongst the top ranked items in all test cases. Recall does not consider the actual rank of the item as long as it is amongst the recommendation list. This accords with certain real-world scenarios well where there is no highlighting of recommendations and the absolute order does not matter [25].
- MRR: Another used metric is MRR (Mean Reciprocal Rank), which 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 recommendation list. MRR takes the rank of the items into consideration, which is vital in settings where the order of recommendations matters. We choose MRR instead of other ranking metrics, because there is only one ground truth item for each recommendation; no ratings or grade levels are available.

For significance testing we use a paired t-test with $p < 0.05$.

4.5 Implementation details

We set the item embedding size and GRU hidden state size to 90. We use dropout [58] with drop ratio $p = 0.8$. These settings are chosen with grid search on the validation set. For the latent user size K, we try different settings, the analysis of which can be found in Section 6.2. We initialize model parameters randomly using the Xavier method [20]. We take Adam as our optimizing algorithm. For the hyper-parameters of the Adam optimizer, we set the learning rate $\alpha = 0.001$. We also apply gradient clipping [49] with range $[-5, 5]$ during training. To speed up the training and converge quickly, we use mini-batch size 64. We test the model performance on the validation set for every epoch. Both ψ-Net-I and ψ-Net-II are implemented in Tensorflow and trained on a GeForce GTX TitanX GPU.

Table 2. Experimental results (%) on the HAMAZON dataset.

Methods	M-domain recommendation						B-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
POP	0.36	0.44	0.49	0.73	1.32	2.02	0.14	0.19	0.22	0.42	0.78	1.22
Item-KNN	1.28	1.57	1.86	2.58	4.83	9.00	3.23	3.94	4.55	6.65	12.11	20.94
BPR-MF	2.90	3.00	3.06	3.90	4.65	5.50	0.88	0.92	0.96	1.23	1.50	2.15
VUI-KNN	–	–	–	–	–	–	–	–	–	–	–	–
NCF-MLP++	13.68	13.96	14.21	18.44	20.58	24.31	13.67	13.90	14.05	18.14	19.92	22.08
Conet	14.64	14.90	15.12	19.25	21.25	24.46	15.85	16.09	16.28	20.98	22.83	25.56
GRU4REC	82.01	82.08	82.11	83.10	83.61	84.06	81.34	81.41	81.44	82.77	83.32	83.76
HGRU4REC	83.07	83.12	83.14	84.24	84.65	84.91	82.15	82.26	82.31	83.46	84.30	84.91
ψ-Net-I	83.91	83.94	83.95	**84.91**	**85.13**	**85.33**	84.93	84.93	84.93	**85.33**	85.36	**85.38**
ψ-Net-II	**84.01**†	**84.04**†	**84.05**†	84.88	85.10	85.28	**85.10**†	**85.10**†	**85.11**†	85.32	**85.37**	**85.38**

Bold face indicates the best result in terms of the corresponding metric. Significant improvements over the best baseline results are marked with † (t-test, $p < .05$). To ensure a fair comparison, we re-implemented GRUE4REC and HGRU4REC in Tensorflow, just like ψ-Net; the final results may be slightly different from the Theano version released by the authors. To obtain the results of NCF-MLP++ and Conet, we run the code released by Hu et al. [27]. Same settings apply to Table 3. VUI-KNN does not work on this dataset because it needs specific time in a day which is not available on HAMAZON dataset.

Table 3. Experimental results (%) on the HVIDEO dataset.

Methods	V-domain recommendation						E-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
POP	2.66	3.07	3.27	5.01	7.77	10.49	1.71	1.96	2.24	2.21	3.61	6.58
Item-KNN	4.43	4.16	2.93	10.48	16.49	23.93	2.11	2.39	2.90	3.01	5.77	12.11
BPR-MF	1.21	1.31	1.36	1.88	2.56	3.38	1.34	1.52	1.64	2.74	4.05	5.83
VUI-KNN	3.44	3.53	2.87	16.46	24.85	34.76	2.03	2.51	3.48	6.36	11.55	24.27
NCF-MLP++	16.25	17.25	17.90	26.10	33.61	43.04	3.92	4.57	5.14	7.36	12.27	20.84
Conet	21.25	22.61	23.28	32.94	43.07	52.72	5.01	5.63	6.21	9.26	14.07	22.71
GRU4REC	78.27	78.46	78.27	80.15	81.63	83.04	12.27	13.00	13.70	16.24	21.89	32.16
HGRU4REC	80.37	80.53	80.62	81.92	83.21	84.43	14.47	15.37	16.11	19.79	26.72	37.52
ψ-Net-I	80.51	80.80	80.95	83.22	85.34	87.48	14.63	15.83	16.88	20.41	29.61	**45.19**
ψ-Net-II	**81.97**†	**82.20**†	**82.32**†	**84.32**†	**86.11**†	**87.75**†	**16.63**†	**17.62**†	**18.46**†	**22.12**†	**29.64**	42.20

Same settings are applied as in Table 2.

5 EXPERIMENTAL RESULTS (RQ1, RQ2 & RQ3)

To answer RQ1, RQ2 and RQ3, we report the results of ψ-Net compared with the baseline methods on the HAMAZON and HVIDEO datasets, as shown in Table 2 and 3 respectively. From the tables, we can see that both ψ-Net-I and ψ-Net-II outperform all baselines in terms of MRR and Recall for all reported values. Below, we discuss several insights we can get from Table 2 and 3.

First, both two ψ-Net variants significantly outperform all baselines and achieve the best results on all metrics, including strong baselines, i.e., GRU4REC and HGRU4REC. It is worth to note that although recent studies on SR propose many neural network models, we choose GRU4REC and HGRU4REC because GRU4REC and HGRU4REC have very similar architectures as ψ-Net. And to obtain a fair comparison, we re-implement them within the same TensorFlow framework as we use for ψ-Net. Specifically, on the HVIDEO dataset, the largest increase of ψ-Net-II over GRU4REC is 4.04% in terms of MRR@20, and 4.48% in terms of Recall@10 on the V-domain. On the E-domain, the increase is even larger with 4.70% increase of ψ-Net-II over GRU4REC in terms of MRR@20 and 13.03% increase of ψ-Net-I over GRU4REC in terms of Recall@20. And the increase over HGRU4REC on the V-domain is 1.69% and 3.45% (at most) in terms of MRR and Recall respectively. On the E-domain, the increase is 2.29% and 7.67% respectively. We believe that those performance improvements are due to the fact that ψ-Net considers two important factors (shared-account and cross-domain) with its parallel modeling architecture and two main components for as part of its end-to-end recommendation model, namely the "Segregation" and "Integration". With these three modules, ψ-Net is able to model user preferences more accurately by leveraging complementary information from both domains and improve recommendation performance in both domains. We will analyze the effects of the three modules in more depth in Section 6.1.

Second, we can see that ψ-Net-II is better than ψ-Net-I on both datasets generally. Specifically, ψ-Net-II outperforms ψ-Net-I in terms of most metrics on both domains on the HVIDEO dataset, especially for MRR@5 and Recall@5. Their performances are comparable on the HAMAZON dataset. But as we mentioned in §4.2, HAMAZON is not a good dataset for SCSR because the shared-account characteristic is simulated, and the two domains are quite different and not well correlated in content. Since both ψ-Net-I and ψ-Net-II adopt the parallel modeling architecture, we can conclude that the superiority of ψ-Net-II over ψ-Net-I mainly comes from the separate modeling of "Segregation" and "Integration". We will show this in more depth in Section 6.1.

Third, we can observe that the Recall values of ψ-Net on the HAMAZON dataset are comparable on two domains while Recall values of V-domain are better than those on the E-domain on the HVIDEO dataset. This is also true for the other methods. We believe this is because of the data balance issues. On the HAMAZON dataset, the data is generally balanced on two domains. Most accounts have equal amount of data on both domains. This means the models can learn pretty well with just one domain data. Cross-domain information is not that important. This can be proved by the fact that the increase of ψ-Net on the HAMAZON dataset is relatively small. However, this is totally different on the HVIDEO dataset. Most accounts have much more data on the V-domain due to its content diversity; because of this, models can better learn users viewing characteristics on the V-domain. Therefore, on the HAMAZON dataset, the space for potential improvements on both domains is smaller that that on the HVIDEO dataset. Additionally, comparing ψ-Net with the best baseline, HGRU4REC, we find that the largest increase on the E-domain is larger than on the V-domain. The largest increase in MRR is 1.69% on the V-domain and 2.29% on the E-domain. And for the Recall values, the largest increase is 3.45% on the V-domain, and 7.67% on the E-domain. This shows that the space for potential improvements on the V-domain is smaller than that on the E-domain after considering shared account and cross-domain information.

Fourth, RNN-based methods (e.g., GRU4REC, HGRU4REC, and our ψ-Net) perform much better than traditional methods, which demonstrates that RNN-based methods are good at dealing with sequential information. The reason is that they are able to learn better dense representations of the data through nonlinear modeling, which we assume is able to provide a higher level of abstraction. The shared account and cross-domain baselines (e.g., VUI-KNN, NCF-MLP++ and Conet) perform much worse than ψ-Net. They also perform substantially worse than HGRU4REC. This is because these shared account and cross-domain baselines ignore sequential information, VUI-KNN only considers the length of watching time, and NCF-MLP++ and Conet do not use any time information. Another reason is that NCF-MLP++ and Conet just map each overlapped account in both domains to the same latent space to calculate the user-item similarity. However, the existence of shared accounts makes it difficult to find the same latent space for different latent users under one account.

Table 4. Analysis of the parallel modeling, segregation unit and integration unit on the HAMAZON dataset.

Methods	M-domain recommendation						B-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
ψ-Net (-PSI)	77.26	77.44	77.51	82.22	83.52	84.39	81.69	81.72	81.73	85.03	85.27	85.34
ψ-Net-I (-SI)	83.30	83.32	83.33	84.03	84.20	84.31	84.04	84.04	84.04	85.31	85.35	**85.38**
ψ-Net-II (-S)	83.55	83.59	83.60	84.61	84.90	85.14	84.87	84.88	84.88	85.26	85.31	85.35
ψ-Net-II (-I)	82.28	82.35	82.38	84.02	84.52	84.92	83.42	83.45	83.46	84.79	84.96	85.08
ψ-Net-I	83.91	83.94	83.95	**84.91**	**85.13**	**85.33**	84.93	84.93	84.93	**85.33**	85.36	**85.38**
ψ-Net-II	**84.01**	**84.04**	**84.05**	84.88	85.10	85.28	**85.10**	**85.10**	**85.11**	85.32	**85.37**	**85.38**

ψ-Net (-PSI) is ψ-Net without parallel modeling, i.e., no cross-domain representations are used for recommendations. Without parallel modeling, both ψ-Net-I and ψ-Net-II become the same ψ-Net (-PSI). ψ-Net-I (-SI) is ψ-Net-I without "segregation-by-integration" unit. Because "segregation-by-integration" is a whole indivisible unit, there is no ψ-Net-I (-S) or ψ-Net-I (-I). ψ-Net-II (-S) is ψ-Net-II without the "segregation" unit and ψ-Net-II (-I) is ψ-Net-II without the "integration" unit.

Besides, VUI-KNN is not a deep learning method and it simply distinguishes user roles based on the fixed divided time periods in a day, which means it assumes there is only one family member in each time period. However, in the smart TV scenario, many people usually watch programs together [67]. This situation cannot be captured very well by VUI-KNN. And it requires the specific time of the user behaviors in a day in order to distinguish different user roles. That is why we cannot get its results on the HAMAZON dataset because there is no such information. In contrast, ψ-Net can extract components of each hidden user role according to their viewing records in another domain with the "Segregation" module, which proves to be informative. We can also see the results of BPR-MF are lower than POP, which indicates that most items users watched are very popular, which is also in line with the phenomenon of people like pursuing popularity in the video and book recommendation scenarios.

Fifth, the increases of MRR and Recall are different on two datasets. On the HAMAZON dataset, there is no significant difference for both MRR and Recall from @5 to @20. This means ψ-Net can already predict the groundtruth item within @5 for most cases. This is not true on the HVIDEO dataset, especially on the E-domain. Specifically, the largest increase is 2.25% for MRR from @5 to @20, and 24.78% for Recall from @5 to @20. Besides, the increase in Recall is greater than the increase in MRR. This is because Recall measures the proportion of relevant items when they are amongst the top-k list, while MRR takes the rank of the relevant items into consideration. As the size of k increases, the number of relevant items will increase, and consequently, Recall values will increase. However, the calculation of MRR is the reciprocal of the ranking of each positive item. So an increase k is bound to have a limited impact on the MRR.

6 EXPERIMENTAL ANALYSIS

6.1 Ablation study (RQ4 & RQ5)

We design an ablation study to answer RQ4 and RQ5, i.e., whether the proposed parallel modeling, the "segregation" and "integration" schema are helpful. The results are shown in Table 4 and 5. ψ-Net (-PSI) is the ψ-Net-I or ψ-Net-II without all the three parts, which degenerates into GRU4REC except that ψ-Net (-PSI) is jointly trained on two domains. ψ-Net-I (-SI) is ψ-Net-I without "segregation-by-integration" unit. ψ-Net-II (-S) is ψ-Net-II without the "segregation" unit and ψ-Net-II (-I) is ψ-Net-II without the "integration" unit (i.e., replacing the "integration" unit by summing up the outputs from the "segregation" unit). We can draw the following conclusions from the results.

Table 5. Analysis of the parallel modeling, segregation unit and integration unit on the HVIDEO dataset.

Methods	V-domain recommendation						E-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
ψ-Net (-PSI)	78.02	78.17	78.28	80.13	81.34	82.93	12.69	13.43	14.05	16.54	22.29	31.45
ψ-Net-I (-SI)	78.59	78.85	78.97	81.71	83.58	85.33	16.35	17.04	17.59	20.97	26.29	34.44
ψ-Net-II (-S)	81.61	81.85	81.99	83.93	85.73	87.71	15.94	17.01	17.84	20.96	29.18	41.38
ψ-Net-II (-I)	81.76	81.98	82.12	84.20	85.80	**87.77**	16.43	17.48	**18.46**	21.92	**29.96**	44.30
ψ-Net-I	80.51	80.80	80.95	83.22	85.34	87.48	14.63	15.83	16.88	20.41	29.61	**45.19**
ψ-Net-II	**81.97**	**82.20**	**82.32**	**84.32**	**86.11**	87.75	**16.63**	**17.62**	**18.46**	**22.12**	29.64	42.20

Same settings are applied as in Table 4.

First, the best results are almost all from ψ-Net-I and ψ-Net-II, which demonstrates the effectiveness of combining all the three parts. Especially, the three parts bring around 7% (MRR) and 1%-3% (Recall) improvements on the M-domain of HAMAZON, and around 4% (MRR) and 4%-10% (Recall) on both domains of HVIDEO. Additionally, we see that ψ-Net (-PSI) gets the lowest performance amongst the these methods, while it still outperforms most of the baselines listed in Table 2 and 3. In summary, then, combining information from an auxiliary domain is useful. The MRR improvements are larger on HAMAZON while the Recall improvements are larger on HVIDEO. This is due to the different characteristics and situations of different domains. Take the two domains in HVIDEO for example. Almost all members have viewing records in the V-domain. However, items on the E-domain are mostly educational programs, so children take up a large proportion, and their educational interests are relatively fixed. As a result, the information extracted from the V-domain mostly belongs to children, which is less helpful because we already have enough data on the E-domain to learn such features in most cases.

Second, generally the parallel modeling brings the most improvements followed by the "segregation" and "integration" units. Specifically, ψ-Net-I achieves around 5% (MRR) and 2% (Recall) improvements on the M-domain of HAMAZON with the parallel modeling while the further improvements with the "segregation-by-integration" unit are just around 0.6% (MRR) and 1% (Recall). Similar results can be found on B-domain of HAMAZON and E-domain of HVIDEO. We believe this is because that the model is already able to leverage the information from both domain to achieve recommendations with the parallel modeling schema, which for sure will greatly improve the results. However, this does not mean that the parallel modeling schema is good enough because it could be further improved by taking the other factors, e.g. the shared-account characteristics, into account in order to better leverage the cross-domain information. This is why the "segregation" and "integration" units could further improve the results upon the parallel modeling schema. An exception is that the "segregation" and "integration" units achieve more improvements than the parallel modeling on the V-domain of HVIDEO for ψ-Net-I. We think the reason is that ψ-Net-I (-SI) cannot effectively use the cross-domain information without the "segregation-by-integration" unit, while ψ-Net-II (-S) is better because the function of "segregation" unit is replaced by the "integration" unit to some extent. The same is true for ψ-Net-II (-I). This could be verified by the fact that both ψ-Net-I and ψ-Net-II get big improvements with both units than with neither, but the improvements are smaller than with one unit for ψ-Net-II.

Third, the "segregation" unit is generally more effective than the "integration" unit for ψ-Net-II as we find the gap between ψ-Net-II and ψ-Net-II (-I) is smaller than that between ψ-Net-II and ψ-Net-II (-S). On the one hand, this shows that the "segregation" unit plays a more important role which addresses the challenge raised by shared accounts, i.e., filtering out information of some user

Table 6. Analysis of the hyperparameter K on the HAMAZON dataset.

K values	M-domain recommendation						B-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
						ψ-Net-I						
1	82.45	82.52	82.54	84.23	84.69	85.07	84.72	84.73	84.73	85.29	85.35	85.38
2	83.35	83.40	83.41	84.66	85.02	85.18	84.74	84.75	84.75	85.30	85.25	85.37
3	83.65	83.68	83.70	84.81	85.08	85.30	84.89	84.89	84.89	85.32	85.35	85.38
4	**83.91**	**83.94**	**83.95**	**84.91**	**85.13**	**85.33**	84.93	84.93	84.93	**85.33**	**85.40**	85.38
5	83.73	83.76	83.78	84.82	85.08	85.32	**84.94**	**84.94**	**84.94**	**85.33**	85.38	**85.39**
						ψ-Net-II						
1	84.33	**84.36**	**84.37**	**85.01**	**85.19**	**85.32**	85.09	85.10	85.10	85.32	85.36	85.39
2	**84.08**	84.12	84.13	84.92	85.15	85.30	85.13	85.13	85.13	**85.33**	85.36	**85.40**
3	84.03	84.06	84.07	84.92	85.12	85.29	**85.16**	**85.16**	**85.16**	**85.33**	85.35	85.37
4	84.01	84.04	84.05	84.88	85.10	85.28	85.10	85.10	85.11	85.32	**85.37**	85.38
5	82.34	82.42	82.44	84.06	84.63	84.99	84.67	84.68	84.69	85.23	85.30	85.37

Table 7. Analysis of the hyperparameter K on the HVIDEO dataset.

K values	V-domain recommendation						E-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
						ψ-Net-I						
1	80.19	80.50	80.66	82.85	85.15	87.40	13.92	15.06	16.10	19.76	28.74	43.98
2	80.48	80.75	80.91	83.08	85.06	87.31	14.29	15.47	16.54	19.83	28.96	44.77
3	80.53	80.79	80.93	**83.34**	85.31	87.31	14.45	15.54	16.64	20.23	28.61	44.64
4	80.51	80.80	80.95	83.22	**85.34**	**87.48**	**14.63**	**15.83**	**16.88**	20.41	**29.61**	**45.19**
5	**80.60**	**80.86**	**81.02**	83.25	85.19	87.47	14.59	15.71	16.75	**20.42**	28.97	44.38
						ψ-Net-II						
1	81.93	82.18	82.32	**84.33**	**86.17**	**88.21**	16.17	17.18	18.13	21.42	29.23	**43.29**
2	81.80	82.04	82.17	84.26	86.05	87.90	16.62	17.67	18.55	21.60	29.60	42.63
3	81.86	82.08	82.20	84.14	85.80	87.53	**16.90**	**17.94**	**18.77**	**22.42**	**30.36**	42.51
4	**81.97**	**82.20**	**82.32**	84.32	86.11	87.75	16.63	17.62	18.46	22.12	29.64	42.20
5	81.78	82.02	82.14	83.99	85.67	87.68	16.78	17.84	18.66	22.01	30.07	42.13

roles that might be useful for another domain from the mixed user behaviors. On the other hand, the also shows that the current the "integration" unit is not effective enough as directly summing up the outputs from the "segregation" unit also achieves competitive performances, and/or the improvement space for the it is limited.

6.2 Influence of hyperparameter K (RQ6 & RQ7)

Both ψ-Net-I and ψ-Net-II introduce a hyperparameter K in the "segregation" unit which represents the number of latent users. To answer RQ6 and RQ7, we carry out experiments to study how the setting K affects recommendation performances of ψ-Net-I and ψ-Net-II on both datasets. Taking into account the popular sizes of families, we consider $K = 1, \ldots, 5$ and compare different values of K while keeping other settings unchanged. The results are shown in Table 6 and 7.

First, we can see that most best values of MRR and Recall are achieved when $K = 3, 4$, $K = 4$ for ψ-Net-I and $K = 3$ for ψ-Net-II especially. This is mostly consistent with the size of modern families on HVIDEO and the simulation settings on HAMAZON, which is another signal to show that our ψ-Nets are effective for handling shared accounts. For ψ-Net-I, the lowest MRR and Recall values are achieved when $K = 1$. But for ψ-Net-II, the gap between the best and worst performances is much smaller, which indicates that ψ-Net-II is less sensitive to K than ψ-Net-I.

Seond, both ψ-Net-I and ψ-Net-II show mostly consistent trends and conclusions on both datasets, i.e., the best K values are basically the same. On the one hand, this demonstrates the performance stability of both ψ-Net-I and ψ-Net-II. On the other hand, this also shows the

Third, although K can affect the recommendation performance, the influence is limited. As we can see that the largest gaps between the best and worst performances are 1.94% (MRR) and 0.56% (Recall) on HAMAZON, 0.78% (MRR) and 1.21% (Recall) on HVIDEO. This is because even if $K = 1, 2$, our models still consider the information of all members except that some members are modeled as a single latent user.

7 CONCLUSION AND FUTURE WORK

In this paper, we have proposed the new task of SCSR and released two datasets for the task. We have proposed a novel parallel segregation-integration modeling framework named ψ-Net for SCSR. Under this framework, we have proposed two variants, ψ-Net-I and ψ-Net-II, with different segregation-integration schemes. Experimental results demonstrate that ψ-Net outperforms state-of-the-art methods in terms of MRR and Recall, which means it is able to improve recommendations performance by modeling shared-account and cross-domain characteristics of SCSR. We also conducted extensive analysis experiments to show the effectiveness of the two ψ-Net variants.

A limitation of ψ-Net is that it works better only when we have shared information in two domains that are complementary to each other. When there is only one domain or the data in two domains share less information, ψ-Net only achieves comparable performance with state-of-the-art methods. As to future work, ψ-Net can be advanced in several directions. First, we assume the same number of latent user roles under each account in this study. This can be further improved by automatically detecting the number of user roles, e.g., adaptively setting the number of family members in smart TV scenarios. Second, we have simplified the architecture of ψ-Net (e.g., encoders, decoders and loss functions), because they are beyond the scope of this study and we hope to minimize their influence. It would be interesting to see whether more complex architectures will further improve the performance of ψ-Net. Third, side information (e.g., movie categories, attributes or labels, etc.) has been proven effective in improving recommendation performance in traditional recommendation [19, 62] and SR [10]. We hope to explore how to better incorporate side information into ψ-Net for SCSR.

ACKNOWLEDGMENTS

This work is supported by Ahold Delhaize, the Association of Universities in the Netherlands (VSNU), the Innovation Center for Artificial Intelligence (ICAI), the Natural Science Foundation of China (61672324, 61672322, 61972234, 61902219), the Natural Science Foundation of Shandong province (2016ZRE27468), the Tencent AI Lab Rhino-Bird Focused Research Program (JR201932), and the Fundamental Research Funds of Shandong University. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

CODE AND DATA

The code used to run the experiments in this paper is available at https://bitbucket.org/Catherine_Ma/sequentialrec/src/master/tois-PsiNet/code/. The datasets released in this paper are shared at https://bitbucket.org/Catherine_Ma/sequentialrec/src/master/tois-PsiNet/datasets/.

REFERENCES

[1] Fabian Abel, Eelco Herder, Geert-Jan Houben, Nicola Henze, and Daniel Krause. 2013. Cross-system User Modeling and Personalization on the Social Web. *User Modeling and User-Adapted Interaction* 23, 2 (2013), 169–209.

[2] Michal Aharon, Eshcar Hillel, Amit Kagian, Ronny Lempel, Hayim Makabee, and Raz Nissim. 2015. Watch-it-next: A Contextual TV Recommendation System. In *Machine Learning and Knowledge Discovery in Databases*. Springer International Publishing, Cham, 180–195.

[3] Payal Bajaj and Sumit Shekhar. 2016. Experience Individualization on Online TV Platforms Through Persona-based Account Decomposition. In *Proceedings of the 24th ACM International Conference on Multimedia (MM '16)*. ACM, New York, NY, USA, 252–256.

[4] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Cross-Domain Mediation in Collaborative Filtering. In *Proceedings of the 11th International Conference on User Modeling (UM '07)*. Springer-Verlag, Berlin, Heidelberg, 355–359.

[5] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2008. Mediation of User Models for Enhanced Personalization in Recommender Systems. *User Modeling and User-Adapted Interaction* 18, 3 (2008), 245–286.

[6] Veronika Bogina and Tsvi Kuflik. 2017. Incorporating Dwell Time in Session-based Recommendations with Recurrent Neural Networks. In *In Proceedings of RecTemp Workshop co-located with ACM RecSys (RecTemp '17)*. ACM, New York, NY, USA, 57–59.

[7] Bin Cao, Nathan Nan Liu, and Qiang Yang. 2010. Transfer Learning for Collective Link Prediction in Multiple Heterogenous Domains. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML'10)*. Omnipress, USA, 159–166.

[8] Leihui Chen, Jianbing Zheng, Ming Gao, Aoying Zhou, Wei Zeng, and Hui Chen. 2017. TLRec: Transfer Learning for Cross-domain Recommendation. In *2017 IEEE International Conference on Big Knowledge (ICBK '17)*. IEEE, New York, NY, USA, 167–172.

[9] Shuo Chen, Josh L. Moore, Douglas Turnbull, and Thorsten Joachims. 2012. Playlist Prediction via Metric Embedding. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. ACM, New York, NY, USA, 714–722.

[10] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential Recommendation with User Memory Networks. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, New York, NY, USA, 108–116.

[11] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan Kankanhalli. 2018. Aspect-Aware Latent Factor Model: Rating Prediction with Ratings and Reviews. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 639–648.

[12] Zhiyong Cheng, Jialie Shen, Lei Zhu, Mohan S Kankanhalli, and Liqiang Nie. 2017. Exploiting Music Play Sequence for Music Recommendation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI '17)*. AAAI Press, 3654–3660.

[13] Paolo Cremonesi and Massimo Quadrana. 2014. Cross-domain Recommendations Without Overlapping Data: Myth or Reality?. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys '14)*. ACM, New York, NY, USA, 297–300.

[14] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-based Recurrent Neural Network Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 152–160.

[15] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 278–288.

[16] Aleksandr Farseev, Ivan Samborskii, Andrey Filchenkov, and Tat-Seng Chua. 2017. Cross-Domain Recommendation via Clustering on Multi-Layer Graphs. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. ACM, New York, NY, USA, 195–204.

[17] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain Recommender Systems: A Survey of the State of the Art. In *Proceedings of the 2nd Spanish Conference on Information Retrieval (CERI '12)*. –.

[18] Ignacio Fernández-Tobías, Paolo Tomeo, Iván Cantador, Tommaso Di Noia, and Eugenio Di Sciascio. 2016. Accuracy and Diversity in Cross-domain Recommendations for Cold-start Users with Positive-only Feedback. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 119–122.

[19] Rana Forsati, Mehrdad Mahdavi, Mehrnoush Shamsfard, and Mohamed Sarwat. 2014. Matrix Factorization with Explicit Trust and Distrust Side Information for Improved Social Recommendation. *ACM Transactions on Information Systems* 32, 4 (2014), 17:1–17:38.

[20] Xavier Glorot and Yoshua Bengio. 2010. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics Society for Artificial Intelligence and Statistics (AISTATS '10)*. 249–256.

[21] Ruining He and Julian McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *Proceedings of the IEEE 16th International Conference on Data Mining (ICDM '01)*. IEEE, New York, NY, USA, 191–200.

[22] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 507–517.

[23] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial Personalized Ranking for Recommendation. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 355–364.

[24] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 173–182.

[25] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *International Conference on Learning Representations (ICLR '16)*.

[26] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 241–248.

[27] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. ACM, New York, NY, USA, 667–676.

[28] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized Recommendation via Cross-domain Triadic Factorization. In *Proceedings of the 22Nd International Conference on World Wide Web (WWW '13)*. ACM, New York, NY, USA, 595–606.

[29] Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. 2019. Taxonomy-Aware Multi-Hop Reasoning Networks for Sequential Recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, New York, NY, USA, 573–581.

[30] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 505–514.

[31] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management (CIKM '13)*. ACM, New York, NY, USA, 2333–2338.

[32] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks Meet the Neighborhood for Session-Based Recommendation. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 306–310.

[33] Jyun-Yu Jiang, Cheng-Te Li, Yian Chen, and Wei Wang. 2018. Identifying Users Behind Shared Accounts in Online Streaming Services. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 65–74.

[34] Heishiro Kanagawa, Hayato Kobayashi, Nobuyuki Shimizu, Yukihiro Tagami, and Taiji Suzuki. 2018. Cross-domain Recommendation via Deep Domain Adaptation. *arXiv preprint arXiv:1803.03018* abs/1803.03018 (2018).

[35] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[36] ClÃľment Lesaege, FranÃğois Schnitzler, Anne Lambert, and Jean-Ronan Vigouroux. 2016. Time-Aware User Identification with Topic Models. In *2016 IEEE 16th International Conference on Data Mining (ICDM '16)*. IEEE, New York, NY, USA, 997–1002.

[37] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can Movies and Books Collaborate? Cross-domain Collaborative Filtering for Sparsity Reduction. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI '09)*. AAAI Press, 2052–2057.

[38] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. ACM, New York, NY, USA, 1419–1428.

[39] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. CCCFNet: A Content-Boosted Collaborative Filtering Neural Network for Cross Domain Recommender Systems. In *Proceedings of the 26th International Conference on*

World Wide Web Companion (WWW '17 Companion). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 817–818.

[40] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.Com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing* 7, 1 (2003), 76–80.

[41] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. 2016. Context-Aware Sequential Recommendation. In *Proceedings of the IEEE 16th International Conference on Data Mining (ICDM '16)*. IEEE, New York, NY, USA, 1053–1058.

[42] Babak Loni, Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Cross-domain Collaborative Filtering with Factorization Machines. In *Proceedings of the 4th Spanish Conference on Information Retrieval (CERI '14)*. 656–661.

[43] Zhongqi Lu, Erheng Zhong, Lili Zhao, Evan Wei Xiang, Weike Pan, and Qiang Yang. 2013. Selective Transfer Learning for Cross Domain Recommendation. In *Proceedings of the 2013 SIAM International Conference on Data Mining (SIAM '13)*. arXiv, Austin, Texas, USA, 641–649.

[44] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. π-Net: A Parallel Information-sharing Network for Cross-domain Shared-account Sequential Recommendations. In *The 42st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*. ACM, New York, NY, USA, 685–694.

[45] Renfeng Ma, Qi Zhang, Jiawen Wang, Lizhen Cui, and Xuanjing Huang. 2018. Mention Recommendation for Multimodal Microblog with Cross-attention Memory Network. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 195–204.

[46] Lei Mei, Pengjie Ren, Zhumin Chen, Liqiang Nie, Jun Ma, and Jian-Yun Nie. 2018. An Attentive Interaction Network for Context-aware Recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. ACM, New York, NY, USA, 157–166.

[47] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch Networks for Multi-task Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*. IEEE, New York, NY, USA, 3994–4003.

[48] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction. In *The 24th AAAI Conference on Artificial Intelligence (AAAI '10)*. AAAI Press, 230–235.

[49] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the Difficulty of Training Recurrent Neural Networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning (ICML '13)*. JMLR.org, III–1310–III–1318.

[50] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *Comput. Surveys* 51, 4, Article 66 (2018), 36 pages.

[51] Massimo Quadrana, Alexandros Karatzoglou, Balzs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 130–137.

[52] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-based Recommendation. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI '19)*. AAAI Press, 4806–4813.

[53] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, Arlington, Virginia, United States, 452–461.

[54] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-basket Recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. ACM, New York, NY, USA, 811–820.

[55] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. ACM, New York, NY, USA, 285–295.

[56] Guy Shani, David Heckerman, and Ronen I. Brafman. 2005. An MDP-Based Recommender System. *Journal of Machine Learning Research* 6 (2005), 1265–1295.

[57] Bracha Shapira, Lior Rokach, and Shirley Freilikhman. 2013. Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction* 23, 2 (2013), 211–247.

[58] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[59] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-based Recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS '16)*. ACM, New York, NY, USA, 17–22.

[60] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, New York, NY, USA, 565–573.

[61] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. 2012. Cross-domain Collaboration Recommendation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. ACM, New York, NY, USA, 1285–1293.

[62] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-Prod2Vec: Product Embeddings Using Side-Information for Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 225–232.

[63] Koen Verstrepen and Bart Goethals. 2015. Top-N Recommendation for Shared Accounts. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. ACM, New York, NY, USA, 59–66.

[64] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Ma Jun, and Maarten de Rijke. 2019. A Collaborative Session-based Recommendation Approach with Parallel Memory Modules. In *The 42st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*. ACM, New York, NY, USA, 345–354.

[65] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning Hierarchical Representation Model for Next Basket Recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. ACM, New York, NY, USA, 403–412.

[66] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item Silk Road: Recommending Items from Information Domains to Social Users. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. ACM, New York, NY, USA, 185–194.

[67] Zhijin Wang, Yan Yang, Liang He, and Junzhong Gu. 2014. User Identification Within a Shared Account: Improving IP-TV Recommender Performance. In *Advances in Databases and Information Systems (ADBIS '14)*. Springer International Publishing, Cham, 219–233.

[68] Xiang Wu, Qi Liu, Enhong Chen, Liang He, Jingsong Lv, Can Cao, and Guoping Hu. 2013. Personalized Next-song Recommendation in Online Karaokes. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. ACM, New York, NY, USA, 137–140.

[69] Shuo Yang, Somdeb Sarkhel, Saayan Mitra, and Viswanathan Swaminathan. 2017. Personalized Video Recommendations for Shared Accounts. In *2017 IEEE International Symposium on Multimedia (ISM '17)*. 256–259.

[70] Yan Yang, Qinmin Hu, Liang He, Minjie Ni, and Zhijin Wang. 2015. Adaptive Temporal Model for IPTV Recommendation. In *Web-Age Information Management (WAIM '15)*. Springer International Publishing, Cham, 260–271.

[71] Ghim-Eng Yap, Xiao-Li Li, and S Yu Philip. 2012. Effective Next-items Recommendation via Personalized Sequential Pattern Mining. In *Database Systems for Advanced Applications (DASFAA '12)*. Springer Berlin Heidelberg, Berlin, Heidelberg, 48–64.

[72] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A Dynamic Recurrent Model for Next Basket Recommendation. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. ACM, New York, NY, USA, 729–732.

[73] Amy Zhang, Nadia Fawaz, Stratis Ioannidis, and Andrea Montanari. 2012. Guess Who Rated This Movie: Identifying Users Through Subspace Clustering. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence (UAI '12)*. AUAI Press, Arlington, Virginia, United States, 944–953.

[74] Yafeng Zhao, Jian Cao, and Yudong Tan. 2016. Passenger Prediction in Shared Accounts for Flight Service Recommendation. In *The 10th International Conference on Asia-Pacific Services Computing (APSCC '16)*. Springer International Publishing, Cham, 159–172.

[75] Fuzhen Zhuang, Ping Luo, Hui Xiong, Yuhong Xiong, Qing He, and Zhongzhi Shi. 2010. Cross-domain Learning from Multiple Sources: A Consensus Regularization Perspective. *Transactions on Knowledge and Data Engineering* 22 (2010), 1664–1678.

[76] Fuzhen Zhuang, Yingmin Zhou, Fuzheng Zhang, Xiang Ao, Xing Xie, and Qing He. 2017. Sequential Transfer Learning: Cross-domain Novelty Seeking Trait Mining for Recommendation. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 881–882.

[77] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. 2001. Using Temporal Data for Making Recommendations. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI '01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 580–588.