

# 对跨域序列推荐中的工作 TPUF 的复现与改进

## 摘要

推荐系统通过对用户与物品或其他信息的历史交互记录的建模有效地挖掘和学习用户的真实偏好，进而为用户提供精准的个性化推荐服务。本文主要是针对跨域序列推荐算法中的 TPUF 算法进行复现，该算法不仅能够有效的缓解数据稀疏性的问题，也较好的解决了在传统跨域序列推荐中，用户的隐私保护的问题。本文首先对跨域序列推荐的背景和相关工作进行介绍，随后介绍本文复现的工作 TPUF，并在此基础上提出增加时间感知模块。使其不仅可以对交互的顺序进行建模，还能够对交互之间的时间间隔进行建模，最后三个跨域数据集上证明本文复现结果的准确性以及改进方法的有效性。

**关键词：**推荐系统；跨域序列推荐；

## 1 引言

随着移动互联网的发展，各式各样的信息服务如潮水般向人们涌来。人们在享受这些互联网平台所带来的便捷生活的同时也面临着信息过载的问题，人们早已从一个信息匮乏的时代进入到一个信息爆炸的时代。对于信息过载的问题，目前已经有许多的解决方案，但其中较为常见且有效的两种解决方案就是搜索引擎和推荐系统。其中搜索引擎主要针对的是具有明确需求的用户，用户可以通过输入词条的方式来快速获取自己想要的信息。但如果当用户无法准确描述自己的需求时，搜索引擎就难以提供有效的服务。在用户需求不明确的情况下，如何从海量的信息中帮助用户寻找所需的产品并提供个性化的服务，已然成为当前研究的热点问题之一。

然而随着推荐系统的出现，上述问题有了切实可行的解决方案。推荐系统是一种根据用户历史行为数据来预测用户潜在偏好的一种算法。用户无需提供准确的关键词，就能获取潜在感兴趣的内容。推荐系统主要通过分析用户的历史行为数据，例如浏览记录、评分记录、购买记录等主动地为用户推送相关信息与服务，帮助用户缩小目标范围，使其能快速寻找到相关内容，提高用户的使用体验，促进相应指标的提升。推荐系统在满足用户个性化需求的同时也为企业带来了较大的经济价值。

传统的推荐系统通常依赖于静态的用户物品交互，因此很难捕捉用户偏好的动态变化。因此，在用户偏好随时间演变的情况下，准确提供个性化推荐变得具有挑战性。相比之下，序列推荐 (SR) 方法旨在通过捕获时间用户的动态兴趣的变化来解决这个问题。他们通过考虑用户的历史交互序列来预测用户的偏好。其中序列推荐中最具代表性的工作就是利用注意力机制来进行建模，它能够有效识别交互之间的顺序依赖关系并精确定位最相关的项目来进行

推荐。尽管近年来随着技术的不断发展，序列推荐取得了较大的进展，但序列推荐模型仍然面临着数据稀疏性的挑战，这阻碍了建立训练序列推荐模型所需的信息交互序列的能力。

为了缓解数据稀疏性，有关学者提出了跨域推荐 (CDR)。跨域推荐的目标是利用来自一个或多个源域中丰富且相关的知识来对目标域进行建模，以达到缓解目标域中数据稀疏的问题。典型的跨域推荐方法是通过学习源域和目标域之间的映射函数来实现的，或者使用带有跨连接单元的深度神经网络来对齐跨域的对象。然而，这些传统的跨域推荐的方法都没有考虑用户在不同域中行为的序列信息，因此不能很好地对用户的动态偏好或行为序列之间的顺序依赖关系进行建模。

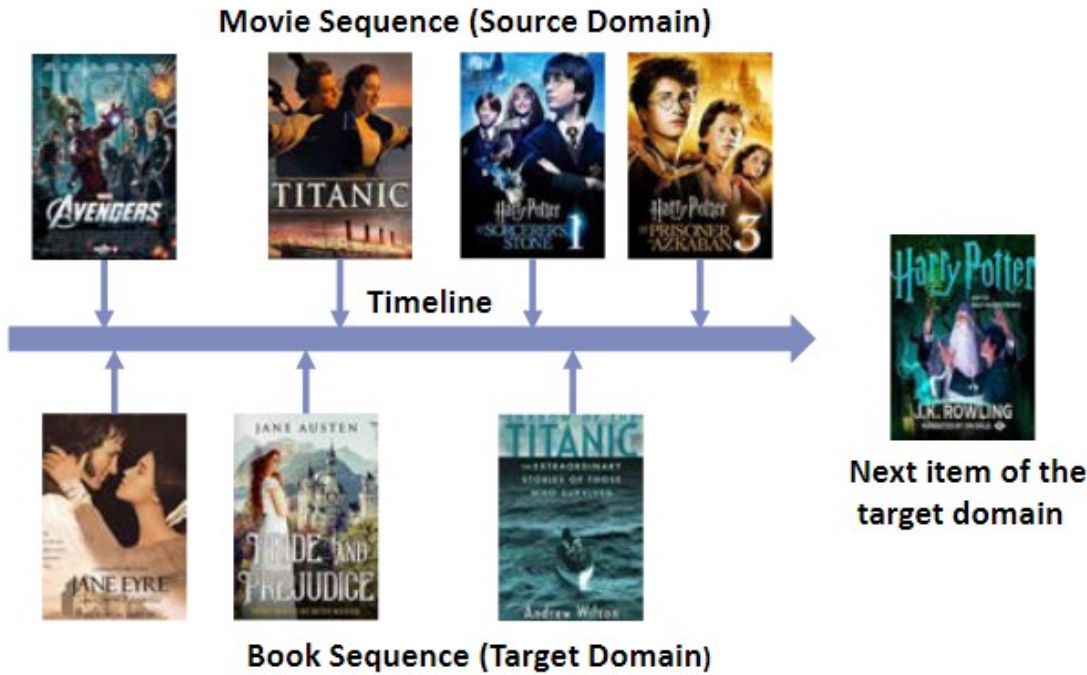


图 1. 跨域序列推荐场景示意图<sup>1</sup>

为了解决上述问题，有关学者提出了跨域序列推荐 (CDSR)。跨域序列推荐广泛存在于现实场景中，如图 1 所示，其中电影领域和书籍领域分别是源域和目标域。如果一个人最近从哈利波特系列观看了几部电影，跨域序列推荐可能会推断在书籍领域他们可能会阅读一些哈利波特的书籍。这种推理仅仅只靠书籍领域的历史交互行为中是无法得到的。现有的研究已经证明了从其他领域转移信息的有效性，使得具有稀疏数据的目标域推荐性能显着提高。

## 2 相关工作

本节将从以下四个部分简要介绍一下跨域序列推荐算法的相关工作。它们分别是传统的推荐算法、序列推荐算法、跨域推荐算法以及跨域序列推荐算法。

<sup>1</sup><https://doi.org/10.1145/3583780.3615094>

## 2.1 传统的推荐算法

传统推荐算法 [5,13] 大多数是通过填充 (用户、项目) 对的评分矩阵来处理一般的推荐任务, 并没有利用序列和行为信息。到目前为止, 传统推荐算法的研究工作已经相当成熟。在早期有如以因子分解机 (FM) [12] 为代表的基于内容的推荐算法, 该方法首先对用户的交互行为特征进行提取, 将其转换为一个行向量, 随后拼接成一个矩阵再进行因式分解, 最后通过用户编号信息与待预测的项目进行匹配的方式进行推荐; 以改进的奇异值分解 (SVD++) [7] 为代表的基于协同过滤的推荐算法, 该算法是在奇异值分解 [8] 算法的基础上进行改进, 同时从两个不同的角度进行建模, 既考虑了用户对物品偏好的信息, 又考虑了用户与物品交互的行为信息, 开创性的将显示反馈建模和隐式反馈建模无缝融合, 还通过扩展预测公式的方式来融合其他相关信息, 为建模提供了一个重要的思路。

## 2.2 序列推荐算法

序列推荐中最早的算法是基于马尔科夫链模型 [3], 它主要捕获的是项目与项目之间的共性关系。随着深度学习的发展, 一大批优秀的基于深度学习的序列推荐算法被提出, 其中包括基于循环神经网络的 GRU4Rec [4]、基于卷积神经网络的 Caser [14] 以及基于图神经网络的 SR-GNN [15]。其中基于 Transformer 的模型因其处理稀疏数据和建模长期用户兴趣的能力而受到广泛的欢迎 [6,9]。SASRec [6] 是其中的代表性的算法, 该算法通过引入自注意力机制来建模, 采用位置编码函数来确定交互序列中商品的位置关系, 计算交互序列中每个物品的贡献值, 最后将得到的所有信息与所有物品的 embedding 进行内积, 计算相关性并对其进行排序, 选取 Top-k 个物品进行推荐, 后续许多研究都是将其与其变体进行了扩展。例如, 有关学者提出, 传统的基于 SASRec [6] 的推荐算法, 大多只考虑了物品之间的先后顺序关系, 并没有考虑时间间隔信息, 隐含的假设是所有物品之间的交互的时间间隔是等价的, 即只考虑了物品之间的绝对位置关系。然而在现实场景中, 一天之内产生的序列和一个月之内产生的序列所进行的推荐, 是不等价的, 这较长的时间跨度内, 用户的兴趣会发生较大的偏移。受到自注意力+相对位置编码的启发。有关学者提出了 TiSASRec [9], 在 Self-Attention 的基础上加入绝对位置和相对时间间隔信息, 该算法在稠密和稀疏的数据集上都取得了 SOTA 的性能, 证明了使用相对时间间隔的有效性。

## 2.3 跨域推荐算法

为了解决数据稀疏性和冷启动的问题, 有关学者提出了跨域推荐算法。它通过从源域提取知识, 并将其迁移到目标域中, 以达到提高目标域的推荐性能的目的。在跨域推荐中, 最重要的问题就是确定不同域之间, 知识如何迁移。传统的跨域推荐中, 知识的传递大多是基于映射的方法, 它对两个域之间的联系进行建模, 并通过显示学习的方式, 学习映射函数。例如, EMCDDR [11] 从两个域的对齐特征中学习映射函数, 而 DCDCSR [17] 则通过考虑评级的稀疏程度, 为映射函数生成基准因子矩阵。

## 2.4 跨域序列推荐算法

跨域序列推荐 (CDSR) 通过利用来自不同领域的序列信息来提高推荐性能。早期的工作, 大多通过采样循环神经网络的方式来建模两个域之间的信息。例如,  $\pi$ -net [10] 设计了一

个跨域传输单元来提取共享用户在每个时间戳处两个域之间的信息。后续研究中，也有部分学者采用了图神经网络来学习跨域用户兴趣。例如，C2DSR 采样了图神经网络来提取序列间的信息，并采样注意力机制来捕获序列内的信息 [1]。

### 3 本文方法

#### 3.1 本文方法概述

本文主要复现的工作是于 2023 年发表在 CIKM 会议上的一篇论文 “TPUF: Enhancing Cross-domain Sequential Recommendation via Transferring Pre-trained User Features” [2]。该算法首先对给定源域中预训练的用户潜在特征向量进行特征提取，利用源域中的用户特征和目标域中的顺序交互来预测目标域中的下一个项目。为了对齐源域与目标域的交互序列，该算法通过时间特征映射单元 (TFM)，从预训练的 UFS 中提取时域信息，并将其转换为时域特定的用户特征。然而由于预训练的用户潜在特征，仅使用来自于源域的数据进行训练，它可能表现出对某些类型的数据的偏见。因此本文通过对抗性特征对齐单元 (AFA)，对齐两个域之间的中间特征，使得来自两个域之间的中间特征的分布更为相似，以提高推荐质量。

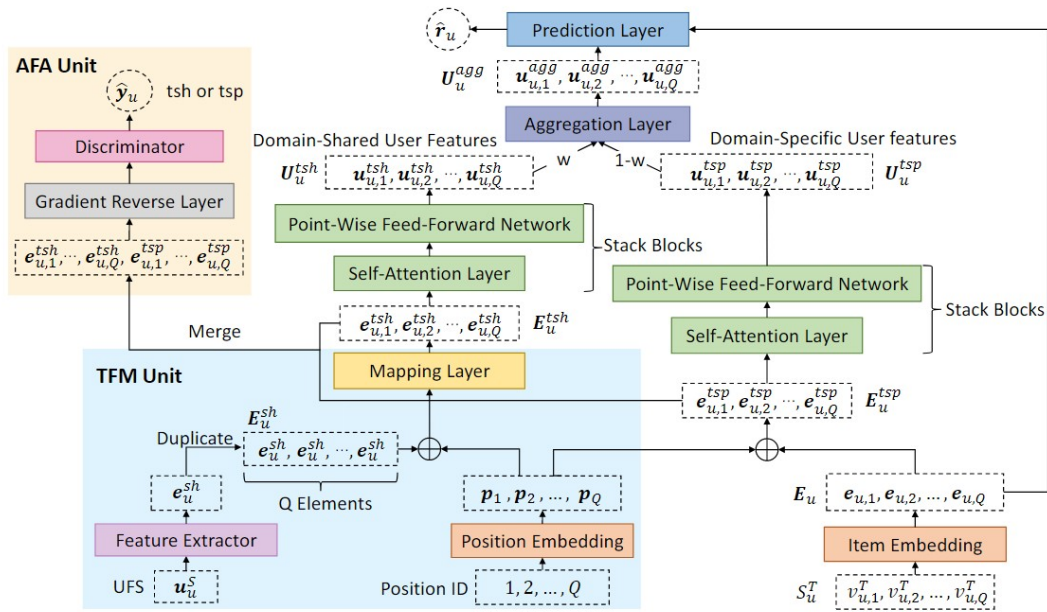


图 2. TPUF 示意图<sup>2</sup>

#### 3.2 目标域的用户特征提取模块

##### 3.2.1 嵌入层

假定对于目标域的用户  $u$  而言，最近交互的  $Q$  个物品的交互序列为  $S_u^T = (v_{u,1}^T, \dots, v_{u,Q}^T)$ 。如果交互序列的长度小于  $Q$ ，本文将用 0 对其进行填充，使其序列长度为  $Q$ 。随后我们对目标域中的交互序列  $S_u^T$  进行编码，得到相应的特征向量矩阵  $E_u = \{e_{u,1}, \dots, e_{u,Q}\}$ ，其中  $E_u \in \mathbb{R}^{Q \times d}$ ， $Q$

<sup>2</sup><https://doi.org/10.1145/3583780.3615094>

为序列长度,  $d$  为嵌入向量的维度。随后, 我们将位置编码信息  $P \in \mathbb{R}^{Q \times d}$  合并至序列的特征编码  $E_u$  中, 得到融合了时域特征的特征编码  $E_u^{tsp}$ , 其计算公式如下:

$$E_u^{tsp} = E_u + P \quad (1)$$

### 3.2.2 自注意力块

本文利用自注意力模块来对序列  $E_u^{tsp}$  中的每个元素进行建模, 将其输入至自注意力块  $\{B^k | k = 1, 2, \dots, n\}$  中, 获取用户在目标域中的潜在兴趣偏好  $U_u^{tsp}$ , 该模块主要由自注意力层  $SA(\cdot)$  和逐点前馈网络层  $FFN(\cdot)$  构成, 为了缓解由自注意力块堆叠而引起的过拟合和梯度消失的问题, 本文在自注意力块中的每一层中都进行了归一化、dropout 以及残差连接的操作, 具体可表示为:

$$SA(X) = softmax\left(\frac{X \cdot W^Q \cdot (X \cdot W^K)^T}{\sqrt{d}}\right) X \cdot W^V \quad (2)$$

$$B^k(X) = FFN(SA(X)), k = 1, 2, \dots, n \quad (3)$$

$$F_u^{tsp} = B^n(\dots(B^1(E_u^{tsp}))) \quad (4)$$

$$G(F_u^{tsp}) = F_u^{tsp} + Dropout(g(LayerNorm(F_u^{tsp}))) \quad (5)$$

其中  $W^Q, W^K, W^V \in \mathbb{R}^{d \times d}$  是可学习的参数,  $g(\cdot)$  表示块中的  $SA$  或者  $FFN$  层,  $G(\cdot)$  是每一层的最终输出, 最后将最后一个块的输出作为用户在目标域中的潜在兴趣偏好  $U_u^{tsp} = (u_{u,1}^{tsp}, u_{u,2}^{tsp}, \dots, u_{u,Q}^{tsp})$

### 3.3 源域的用户特征提取模块

为了保护用户的隐私, 本文采用的是在源域中通过对数据进行预训练的方式, 获得源域中预训练模型, 并将该模型送至目标域中, 在目标域中使用特征提取模块  $F$ , 从源域中获取用户的特征, 其中  $F$  是一个参数为  $\Theta_f$  的多层神经网络, 它从用户  $u_u^S$  中提取共享特征  $e^{sh}$ :

$$e^{sh} = F(u_u^S; \Theta_f) \quad (6)$$

随后将共享域的特征  $e^{sh}$  重复  $Q$  次, 生成共享域特征矩阵  $E_u^{sh} \in \mathbb{R}^{Q \times d}$ , 然后将该矩阵域目标域中的位置嵌入矩阵  $P$  进行融合, 并将融合后的结果输入到一个参数为  $\Theta_m$  的全连接层  $M(\cdot)$  中, 得到  $E_u^{tsh}$ :

$$E_u^{tsh} = M(E_u^{sh} + P; \Theta_m) \quad (7)$$

最后将  $E_u^{tsh}$  输入到自注意力块中, 合并每个时间步的  $E_u^{tsh}$  元素, 然后使用残差网络等方式来缓解梯度消失以及过拟合的问题。并将最终的自注意力块的输出用作共享域中用户的潜在特征, 表示为  $U_u^{tsh}$ :



$$SA(X) = softmax \left( \frac{X \cdot W^Q \cdot (X \cdot W^K)^T}{\sqrt{d}} \right) X \cdot W^V \quad (8)$$

$$B^k(X) = FFN(SA(X)), k = 1, 2, \dots, n \quad (9)$$

$$F_u^{tsh} = B^n(\dots(B^1(E_u^{tsh}))) \quad (10)$$

$$G(F_u^{tsh}) = F_u^{tsh} + Dropout(g(LayerNorm(F_u^{tsh}))) \quad (11)$$

其中  $W^Q, W^K, W^V \in \mathbb{R}^{d \times d}$  是可学习的参数,  $\{B^k | k = 1, 2, \dots, n\}$  表示自注意力块,  $g(\cdot)$  表示块中的  $SA$  或者  $FFN$  层,  $G(\cdot)$  是每一层的最终输出, 用户在共享域中的潜在兴趣偏好  $U_u^{tsh} = (u_{u,1}^{tsh}, u_{u,2}^{tsh}, \dots, u_{u,Q}^{tsh})$

### 3.4 特征聚合与预测层

最后我们将目标域和共享域的特征, 通过加权聚合的方式合并:

$$U_u^{agg} = w \cdot U_u^{tsh} + (1 - w) \cdot U_u^{tsp} \quad (12)$$

其中  $w$  是一个超参数, 用于控制源域到目标域中特征迁移的比例。  $U_u^{agg}$  是合并后的用户的特征表示。最后将合并后的用户的特征向量  $u_{u,q}^{agg} \in U_u^{agg}$  与用户下一步  $q+1$  的候选物品的嵌入  $e'_{u,q+1}$  进行内积, 得到用户候选物品的偏好:

$$\hat{r}_u^{q+1} = \sigma(u_{u,q}^{agg} \odot e'_{u,q+1}) \quad (13)$$

其中  $\odot$  表示内积,  $\hat{r}_u^{q+1}$  表示用户  $u$  在第  $q+1$  个时间步内, 与候选物品交互的概率。  $\sigma$  表示 sigmod 的激活函数。

### 3.5 对抗性特征对齐单元

由于不同域之间的特征分布可能存在显著的差异, 为了缓解这种数据偏差, 对齐两个域之间的中间特征。此模块通过一个对抗性特征对齐单元来对齐两个域的特征, 其中本文使用了鉴别器  $D$  作为多层神经网络层, 来对共享域和目标域的中间特征进行分类, 具体表示如下:

$$\hat{y}_u^i = D(q_{u,i}; \Theta_d) \quad (14)$$

其中  $q_{u,i} \in E_u^{tsh} \cup E_u^{tsp}$ ,  $\Theta_d$  是鉴别器  $D$  的参数集合,  $\hat{y}_u^i \in [0, 1]$ 。

## 4 复现细节

### 4.1 与已有开源代码对比

本篇论文的作者公布了算法相应的源代码 [2], 本次复现工作参考了作者发布的源代码, 采用了上述论文中相同的数据集 (代码中已提供), 以确保复现算法的准确性。实验结果与

论文报告中的结果较为一致。由于作者没有提供数据集中数据的处理方式，且原论文数据中没有包含每个物品的交互时间戳信息，为了便于后续对算法进行改进，本文使用了实验室内部的数据处理方式，并在数据中添加了时间戳信息。不仅如此，本文还更换了负采样的方式，原论文使用的是随机采样，本文替换成了基于流行度的采样。

由于，现有的跨域序列推荐模型所做的一个简化假设是将交互历史视为有序序列，而不考虑每个交互之间的时间间隔（也就是说，他们建模的是时间顺序，而不是实际的时间戳）。但在真实的应用场景中，用户在不同时间对于不同物品的关注度是不同的。如果能够充分利用用户与物品之间、用户与时间和物品与时间之间的关系，从多方面捕捉用户的兴趣，可以取得更好的推荐效果。所以，基于时间感知的跨域序列推荐的研究非常具有实际的研究价值。因此，本文除了在对最新的跨域序列推荐算法 TPUF 进行复现以及更换数据集之外，还考虑在原算法的基础上交互之间的时间间隔信息，以此来提高算法性能。

受到 TiSASRec [9] 的启发，本文尝试在 TPUF 算法的自注意力块中添加相对时间间隔信息。具体而言，首先本文先计算了每个用户的交互序列中，任意两个物品之间的时间间隔，并得到相应的时间关系  $r_{ij}$ ，所有的时间间隔关系构成时间间隔集合  $R^u$ 。随后设定缩放系数  $r_{min}^u = \min(R^u)$ ，即序列中最小的时间间隔信息，然后再对所有的时间间隔进行缩放  $r_{ij}^u = \lfloor \frac{|r_i - r_j|}{r_{min}^u} \rfloor$ ，得到时间间隔矩阵  $M^u \in N^{n \times n}$ 。然后用两个  $K, V$  矩阵  $E_K^P \in R^{k \times d}$  和  $E_V^P \in R^{k \times d}$  来表示每个位置的 Key 和 Value 的值，其中  $k$  表示有  $k$  种相对的时间间隔， $d$  表示维度信息。随后把时间间隔矩阵  $M^u$  中的  $r_{ij}$  替换成对应的  $K, V$  向量，即  $r_{ij}^v$  和  $r_{ij}^k$ 。 $r_{ij}^v$  和  $r_{ij}^k$  又分别构成关于  $K, V$  的时间间隔关系嵌入矩阵  $E_K^R \in R^{n \times n \times d}$  和  $E_V^R \in R^{n \times n \times d}$ 。对于目标域中修改后的用户特征提取模块的计算过程如下所示：

$$TiSA(X) = softmax \left( \frac{X \cdot W^Q \cdot (X \cdot W^K + r_{ij}^k)^T}{\sqrt{d}} \right) (X \cdot W^V + r_{ij}^v) \quad (15)$$

$$B^k(X) = FFN(TiSA(X)), k = 1, 2, \dots, n \quad (16)$$

$$F_u^{tsp} = B^n(\dots(B^1(E_u^{tsp}))) \quad (17)$$

$$G(F_u^{tsp}) = F_u^{tsp} + Dropout(g(LayerNorm(F_u^{tsp}))) \quad (18)$$

其中  $W^Q, W^K, W^V \in \mathbb{R}^{d \times d}$  是可学习的参数， $\{B^k | k = 1, 2, \dots, n\}$  表示自注意力块， $g(\cdot)$  表示块中的  $TiSA$  或者  $FFN$  层， $G(\cdot)$  是每一层的最终输出，最后将最后一个块的输出作为用户在目标域中的潜在兴趣偏好  $U_u^{tsp} = (u_{u,1}^{tsp}, u_{u,2}^{tsp}, \dots, u_{u,Q}^{tsp})$ 。

## 4.2 实验

本实验采用的实验环境是 Python 3.7.12 和 Pytorch 1.13.0，模型的训练使用了 Tesla V100 PCIe GPU with 32GB memory。接下来将详细介绍实验所采用的数据集、评估指标、参数设置以及本次复现工作的创新点。

#### 4.2.1 数据集及预处理

本实验采用的是 Amazon 的评论数据。我们从 Amazon 的数据中选择了三个不同类别的数据，分别为 Movie、CD 和 Book。参照经典算法 SASRec [6] 的实验设置，我们对数据集进行了预处理，并做了如下假设：a) 假设评论和购买行为是正反馈数据，并通过时间戳来确定其交互顺序。b) 我们只保留用户与物品交互数不少于五个的物品，并且剔除掉后续重复的 (用户, 物品) 对。c) 只保留在所有域中都有交互序列的用户。d) 采用留一法对实验进行评估，即将每个序列分成三个部分，测试集由每条序列的最后一个交互构成，验证集由每条序列倒数第二个交互构成，其余部分用于训练。数据集具体细节如表 1 所示。

表 1. 预处理后数据集情况描述

数据集	跨域用户数	物品数	交互数	序列平均长度	稠密度
Movie	10929	60902	462314	42.30	0.07%
CD		94171	348746	31.91	0.03%
Book		242363	615912	56.36	0.02%

#### 4.2.2 评估指标

由于本次实验所研究的任务属于推荐系统中物品排序的任务，为了验证本文复现算法以及改进后的算法的有效性，此次实验将通过命中率 (**HR@N**) 和归一化折损累积增益 (**NDCG@N**) 这两个推荐系统中常用的指标来评估 Top-N 的推荐性能，下面将对其进行详细的介绍。

**命中率 (Hit Rate, HR)**: 该指标反映的是在推荐列表中，用户真实交互过的物品的数量与推荐列表中物品数量的一个比值，具体计算公式如下：

$$HR@N = \frac{1}{N} \sum_{i=1}^N hit(i) \quad (19)$$

其中， $N$  表示推荐列表的长度， $hit(i)$  为指示函数，表示推荐物品是否命中，如果推荐物品属于用户真实交互过的物品则为 1，否则为 0。

**归一化折损累积增益 (normalized discounted cumulative gain, NDCG)**: 该指标反映的是推荐的排序列表与用户真实交互过的物品列表之间的差距，即在推荐列表中用户真实交互过的物品排名越靠前，那么该物品所获得的评估分也就越高，排名越靠后的物品所获得的评估分越低。基于此对于用户  $u$ ，归一化折损累积增益具体可以定义为如下形式：

$$DCG_u@N = \sum_{\ell=1}^N \frac{2^{\delta(i(\ell)I_u^{te})} - 1}{\log(\ell + 1)} \quad (20)$$

$$NDCG_u@N = \frac{1}{Z_u} DCG_u@N \quad (21)$$

其中， $\delta(x)$  为指示函数，当且仅当  $x$  为真时  $\delta(x) = 1$ ，否则  $\delta(x) = 0$ 。 $i(\ell)$  表示该推荐列表中第  $\ell$  位的物品， $\delta(i(\ell) \in I_u^{te})$  表示推荐列表  $I_u^{te}$  中第  $\ell$  位的物品是否是用户真实交互过的物品， $Z_u$  表示最优的  $DCG_u@N$  的得分，即对应真实交互列表  $I_u^{te}$  中的物品恰好出现在推荐



列表  $I_u^{re}$  的开头的情况。然而，对于整个测试用户数据集而言，归一化折损累积增益可定义为如下形式：

$$NDCG@N = \frac{1}{|U^{te}|} \sum_{u \in U^{te}} NDCG_u@N \quad (22)$$

上述两个指标分别从两个不同的角度来评估推荐模型的性能。命中率通过比较在推荐列表中用户真实交互过的物品的占比的大小来进行衡量模型的性能，而归一化折损累积增益则进一步考虑了用户真实交互过的物品之间的顺序问题来衡量模型的性能，排名越靠前，分数越高，反之分数也就越低。因此上述两个指标的值越大，模型的性能也就越好。

#### 4.2.3 参数设置

本次实验采用了两个不同的数据处理方式，复现过程中的实验设置与原论文一致，改进后的实验参数设置，与前人研究的一致 [16]，序列长度为 100，batch size 为 128，hidden\_units 为 50，dropout 率为 0.2，学习率为 0.001，time\_span 为 256，combine\_weight 为 0.5，loss\_weight 为 0.9。

#### 4.3 创新点

本次实验的创新点在于，在跨域序列推荐的基础上考虑交互之间的时间间隔信息。通过对目标域中的自注意力块进行改进，融合了时间间隔信息，使得其推荐性能大幅提升，性能远超改进前的跨域序列推荐算法 TPUF [2] 以及单域中基于时间感知的序列推荐算法 TiSASRec [9]。

### 5 实验结果分析

为了验证复现算法的准确性，本文在复现阶段，采用的是原论文的数据集进行建模。在表 2 中，Original 表示原论文中报告的结果，Repro 表示复现的结果。由表 2 可知，SASRec(Original) 和 SASRec(Repro) 以及 TPUF(Original) 和 TPUF(Repro) 在 HR@10 和 NDCG@10 上结果相差都小于 0.01，这表明本次实验成功复现了 SASRec 以及 TPUF。

由于原论文并未给出数据集的处理方式。因此，为了便于后续对算法进行改进，我们更换了实验数据集以及数据处理的方式，对于用户的每一个交互记录都保留了时间戳信息，以便后续对时间进行建模。为了同时验证跨域和时间的有效性，本文还复现了 TiSASRec [9] 算法，用于检验改进后的模型在考虑时间情况下，是否能够优于经典的基于时间间隔感知的算法，和其他实验形成互补。由表 3 可知，TiSASRec 算法在三个数据集上的性能都远胜单域中的 SASRec 算法以及跨域序列推荐中的 TPUF 算法，这说明时间间隔这一因素是十分有效的，而改进后的 TPUFpp 算法在 HR@5、HR@10、NDCG@5 和 NDCG@10 上都达到了最优，且性能都远超其他算法，这说明了算法同时考虑跨域和时间间隔信息能够很好的提升模型的性能，也进一步论证了改进后算法的有效性。

表 2. 原论文数据集实验结果

Datasets	Metric	Single-Domain Method		Cross-Domain Method	
		SASRec(Original)	SASRec(Repro)	TPUF(Original)	TPUF(Repro)
Douban Movies $\rightarrow$ Books	HR@5	0.3267	0.3385	0.3773	0.3667
	NDCG@5	0.2346	0.2455	0.2675	0.2582
	HR@10	0.4568	0.4556	0.5156	0.5144
	NDCG@10	0.2767	0.2834	0.3122	0.3057
Amazon Movies $\rightarrow$ Books	HR@5	0.1198	0.1306	0.1572	0.1532
	NDCG@5	0.0836	0.0888	0.1160	0.1129
	HR@10	0.1812	0.1845	0.2144	0.2161
	NDCG@10	0.1034	0.1062	0.1344	0.1332
Amazon Sports $\rightarrow$ Clothing	HR@5	0.1865	0.1904	0.2015	0.2021
	NDCG@5	0.1580	0.1623	0.1665	0.1642
	HR@10	0.2265	0.2333	0.2545	0.2574
	NDCG@10	0.1709	0.1761	0.1835	0.1820

表 3. TPUF 算法改进后的实验结果

Datasets	Metric	Single-Domain Method		Cross-Domain Method	
		SASRec	TiSASRec	TPUF	TPUFpp
Amazon Books $\rightarrow$ Movies	HR@5	0.2146	0.2938	0.2127	0.3741
	NDCG@5	0.1466	0.2100	0.1432	0.2703
	HR@10	0.3061	0.4089	0.3102	0.4969
	NDCG@10	0.1760	0.2471	0.1745	0.3100
Amazon Books $\rightarrow$ CDs	HR@5	0.2368	0.2686	0.2307	0.3686
	NDCG@5	0.1671	0.1806	0.1584	0.2708
	HR@10	0.3391	0.3928	0.3325	0.4747
	NDCG@10	0.2001	0.2205	0.1912	0.3051
Amazon Movies $\rightarrow$ Books	HR@5	0.1609	0.2381	0.1645	0.2775
	NDCG@5	0.1077	0.1620	0.1090	0.2077
	HR@10	0.2396	0.3433	0.2511	0.3549
	NDCG@10	0.1329	0.1960	0.1368	0.2327

## 6 总结与展望

本文首先对跨域序列推荐的背景以及相关工作进行了简要介绍，随后对复现的跨域序列推荐算法 TPUF 进行了介绍，详细介绍了目标域的用户特征提取模块、源域的用户特征提取模块、特征聚合与预测层以及对抗性特征对齐单元。随后本文基于 Pytorch 对该算法进行了复现（作者也已提供了源代码）并提出了在目标域中的自注意力块中增加时间间隔感知的模块。最后本文在原论文数据集以及更换数据处理方式的 Amazon 的三个跨域数据集上进行实验，实验结果验证了复现结果的准确性以及改进后算法的有效性。

虽然本次提出的改进方案，在性能上有较大的提升，但是受限于算法的建模方式，此次改进无法直接同时考虑两个域的交互行为的时间间隔信息。因此后续，将沿着此次复现的工作，对算法将进行进一步的实验与改进，即同时考虑时间间隔在两个不同域之间的影响，以达到更好的推荐性能。

## 参考文献

- [1] Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. Contrastive cross-domain sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 138–147, 2022.
- [2] Yujia Ding, Huan Li, Ke Chen, and Lidan Shou. Tpuf: Enhancing cross-domain sequential recommendation via transferring pre-trained user features. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 410–419, 2023.
- [3] Ruining He and Julian McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 191–200. IEEE, 2016.
- [4] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.
- [5] Santosh Kabbur, Xia Ning, and George Karypis. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 659–667, 2013.
- [6] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.
- [7] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.
- [8] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [9] Jiacheng Li, Yujie Wang, and Julian McAuley. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*, pages 322–330, 2020.
- [10] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke.  $\pi$ -net: A parallel information-sharing network for shared-account cross-domain sequential recommendations. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 685–694, 2019.
- [11] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. Cross-domain recommendation: An embedding and mapping approach. In *IJCAI*, volume 17, pages 2464–2470, 2017.

- [12] Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):1–22, 2012.
- [13] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.
- [14] Jiayi 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.
- [15] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 346–353, 2019.
- [16] 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.
- [17] Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, Mehmet Orgun, and Jia Wu. A deep framework for cross-domain and cross-system recommendations. *arXiv preprint arXiv:2009.06215*, 2020.