

赞同 39

分享

2024腾讯：跨域推荐中用户兴趣和正负反馈建模



SmartMindAI

专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我

已关注

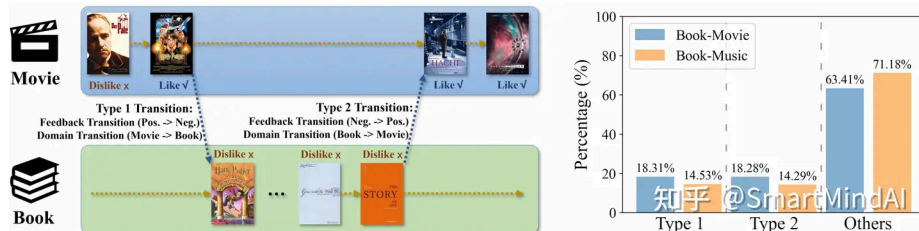
39 人赞同了该文章

收起

Introduction

传统的顺序推荐模型在单一领域内，如音乐或视频，对用户的历史行为进行建模。然而，现代推荐系统⁺往往涵盖了多个领域，如书籍和电影，导致用户在不同领域之间进行切换。在这个切换过程中，不仅领域切换的信息（称为“领域切换”）有价值，用户在领域切换期间对推荐项目偏好的变化（称为“反馈切换”）也同样关键。

我们将用户在领域切换期间的反馈变化分为两种类型：（1）类型1切换：在跨领域切换时，用户反馈从正面变为负面；（2）类型2切换：在浏览新领域时，用户反馈从负面变为正面。例如，如图所示，对于“类型1”的切换，用户在喜欢的电影后可能会搜索相关书籍。然而，由于电影是根据原著改编的，直接推荐原著书籍可能会被用户不喜欢，因为她已经看过电影了。在这种情况下，理想的推荐应该是那本书的续作。对于“类型2”的切换，用户对连续推荐的几本书感到不满意，转而观看电影。这也表明，当用户在某一领域内持续对项目不满意时，推荐来自另一个领域的项目可能会有好处。



我们通过分析Douban数据中所有切换中这两种类型切换的比例，如图所示。具体来说，我们分析了书籍-电影和书籍-音乐之间的跨领域行为，关注“类型1”和“类型2”切换的计数。我们发现这两种切换类型在书籍-电影的跨领域行为中占了18.31%和18.28%，在书籍-音乐的跨领域行为中占了14.53%和14.29%。这一发现强调了建模跨领域行为和反馈切换的重要性。现有的工作通常侧重于建模用户在不同领域的切换行为，以提高跨领域顺序推荐的性能。其中一项开创性工作通过一个门控转移模块将从单一领域学习的表示转移到其他领域。

近期的研究分别建模单个领域和跨领域序列，并通过自我注意力机制⁺生成表示。尽管现有工作的效果很好，但它们忽略了用户在跨领域行为期间的反馈切换的影响，往往只关注用户的正面反馈。为了建模用户在跨域行为中的领域转换和反馈转换，本文提出了一种名为Transition²的方法，该方法用于跨域序列推荐中的领域和反馈转换建模。

首先，我们使用感知转换的图编码器编码用户在不同领域的混合历史。具体来说，在构建图时，我们连接历史中的连续项目，并根据两个项目之间的反馈为边分配不同的权重。这使得我们能够捕捉不同领域和反馈类型的转换。在图编码器编码后，我们获得了一种捕获不同领域和反馈类型转换的项目嵌入。然后，我们将这些历史嵌入输入到多头自注意力中，进一步对用户历史中的转换信息进行建模。为了建模不同领域和不同反馈类型之间的转换，我们向转换器的多头自注意力的头部引入不同的遮罩。这使得每个头部能够捕获不同类型的转换。然后，我们融合输出，得到包含各种转换信息的最终历史表示。最后，我们使用此表示进行在不同领域推荐任务中的预测。

在本文中，我们探讨了一个全面的CDSR场景，其中每个用户的历史记录涉及两个领域，分别为领域 A 和 B 。每个实例 S_A, S_B, S_C 对应特定的用户。对于任何给定的实例 $S_A = [(a_1, y_1), \dots, (a_{|S_A|}, y_{|S_A|})]$ 和 $S_B = [(b_1, y'_1), \dots, (b_{|S_B|}, y'_{|S_B|})]$ 代表单域用户历史记录，其中 y_i 和 y'_i 分别表示与 a_i 和 b_i 关联的反馈，取值为+1（正面反馈）或-1（负面反馈）。

序列 $S_C = [(b_1, y'_1), (a_1, y_1), \dots, (b_{|S_B|}, y'_{|S_B|}), \dots, (a_{|S_A|}, y_{|S_A|})]$ 代表跨域用户历史记录，通过按时间顺序合并 S_A 和 S_B 形成，其中每个 $a \in A$ 和 $b \in B$ 是被交互的项目，而 $|\cdot|$ 表示项目总数。请注意， A 和 B 分别代表领域 A 和领域 B 中的全部项目集。

给定观察到的用户历史记录 $(S_A, S_B, S_C)_u$ ，CDSR的目标是预测下一个项目：

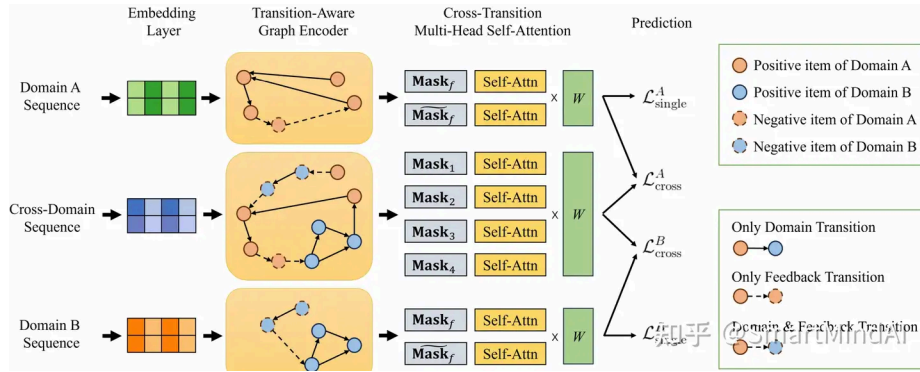
$$\arg \max_{a_i \in A} P_A(a_{|S_A|+1} | S_A, S_B, S_C) \quad \text{if the next item is in } A$$

$$\arg \max_{b_j \in B} P_B(b_{|S_B|+1} | S_A, S_B, S_C) \quad \text{if the next item is in } B$$

在其中， $P_A(a_i | S_A, S_B, S_C) \in \mathbb{R}^{|A|}$ 和 $P_B(b_j | S_A, S_B, S_C) \in \mathbb{R}^{|B|}$ 分别表示域 A 和 B 中候选项目的概率，其中概率最高的项目被选中作为下一个推荐项目。

Transition²: The Proposed Algorithm

Overview of Transition²



我们的模型Transition²的整体架构如图所示。为了有效地捕获领域转换和反馈转换信息：

- (1) **Transition²**首先使用三个嵌入层为两个单一领域序列和交叉领域⁺序列初始化嵌入。
- (2) **Transition²**然后基于用户的交叉领域和单一领域序列构建了转换感知图编码器。与以往仅利用正反馈⁺构建图的方法不同，我们整合了正负反馈⁺，并相应调整了图神经网络的权重矩阵⁺。
- (3) 在通过图编码器获得用户历史序列的表示后，我们设计了交叉转换多头自注意力机制。该机制在用户序列的不同行为之间计算注意力，以捕获转换信息。
- (4) 最后，基于最终序列表示计算预测分数，同时优化单一领域和交叉领域的损失。

Transition-Aware Graph Encoder

受GNN处理序列推荐优势的启发，我们通过GNN将用户的序列行为转换为图表示。为了有效地捕获领域转换和反馈转换，我们引入了具有转换意识的图编码器。

Embedding Initialization

对于序列推荐设置，我们基于所有用户的历史序列构建单域图 G_A 和 G_B ，以及跨域图 G_C ，其中仅包含项目节点。因此，我们分别将这三个图中的项目嵌入初始化为：

$$\mathbf{E}_A^{(0)} = [\mathbf{e}_{A1}^{(0)}; \mathbf{e}_{A2}^{(0)}; \dots] \in \mathbb{R}^{|A| \times d}$$

$$\mathbf{E}_C^{(0)} = [\mathbf{e}_{C1}^{(0)}; \mathbf{e}_{C2}^{(0)}; \dots] \in \mathbb{R}^{(|A|+|B|) \times d}$$

Transition-Aware Graph Construction

基于用户交互序列构建的**序列图**⁺，将序列中的每一项视为一个节点，并在相邻的项之间添加边。通过合并数据集中的所有序列，得到完整的图。不同于以往的方法，只考虑了具有**正面反馈**⁺的项目进行图构建，我们还包含了具有负面反馈的项目。综合跨域图包含了域A和域B中所有具有正面和负面反馈的项目。然而，这种方法引入了一些挑战。首先，在图的传播阶段，节点从其邻居接收信息。本质上，具有不同反馈的项目应该传播相反的信息。例如，具有正面反馈的项目应从具有正面反馈的相邻项目接收正面信息，从具有负面反馈的相邻项目接收负面信息。因此，我们引入反馈转换信息以获得转换矩阵**T**。如果两个相邻的节点具有不同类型的反馈，我们设置它们对应的值相反。具体地，对于相邻的两项*i*和*j*，以及对应的反馈 y_i 和 y_j ，转换矩阵定义如下：

正面反馈的项目*i*和正面反馈的项目*j*之间的转换矩阵值为： $T_{ij} = -T_{ji}$ 负面反馈的项目*i*和负面反馈的项目*j*之间的转换矩阵值为： $T_{ij} = T_{ji}$ 具有正面反馈的项目*i*和负面反馈的项目*j*之间的转换矩阵值为： $T_{ij} = -T_{ji}$ 具有负面反馈的项目*i*和正面反馈的项目*j*之间的转换矩阵值为： $T_{ij} = T_{ji}$

$$\mathbf{T}_{ij} = \begin{cases} 1 & \text{if } y_i = y_j, \\ -1 & \text{if } y_i \neq y_j, \\ 0 & \text{if } i \text{ and } j \text{ are not adjacent.} \end{cases}$$

然后我们获得转换意识的项目间转换图的**邻接矩阵**⁺表示：

$$\mathbf{W} = \begin{pmatrix} \mathbf{0} & \mathbf{T} \\ \mathbf{T}^\top & \mathbf{0} \end{pmatrix}$$

为了稳定训练，我们使用标准化形式：

$$\widehat{\mathbf{W}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}$$

其中， $\mathbf{D} \in \mathbb{R}^{(|A|+|B|) \times (|A|+|B|)}$ 是一个对角矩阵，其中 \mathbf{D}_{ii} 表示 \mathbf{W} 的第*i*行中非零元素的数量。对于单域图A和B以及跨域图，我们分别定义了它们的转换矩阵为 $\widehat{\mathbf{W}}_A$ $\widehat{\mathbf{W}}_B$ ，以及 $\widehat{\mathbf{W}}_C$ 。

Propagation and Aggregation

对于每张图，我们采用一种广泛使用的基于LightGCN的传播方法，定义了第*k*层的传播为简单的加权求和聚合操作：

$$\mathbf{e}_i^{(k+1)} = \sum_{j \in \mathcal{N}_i} \frac{\mathbf{T}_{ij}}{\sqrt{|\mathcal{N}_i|} \sqrt{|\mathcal{N}_j|}} \mathbf{e}_j^{(k)}$$

其中， $\mathbf{e}_i^{(k)}$ 在矩阵 $\mathbf{E}^{(k)} = [\mathbf{e}_1^{(k)}; \mathbf{e}_2^{(k)}; \dots]$ 中 \mathcal{N}_i 和 \mathcal{N}_j 分别对应图中节点*i*和*j*的邻居数量。然后我们可以表述这三个图的传播过程为矩阵等效形式，即：

$$\begin{aligned} \mathbf{E}_A^{(k+1)} &= \widehat{\mathbf{W}}_A \mathbf{E}_A^{(k)}, \\ \{\mathbf{E}_B^{(k+1)}\} &= \widehat{\mathbf{W}}_B \mathbf{E}_B^{(k)}, \\ \mathbf{E}_C^{(k+1)} &= \widehat{\mathbf{W}}_C \mathbf{E}_C^{(k)}. \end{aligned}$$

在经过*K*层之后，三个图的最终嵌入通过以下方式获得：

$$\begin{aligned} \mathbf{E}_A &= \frac{1}{K} \sum_{k=0}^K \mathbf{E}_A^{(k)}, \\ \{\mathbf{E}_B\} &= \frac{1}{K} \sum_{k=0}^K \mathbf{E}_B^{(k)}, \\ \mathbf{E}_C &= \frac{1}{K} \sum_{k=0}^K \mathbf{E}_C^{(k)}. \end{aligned}$$

Cross-Transition Multi-Head Self-Attention

知乎

换感知的图编码器，为这个序列获取跨域的嵌入表示 \mathbf{E}_C 。接着，为了捕获序列关系，我们进一步引入位置嵌入 \mathbf{P}_C 。对于自注意力机制，我们定义： $\hat{\mathbf{E}}_C = \mathbf{E}_C + \mathbf{P}_C$ 作为输入。然而，如果直接使用所有嵌入作为输入而没有任何限制，那么自注意力机制就难以区分反馈信息和领域信息。因此，我们引入两个遮罩 \mathbf{M}^f 和 \mathbf{M}^d ，以更好地捕获反馈转换和领域转换，其中每个条目的定义为：

$$\mathbf{M}_{ij}^f = \begin{cases} 0 & \text{if } f_i = f_j, \\ 1 & \text{if } f_i \neq f_j. \end{cases}$$

$$\mathbf{M}_{ij}^d = \begin{cases} 0 & \text{if } d_i = d_j, \\ 1 & \text{if } d_i \neq d_j. \end{cases}$$

然而，仅仅引入这两种遮罩机制是不足以强烈提取跨转换信息的。因此，我们对这两种遮罩进行交叉处理，从而得到能够捕获四类跨转换信息的交叉遮罩：

$$\begin{aligned} \mathbf{M}_1 &= \mathbf{M}^f \& \mathbf{M}^d, \\ \mathbf{M}_2 &= \mathbf{M}^f \& \widetilde{\mathbf{M}}^d, \\ \mathbf{M}_3 &= \widetilde{\mathbf{M}}^f \& \mathbf{M}^d, \\ \mathbf{M}_4 &= \widetilde{\mathbf{M}}^f \& \widetilde{\mathbf{M}}^d. \end{aligned}$$

位操作的AND，以及 $\widetilde{\mathbf{M}}$ 代表 \mathbf{M} 的取反。为了考虑所有四种类型的交叉转换，我们为Multi-Head Self-attention的每个头配备不同的掩码。具体来说，我们设置总体掩码⁺为：

$$\mathbf{M}_C = [\mathbf{M}_1; \mathbf{M}_2; \mathbf{M}_3; \mathbf{M}_4; \dots]$$

交叉转换的Multi-Head Self-attention随后以 $\hat{\mathbf{E}}_C$ 和 \mathbf{M} 作为输入，其中 $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \hat{\mathbf{E}}_C$ 然后是一个前馈层，输出最终的嵌入 \mathbf{E}'_C 。

$$\mathbf{E}'_C = \text{FFN} \left(\text{MHSA} \left(\hat{\mathbf{E}}_C, \hat{\mathbf{E}}_C, \hat{\mathbf{E}}_C, \mathbf{M}_C \right) \right)$$

具体来说，多头自注意力（MHSA）的计算如下：

$$\text{MHSA} \left(\hat{\mathbf{E}}_C, \hat{\mathbf{E}}_C, \hat{\mathbf{E}}_C, \mathbf{M}_C \right) = \text{Concate}(\text{head}_1, \text{head}_2, \dots) W$$

其中每个头部为 $i = 1, 2, \dots$

$$\begin{aligned} \text{head}_i &= \text{Attention} \left(\hat{\mathbf{E}}_C, \hat{\mathbf{E}}_C, \hat{\mathbf{E}}_C, \mathbf{M}_{i\%4} \right) \\ &= \text{Softmax} \left(\hat{\mathbf{E}}_C \hat{\mathbf{E}}_C^\top / \sqrt{d/h} \odot \mathbf{M}_{i\%4} \right) \hat{\mathbf{E}}_C, \end{aligned}$$

其中 \odot 表示Hadamard乘积。

然而，对于单一域序列，域转换不发生，因此我们只考虑反馈转换。具体地，对于感知转换的图编码器后的单一域嵌入表示 \mathbf{E}_A 和 \mathbf{E}_B ，以及位置嵌入 \mathbf{P}_A 和 \mathbf{P}_B ，我们定义：

$$\hat{\mathbf{E}}_A = \mathbf{E}_A + \mathbf{P}_A$$

$$\hat{\mathbf{E}}_B = \mathbf{E}_B + \mathbf{P}_B$$

$$\mathbf{M}_A = \mathbf{M}_B = [\mathbf{M}^f; \widetilde{\mathbf{M}}^f; \mathbf{M}^f; \dots]$$

然后，通过多头自注意力机制⁺，得到最终的嵌入表示：

$$\mathbf{E}'_A = \text{FFN} \left(\text{MHSA} \left(\hat{\mathbf{E}}_A, \hat{\mathbf{E}}_A, \hat{\mathbf{E}}_A, \mathbf{M}_A \right) \right)$$

$$\mathbf{E}'_B = \text{FFN} \left(\text{MHSA} \left(\hat{\mathbf{E}}_B, \hat{\mathbf{E}}_B, \hat{\mathbf{E}}_B, \mathbf{M}_B \right) \right)$$

Model Training

$$\mathcal{L}_{\text{total}} = \underbrace{\mathcal{L}_{\text{single}}^A + \mathcal{L}_{\text{single}}^B}_{\text{Single-Domain Loss}} + \underbrace{\mathcal{L}_{\text{cross}}^A + \mathcal{L}_{\text{cross}}^B}_{\text{Cross-Domain Loss}}$$

在损失函数定义的地方，将损失函数设定为预测和实际值之间的交叉熵+损失：

$$\begin{cases} \mathcal{L}_{\text{single}}^A = -\log \text{Softmax}(\text{MLP}_A(\mathbf{E}'_C + \mathbf{E}'_A))_{a_{|S_A|+1}}, \\ \mathcal{L}_{\text{single}}^B = -\log \text{Softmax}(\text{MLP}_B(\mathbf{E}'_C + \mathbf{E}'_B))_{b_{|S_B|+1}}, \\ \mathcal{L}_{\text{cross}}^A = -\log \text{Softmax}(\text{MLP}_A(\mathbf{E}'_C))_{a_{|S_A|+1}}, \\ \mathcal{L}_{\text{cross}}^B = -\log \text{Softmax}(\text{MLP}_B(\mathbf{E}'_C))_{b_{|S_B|+1}}. \end{cases}$$

Experiments

我们在豆瓣公开的数据集上进行实验。我们选择了两个CDSR场景进行实验：“图书-电影”和“图书-音乐”。为了确保顺序约束，我们保留了包含一年内至少3个来自每个领域的项目的所有跨域交互序列。评分大于3被视为正面反馈，而评分小于或等于3被视为负面反馈。

Dataset	Douban		
Domain	Book	Movie	Music
Users	26,877	28,718	23,822
Items	187,520	57,424	185,574
Records	1,097,148	2,828,585	1,387,216
Density	0.0218%	0.1715%	0.0314%
Avg. Rating	4.0391	3.8101	4.1749

Experimental Results and Analysis

Overall Performance

表展示了在“书籍-电影”和“书籍-音乐”跨领域推荐系统（CDSR）场景下方法的性能。

Methods	Book-domain recommendation						Movie-domain recommendation					
	MRR		NDCG		HR		MRR		NDCG		HR	
	@10	@5	@10	@1	@5	@10	@10	@5	@10	@1	@5	@10
GRU4Rec	1.31	1.28	1.39	0.93	1.59	1.97	1.62	1.58	1.80	0.99	2.13	2.83
SASRec	1.35	1.32	1.43	0.98	1.62	1.97	1.89	1.87	2.09	1.24	2.45	3.13
SRGNN	1.40	1.38	1.48	1.00	1.70	2.00	1.69	1.63	1.84	1.11	2.12	2.78
CoNet	1.35	1.35	1.48	0.96	1.72	2.11	1.62	1.55	1.79	1.01	2.07	2.83
π-Net	1.42	1.44	1.61	0.91	1.93	2.48	1.62	1.58	1.85	0.90	2.22	3.08
C2DSR	1.52	1.49	1.64	1.06	1.89	2.45	1.93	1.92	2.15	1.25	2.54	3.26
TriCDR	1.55	1.54	1.68	1.10	1.95	2.39	2.00	1.98	2.23	1.30	2.62	3.37
Transition ²	1.63	1.62	1.77	1.15	2.04	2.51	2.32	2.33	2.59	1.53	3.05	3.80

Table 2: Experimental results (%) of the Book-Movie domains. The best result is bolded and the runner-up is underlined. Improvements over the second-best methods are significant (*t*-test, *p*-value < 0.05).

Methods	Book-domain recommendation						Music-domain recommendation					
	MRR		NDCG		HR		MRR		NDCG		HR	
	@10	@5	@10	@1	@5	@10	@10	@5	@10	@1	@5	@10
GRU4Rec	1.24	1.19	1.31	0.90	1.47	1.84	1.18	1.15	1.29	0.77	1.51	1.93
SASRec	1.22	1.20	1.30	0.88	1.49	1.82	1.25	1.23	1.36	0.84	1.59	1.98
SRGNN	1.24	1.22	1.34	0.91	1.52	1.86	1.28	1.25	1.39	0.86	1.61	2.05
CoNet	1.30	1.31	1.44	0.90	1.69	2.11	1.23	1.21	1.39	0.77	1.64	2.18
π-Net	1.28	1.26	1.45	0.75	1.75	2.35	1.20	1.17	1.38	0.64	1.67	2.34
C2DSR	1.30	1.26	1.38	0.94	1.56	1.92	1.30	<u>1.30</u>	<u>1.43</u>	<u>0.88</u>	<u>1.69</u>	2.10
TriCDR	<u>1.34</u>	<u>1.31</u>	1.43	<u>0.95</u>	1.64	2.03	<u>1.30</u>	1.28	1.42	0.86	1.67	2.10
Transition ²	1.40	1.39	1.53	0.96	1.80	<u>2.22</u>	1.41	1.40	1.57	0.90	1.87	2.38

Table 3: Experimental results (%) of the Book-Music domains. The best result is bolded and the runner-up is underlined. Improvements over the second-best methods are significant (*t*-test, *p*-value < 0.05).

(1) 对于序列推荐（SR）基准线，GRU4Rec、SASRec 和 SRGNN 表现良好，其中SRGNN 表现最佳。这表明使用图神经网络构建交互序列对于提高推荐的准确性具有重要的价值。

架构的弱能力。

(3) 我们的模型在所有评价指标上都显著优于所有基准线，证明了在跨领域推荐任务中，我们的模型具有优越性。这些实验结果表明，在构建单领域和跨领域推荐时，考虑领域转换和反馈转换是非常有效的。

因此，这些实验不仅验证了使用图神经网络构建交互序列在提高推荐准确性方面的价值，还证明了在跨领域推荐系统中，考虑领域间的知识转移和领域转换是关键因素。

原文《Modeling Domain and Feedback Transitions for Cross-Domain Sequential Recommendation》

发布于 2024-09-06 15:52 · IP 属地北京

腾讯 工业级推荐系统 序列推荐



理性发言，友善互动



还没有评论，发表第一个评论吧

推荐阅读



如何借助腾讯生态实现私域用户增长 (腾讯有数2.0深度解析)

自在军姐 发表于产品爱上增...



浅析-腾讯产品项目的流程

PMdiss



盘点腾讯、网易这些大厂都在用的4大增长策略 | 案例解析

日签君AI... 发表于艾体验AI...



浅析-腾讯产品项目的流程

木深