Domain Switch-Aware Holistic Recurrent Neural Network for Modeling Multi-Domain User Behavior, Reproduction and Extension

Donghyun Kim, Sungchul Kim, Handong Zhao, Sheng Li, Ryan A. Rossi, Eunyee Koh

Abstract

In order to analyze sequential behaviors across multiple domains, they require to separately train multiple RNN models, which fails to jointly model the interplay among sequential behaviors across multiple domains. Consequently, they often suffer from lack of information within each domain. To solve these problems, a Domain Switch-Aware Holistic Recurrent Neural Network(DS-HRNN) is proposed. It effectively shares the knowledge extracted from multiple domains by systematically handling domain switch for the multi-domain scenario. In this report, I mainly focus on reproduction of this work according to the available paper on the website https://dl.acm.org/doi/pdf/10.1145/3289600.3291019 without the relevant code. Lastly, as this paper proposes a framework paradigm, this model also has a lot of places worth exploring and adequate space for improvement. I use transformer to replace RNN, and achieve better performance.

Keywords: Multi-domain user behavior, Sequence modeling, Domain switch, Recurrent neural network, Reproduction.

1 Introduction

With the diversity of web services, today's users on the web frequently switch from one service to another within short period of time, which makes their sequential behavior patterns more complex and diverse. For example, users access to Youtube to watch videos, use then Google to search information related to the content of videos and re-access to Youtube to browse videos depending on their search results in Google even within a short amount of time. Typically, such sequential behaviors across multiple domains are locally and globally correlated. In DS-HRNN [9], it is introduced as domain switches where two successive behaviors belong to different domains, e.g., from Youtube to Google and from Google to Youtube. Accordingly, from the marketer's point of view, accurately understanding domain switches from browsing web pages to clicking/buying items within each domain is an vital key when deciding marketing strategies for each domain.

Nowadays, RNNs have been actively used to analyze sequential behaviors in the user behavior modeling [1] or recommendation fields [2]. They have demonstrated their superiority to traditional approaches [15] by effectively considering sequential relationships of user behaviors [14]. However, existing RNN-based approaches have mostly focused on only a singe domain scenario of sequential behaviors. As a result, we must build a separate RNN model for each domain in order to analyze and predict domain-wise sequential behaviors. Unavoidably, they cannot exploit global dynamics of sequential behaviors contained across domains, which leads to an inferior performance of RNN models.

To solve these problems, Domain Switch-Aware Holistic Recurrent Neural Network (DS-HRNN) [9] is proposed. It can effectively addresses missing direct interactions in local dynamics, which is a common phenomenon. Specifically, DS-HRNN [9] first recovers missing direct interactions caused by domain switches, and compute domain switch-aware supplementary loss with respect to missing direct interactions. Moreover, to reflect correlations between global and local past behaviors at the end of each domain switch , it introduces domain switch-aware behavior regularizer. These two techniques attentively take into account local dynamics at every domain switches. Consequently, it not only can effectively reflect global dynamics into a single RNN model but also can preserve local dynamics without compromising further improvement in analyzing future behavior sequence within each domain. I reproduce two techniques and do experiments in two public data. The experiment results demonstrate the model achieves better performance by using the proposed two techniques. In the meantime, I replace transformer to RNN, and the model shows better performance.

2 Related works

The related works is similar to that of DS-HRNN [9].

2.1 RNN-based Sequential Behavior Modeling

In order to effectively model sequential behaviors, a RNN-based recommeder system was firstly introduced for session-based recommendation [6]. On top of that, a variety of RNN-based approaches for the next behavior prediction have been developed by additionally considering personalization [4], context-awareness [3], and different types of user behavior [12]. Personalization. Donkers et al. [4] devised a user-based Gated Recurrent Units (GRU) that attentively consider user embeddings along with sequential item information for personalized next item recommendations. Liu et al. [12] integrated the log-bilinear model [13] into RNN so that hidden sates can capture short-term and long-term contexts in user behavior sequences for predicting what a user will choose next. In addition to that, they tried to model multiple types of behaviors with behavior-specific transition matrices in their model. However, these RNN-based approaches have failed to consider the multi-domain scenario. Specifically, even if different types of behavior that are used as additional information of inputs are assumed to be domains, their approaches inevitably fall into the single-domain scenario since their objectives are modeled to predict next behaviors of users regardless of types of behaviors. Moreover, in existing approaches, one behavior can be concurrently assigned to multiple types such as click, add-to-cart and purchase, however, the multi-domain scenario has the distinctive property where one behavior is exactly assigned to only one domain. Therefore, how to consider the multi-domain scenario in users' sequential behaviors still remains a challenging problem.

2.2 Multi-Domain User Behavior Modeling

As one of cross-domain approaches, multi-domain user behavior modeling approaches have been explored mainly based on conventional non-sequential methods such as matrix factorization [10] and feed forward neural network [5]. It is worth noting that these approaches seamlessly learn shared knowledge across all domains so that the shared knowledge can be effectively used for better domain-wise user behavior prediction. Particularly, Li et al. [10] and Zhang et al. [?] utilized a cluster-level rating matrix from multiple rating matrices in order to share the knowledge collected from multiple domains. Yang et al. [18] observed that users have cross-site as well as site-specific preferences in multiple video websites, and thus they proposed a matrix factorization based model that infers site-specific user variables based on cross-site user variables. In their model, these site-specific user variables are used to predict ratings with video variables. However, these approaches do not seamlessly take into account sequential dynamics of user behaviors. To our best knowledge, DS-HRNN [9] is the first attempt to consider sequential dynamics as well as the multi-domain scenario in order to effectively predict future behaviors for each domain. Note that multi-task learning based RNNs [11] in the natural language processing (NLP) field might be regarded as our related work, but they differ from ours with the following reasons. With respect to objectives, existing approaches for NLP mainly try to simultaneously solve multiple classification problems based on text whereas our approach try to predict domain-wise next behavior based on previous behavior history across multi-domain. Moreover, inputs of existing approaches come from only a single source (e.g., English) whereas those of DS-HRNN [9] come from multiple sources (e.g., domain A, B and C). For example, two pairs of a sentence and a label from different tasks can share English words, however, sequences of behaviors from different domains never have common behaviors. Thus, the task of modeling multi-domain user behavior is more challenging than that of multi-task learning in NLP.

3 Method

3.1 Overview

As depicted in the Figure 1: example of (a)domain switches and two strategies for DS-HRNN (b)supplementary loss and (c)behavior regularizer. Dotted orange line denotes domain switches. The figure 1 is samed as that in DS-HRNN [9];

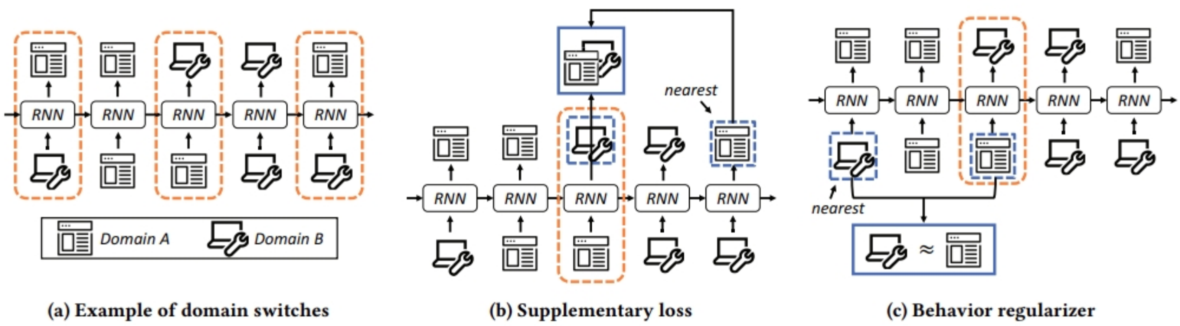

Figure 1. Overview of the method

The key idea of handling multi-domain user behavior are 1) aggregating user behaviors from multiple domains into one sequence in chronological order and 2) exploiting a single RNN model that takes sequences of multi-domain user behaviors. However, in the Multi-domain based approach with a Single RNN model, domain switches by the chronologically ordered aggregation are highly likely to hinder further improvements in multi-domain behavior prediction. The reason is that considering the loss of all domains in the single RNN model gives rise to the disconnection of direct interactions between behaviors in the same domain (i.e., local dynamics). Here, DS-HRNN [9] attribute this issue to the broken local dynamics, the disconnection of local dynamics is alleviated by proposing two types of domain switch aware techniques. Then, the final objective function and briefly describe the architecture of the single RNN model is introduced.

3.2 Domain switch-aware supplementary loss

The key idea for alleviating the disconnection of local dynamics is to recover the disconnection as much as possible while preserving global dynamics of sequential behaviors simultaneously. To this end, in terms of outputs, a domain switch-aware supplementary loss that is an explicit way of recovering the lost connection at domain switches $\mathcal{L}_s$ is defined as follows.

$$\mathcal{L}_s\left(\theta^g\right) = \sum_{t=2} \mathbb{I}\left[x_{t-1} \in \mathcal{B}^{(d)} \wedge x_t \notin \mathcal{B}^{(d)}\right] l\left(x_{t-1+n^{(d)}}, \mathcal{F}_{\theta^g}\left(\mathbf{x}_{<t}\right)\right) \tag{1}$$

where $\mathbb{I}[\cdot]$ is the indicator function, and $n^{(d)}$ is the distance from the current input $x_{t-1} \in \mathcal{B}^{(d)}$ to the nearest next behavior in the same domain d. Note that the indicator function lets the supplementary loss valid only at domain switches since disconnection does not occurs on other than domain change interactions. Figure 1 illustrates how $\mathcal{L}_s$ works in the single RNN model. Precisely, $\mathcal{L}_s$ searches the nearest next behavior which belong to the same domain, Domain A, as the current input of the RNN model on the domain switch, and $\mathcal{L}_s$ computes the correlation between these two behaviors. Through $\mathcal{L}_s$, we aim to explicitly inject local dynamics into the RNN model, which leads to more accurate sequential behavior prediction for each domain.

3.3 Domain switch-aware behavior regularizer

Similarly, alleviating the disconnection of local dynamics can be also achieved in an implicit way in terms of inputs. To be specific, Figure 1 shows that the input on the domain switch is likely to be correlated with the past nearest behavior who belongs to the same domain as the output on the domain switch. The reason is that these two behaviors for inputs can share the same output as the next behavior. Note that the past nearest behavior can be regarded as the past behavior of the current output in terms of local dynamics of the domain, Domain B. Eventually, local dynamics previously lost in the M-S approach can be implicitly recovered via taking into account these correlations for inputs. To reflect this, we introduce a domain switch-aware behavior regularizer $\mathcal{L}_r$ that minimizes the distance between correlated inputs from different domains:

$$\mathcal{L}_r\left(\theta^g\right) = \sum_{t=2} \mathbb{I}\left[x_{t-1} \notin \mathcal{B}^{(d)} \wedge x_t \in \mathcal{B}^{(d)}\right] \left\|\mathcal{F}_{\theta^g_{in}}\left(x_{t-m^{(d)}}\right) - \mathcal{F}_{\theta^g_{in}}\left(x_{t-1}\right)\right\|_2 \tag{2}$$

where $\mathcal{F}_{\theta^g_{in}}(x) \in \mathbb{R}^k$ denotes the k-dimensional embedding vector for input behavior x, and $m^{(d)}$ is the distance from the current input x_{t-1} to the nearest past behavior which belongs to the same domain d as the output x_t. This regularizer might seem to play the same role as the supplementary loss, however, the regularization focuses on building correlated input embeddings for different domains of behaviors, which is not explicitly taken into account in the supplementary loss technique.

3.4 Final objective & Architecture

Given the two techniques, the final objective $\mathcal{J}(\theta^g)$ for all users to be minimized is formulated as follows:

$$\mathcal{J}(\theta^g) = \sum_{\text{user}} \left(\mathcal{L}(\theta^g) + \lambda_s \cdot \mathcal{L}_s(\theta^g) + \lambda_r \cdot \mathcal{L}_r(\theta^g)\right) \tag{3}$$

where λ_s and λ_r are loss coefficients for the domain switch-aware supplementary loss and the domain switch-aware behavior regularizer, respectively. For $\mathcal{L}$ and $\mathcal{L}_s$, the cross entropy loss is used to deal with the probability distribution of next behaviors. The architecture for DS-HRNN [9] consists of three parts: 1) Input module, which projects raw sparse input of each behavior (e.g., one-hot vector) into low dimensional embedding space, 2) Recurrent module, which recursively updates the hidden state from the previous hidden state and the current input in order to reflect sequential information, 3) Output module, which computes the probability distribution of next behaviors for each domain.

4 Implementation details

4.1 Comparing with the released source codes

There is no related source codes in this paper. I reproduce DS-HRNN using python and pytorch.

4.2 Experimental environment setup

I do experments in public dataset. The dataset is same as that in MGCL [17]. The statistic of dataset is shown in table 1:

Table 1. The statistics of the processed Amazon dataset

Domain	#Users	#Items	#Actions	#Avg.Len	#Density
All domains	10929	382089	1412104	129.21	0.0003
Books	10929	236049	607657	55.60	0.0002
CDs	10929	91169	344221	31.50	0.0003
Movies	10929	59513	460226	42.11	0.0007

The training, valid and test dataset are obtained by splitting historical interaction sequence for each user by adopting one-leave-out evaluation. The last interacted item for testing, the second most recent interacted item for validation and all remaining interacted items for training.

For a fair comparison, the parameters of all the models are set similarly. For all models, the dropout rate is set to 0.5, the L_2 regularization coefficient is set to 0.001, the batch size is set to 128, and the learning rate is set to 0.001. The embedding size is searched from 10,20,30,40,50, and finally is set as 50, since the larger embedding size can gain better performance [8, 16, 17]. All models are trained by Adam optimizer for 500 epochs. The early stopping is adopted based on the evaluation on validation set. All models are evaluated every 20 epochs. If the evaluation performance doesn't improve for 40 epochs, the training will be determined. For the proposed model, the sequence length is set to 300 in all domains. For each user, 100 negative items that user doesn't interacted with in the certain domain are selected by popularity-based sampling [17].

Table 2. The results of the variants

Domain	performance	GRU_Rec	GRU_Rec_sl	GRU_Rec_sl_br	Transformer_Rec_sl_br
Books	NDCG@10	0.1187	0.1287	0.1502	0.1503
	HR@10	0.2222	0.2439	0.2709	0.2729
CD	NDCG@10	0.1950	0.1960	0.2017	0.2112
	HR@10	0.3302	0.3383	0.3406	0.3646
Movie	NDCG@10	0.1669	0.1774	0.1906	0.2156
	HR@10	0.3071	0.3220	0.3270	0.3673

4.3 Main contributions

I reproduce a classical multi-domain sequential recommendation model. I can use it as baseline for future research. In the meantime, I replace RNN with transformer, and achieve better performance.

5 Results and analysis

To evaluate the effectiveness of two methods: supplementary loss and behavior regularizer. I implement experiments in for model variants:

- GRU_Rec: The GRU is applied to recommendation systems without supplement loss and behavior regularizer.
- GRU_Rec_sl: The GRU is applied to recommendation systems without supplement loss and with behavior regularizer.
- GRU_Rec_sl_br: The GRU is applied to recommendation systems with supplement loss and behavior regularizer. It equals DS-HRNN.
- Transformer_Rec_sl_br: The GRU in the above 3-th mentioned variants is replaced by Transformer. The supplement loss and behavior regularizer are saved.

The supplement loss λ_s is set as 0.1 for GRU4Rec_sl. The λ_s and λ_r is searched for $\{0.001, 0.01, 0.1\}$. In the same time, we search the parameter λ_s in $\{0.001, 0.01, 0.1\}$ and λ_r in $\{0.001, 0.01, 0.1\}$ for DS_HRNN and Transformer_Rec_sl_br. The results is shown in Table 2. From the results in Table 2, we can get the following conclusions. First, the two techniques including the supplement loss and regularizer are useful. Second, the transformer is more efficient than RNN in modeling the sequential transition.

6 Parameter sensitivity

To analyze the parameter sensitivity, we search the parameter λ_s in $\{0.001, 0.01, 0.1\}$ and λ_r in $\{0.001, 0.01, 0.1\}$ for DS_HRNN. The results is shown in The results in terms of HR@10 are shown in table 3. From the result, we can conclude that the right parameter is important for the model. For different dataset, the proper parameters are different. For λ_s and λ_r, if they are too large or too small, the performance of the model will deteriorate.

Table 3. Parameter Sensitivity

Domain	Books			CD			Movie		
λ_r \ λ_s	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
0.1	0.2589	0.2512	0.2626	0.3288	0.3053	0.3281	0.3280	0.3139	0.3287
0.01	0.2640	0.2709	0.2570	0.3403	0.3407	0.3259	0.3220	0.3270	0.3180
0.001	0.2539	0.2712	0.2657	0.3384	0.3434	0.3411	0.3138	0.3190	0.3203

7 Conclusion and future work

By reproduce the DS-HRNN model. The results of the experiment demonstrate the supplement loss and behavior regularizer can improve the performance of the model. It proves that the two methods solve the disconnection between behaviors in the same domain due to domain switches. In the meantime, we replace RNN with Transformer. The model with the transformer shows better performance. It proves the superiority of the transformer.

References

[1] Aitor Almeida and Gorka Azkune. Predicting human behaviour with recurrent neural networks. *Applied Sciences*, 8(2):305, 2018.

[2] Trapit Bansal, David Belanger, and Andrew McCallum. Ask the gru: Multi-task learning for deep text recommendations. In *proceedings of the 10th ACM Conference on Recommender Systems*, pages 107–114, 2016.

[3] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 46–54, 2018.

[4] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 152–160, 2017.

[5] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 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*, pages 278–288, 2015.

[6] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[8] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

[9] Donghyun Kim, Sungchul Kim, Handong Zhao, Sheng Li, Ryan A Rossi, and Eunyee Koh. Domain switch-aware holistic recurrent neural network for modeling multi-domain user behavior. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 663–671, 2019.

[10] Bin Li, Qiang Yang, and Xiangyang Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th annual international conference on machine learning*, pages 617–624, 2009.

[11] Pengfei Liu, Xipeng Qiu, and Xuan-Jing Huang. Deep multi-task learning with shared memory for text classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 118–127, 2016.

[12] Qiang Liu, Shu Wu, and Liang Wang. Multi-behavioral sequential prediction with recurrent log-bilinear model. *IEEE Transactions on Knowledge and Data Engineering*, 29(6):1254–1267, 2017.

[13] Andriy Mnih and Geoffrey Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648, 2007.

[14] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. Sequence-aware recommender systems. *ACM computing surveys (CSUR)*, 51(4):1–36, 2018.

[15] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 130–137, 2017.

[16] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 565–573, 2018.

[17] Zitao Xu, Weike Pan, and Zhong Ming. A multi-view graph contrastive learning framework for cross-domain sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 491–501, 2023.

[18] Chunfeng Yang, Huan Yan, Donghan Yu, Yong Li, and Dah Ming Chiu. Multi-site user behavior modeling and its application in video recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 175–184, 2017.