

清华大学-MTAW：精准捕获瞬时兴趣：序列推荐中transformer探索



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

16 人赞同了该文章

Introduction

在许多实际的推荐场景中，由于隐私政策，用户通常无法被识别或跟踪。因此，[推荐系统](#)⁺需要在缺乏用户信息的情况下辨别用户的潜在兴趣。SR (Session-based Recommendation) 是解决隐私问题的关键。它使用会话作为序列来预测用户潜在兴趣。GRU4Rec和GNN是最常用的SR模型。它们分别使用RNN和[图神经网络](#)⁺来模拟用户交互的顺序信息和描述相邻交互之间的节点对关系。大多数SR方法在训练时使用通用的[优化方法](#)⁺，这可能导致性能降低。然而，现有模型存在两个问题：(a) 对于[时间序列数据](#)⁺，模型过于强调相邻交互的相对顺序，忽视了其中的噪声。(b) 模型未考虑到样本间的差异，可能导致不同样本在模型训练过程中的重要性不同。为了克服这些问题，我们提出了MTAW模型，它能够捕获用户兴趣变化趋势，而不是关注交互的相对顺序。此外，我们设计了自适应权重损失函数，可以根据每个样本的预测难度自适应调整权重。

- 我们利用用户当前的即时兴趣来跟踪他们的变化趋势，并且通过区分处理不同用户的兴趣，提供更个性化的推荐。
- 设计AW损失函数，使其根据不同样本自动分配权重，从而提高模型拟合效果。
- 实验结果在两组数据上显示了MTAW在大部分情况下的优越性。而且相较于最先进的方法，MTAW不仅效果更好，而且效率更高且所需参数更少。

Method

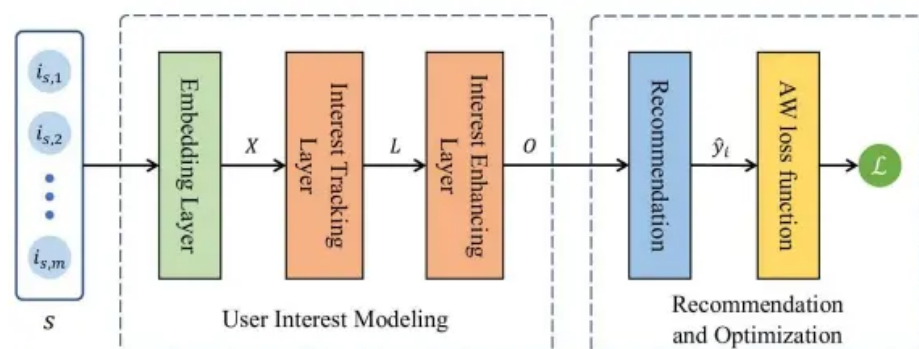


Figure 1: The architecture of MTAW.

1 用户兴趣建模：通过两步追踪用户兴趣演变：i) 兴趣跟踪；ii) 兴趣增强。2. 推荐与优化：根据会话表示为样本分配不同权重进行推荐和优化。

Problem Statement

$$s = [i_{s,1}, i_{s,2}, \dots, i_{s,m}]$$

其中 N 是总项数 m 是会话 s 的长度

$$i_{s,k} \in I (1 \leq k \leq m)$$

表示用户在会话 s 中点击的一个项目。目标是预测会话 s 的下一个点击，即序列标签 $i_{s,m+1}$ 。对于会话 s ，计算所有可能项目的概率 \hat{y} ，并按 K 个最高的评分顺序推荐相应的项目。

User Interest Modeling

为了将输入会话转换为向量，我们构建嵌入层。对于输入会话中的每个项 i ，隐藏表示为：

$x_i = e_i + p_i$ 其中， $e_i \in \mathbb{R}^d$ 表示物品嵌入 d 表示嵌入大小 $p_i \in \mathbb{R}^d$ 表示位置嵌入 $x_i \in \mathbb{R}^d$ 表示物品 i 的隐藏表示。此外，我们使用

$$X = \{x_1, x_2, \dots, x_m\}$$

来表示会话

$$s = [i_1, i_2, \dots, i_m]$$

Interest Tracking Layer

使用注意力网络，根据用户先前交互提取用户当前时刻的即时兴趣。

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q^T K}{\sqrt{d}}\right) V$$

我们取 Q, K 和 V 作为输入矩阵，并将 X 中的所有数据按时间顺序切割成 m 个切片⁺：

$X'_1 = \{x_1\}$ $X'_2 = \{x_1, x_2\}$, ... $X'_m = \{x_1, x_2, \dots, x_m\}$ 。针对每个切片 X'_t （其中 $t \leq m$ 表示切片序号），我们利用注意力机制⁺来挖掘用户的即时兴趣。

$$Q_x = \text{ReLU}(\text{MLP}(x_t)), \\ l'_t = \text{Attention}(Q_x, X'_t, X'_t),$$

通过在注意力网络中使用掩码矩阵，将会话 s 的隐藏状态集合

$$L' = \{l_{\{1\}}, l_{\{2\}}, \dots, l_{\{m\}}\}$$

并行地更新，同时加入位置-wise feed-forward network (FFN)来增加模型的非线性。

$$L = \text{FFN}(L') = \text{MLP}(\text{ReLU}(\text{MLP}(L')))$$

其中两个MLP代表两种不同的多层感知器⁺。然后，我们添加一个残差连接⁺和层标准化来缓解模型训练的不稳定。我们还添加了丢弃机制来防止过拟合⁺。为了简化起见，我们将“兴趣跟踪层”表示为ITL，即： $L = \text{ITL}(X)$ 其中

$$L = \{l_1, l_2, \dots, l_m\}, l_m \in \mathbb{R}^d$$

是该层的最终输出，其中每个都表示当前交互用户的即时兴趣。

Interest Enhancing Layer

设计兴趣增强层深入挖掘并分析用户兴趣。同时将兴趣变化趋势信息注入会话表示中。使用注意力网络区别地整合即时兴趣以获取用户下一个项的兴趣变化趋势。

$$O = \text{Attention}(Q_l, L, L)$$

Recommendation and Optimization

知乎

$$\begin{aligned}\hat{O} &= \text{L2Norm}(O), \hat{x}_i = \text{L2Norm}(x_i), \\ \hat{y}_i &= \text{softmax}(\hat{O}^T \hat{x}_i), \\ AW(\hat{y}_i, y_i) &= CE(\hat{y}_i, y_i) + w_i \left(\frac{1}{\sigma(y_i)} \right) - w_i(0)\end{aligned}$$

其中 w_i 是给定样本的权重 $\sigma(y_i)$ 是样本的预测偏斜。通过调整这个调节因子，我们可以使训练更加稳定并提高模型性能。

$$\begin{aligned}p_i &= \begin{cases} \hat{y}_i, & \text{if } y = 1, \\ 1 - \hat{y}_i, & \text{otherwise,} \end{cases} \\ \mathcal{L} &= - \sum_{i=1}^M (2 - 2p_i)^\gamma \log(p_i),\end{aligned}$$

随着 γ 的增加，调节因子的影响也会增加。 y 是下一项的真实概率分布，是一个one-hot向量。 M 是总的样本数。由于 $p_i \in [0, 1]$ ，所以 $(2 - 2p_i) \approx 1$ 。 $(2 - 2p_i)^\gamma$ 表示当前迭代中预测值与真实值之间的偏差，即样本的难度。直观上讲，调节因子可以减少简单样例的损失贡献并扩大难样例的贡献。

Experiments

Datasets and Metrics.

我们对两份公开数据集进行了实验：一份来自IJCAI-15竞赛的天猫（Tmall），其中包括天猫在线平台上的匿名购物日志；另一份来自Kaggle比赛的零售火箭（RetailRocket），其中包括用户浏览活动的六个月数据。我们使用 S^2 -DHCN提供的预处理版公共数据集来实施我们的模型，并在表中展示了这些数据集的统计结果。我们的评估指标是MRR@K（均值逆向排名在K处）和P@K（精确度在K处），其中K的值为10和20。

| Dataset | # training | # test | # items | Avg. Len. |
|--------------|------------|--------|---------|-----------|
| RetailRocket | 433,643 | 15,132 | 36,968 | 5.43 |
| Tmall | 351,268 | 25,898 | 40,728 | 6.69 |

Baselines and Implementation Details.

我们使用Adam优化器，学习率为0.001。嵌入大小为100，训练轮数为50，批大小为100。在天猫数据集上设置 γ 为2，在RetailRocket数据集上设置 γ 为6。

Results

Overall Performance.

(a) 传统模型（如Item-KNN和FPMC）可能在某些指标上优于RNN方法（如GRU4REC）。这说明仅仅将对话视为严格顺序的序列可能会限制捕捉用户真实兴趣的能力。(b) 图神经网络模型 \star 优于大多数模型，但会话建模为有向图 \star ，而被MTAW超越。这证明了兴趣趋势建模和自适应样例权重分配的优势。

知乎

| | | | | | | | | |
|---------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| FPMC [13] | 13.10 | 7.12 | 16.06 | 7.32 | 25.99 | 13.38 | 32.37 | 13.82 |
| GRU4REC [2] | 14.16 | 6.56 | 18.20 | 6.85 | 34.41 | 15.06 | 44.89 | 15.77 |
| NARM [6] | 19.17 | 10.42 | 23.30 | 10.70 | 42.07 | 24.88 | 50.22 | 24.59 |
| STAMP [8] | 22.63 | 13.12 | 26.47 | 13.36 | 42.95 | 24.61 | 50.96 | 25.17 |
| SASRec [4] | 22.06 | 14.02 | 26.95 | 14.21 | 44.65 | 25.53 | 51.12 | 25.91 |
| NextItNet [20] | 22.67 | 13.12 | 27.22 | 13.32 | 41.12 | 23.99 | 48.26 | 24.48 |
| SR-GNN [16] | 23.41 | 13.45 | 27.57 | 13.72 | 43.21 | 26.07 | 50.32 | 26.57 |
| GC-SAN [18] | 21.32 | 12.43 | 25.38 | 12.72 | 43.21 | 26.07 | 50.32 | 26.57 |
| GCE-GNN [15] | <u>28.02</u> | <u>15.08</u> | <u>33.42</u> | <u>15.42</u> | 46.05 | <u>27.48</u> | 53.63 | <u>28.01</u> |
| S ² -DHCN [17] | 26.22 | 14.60 | 31.42 | 15.05 | <u>46.15</u> | 26.85 | <u>53.66</u> | 27.30 |
| MTAW | 31.67[†] | 18.90[†] | 37.17[†] | 19.14[†] | 48.41[†] | 28.95[†] | 56.39[†] | 30.52[†] |
| Improv. (%) | 13.03 | 25.33 | 11.22 | 24.12 | 4.90 | 9.02 | 5.09 | 8.96 |

Study on AW Loss Function.

验证AW Loss的优点并研究 γ 对最终性能的影响，我们进一步做了实验。在RetailRocket数据集上，我们以P@20作为评估指标，在2, 4, ..., 10范围内搜索 γ 值。同时，我们也比较了不使用AW Loss优化的variant *w/o*AW的结果。结果如图所示，可以发现，随着 γ 值增大，MTAW性能先增后减。当 γ 设置为6时，MTAW表现最佳。此外，当我们优化MTAW时未使用AW Loss时，性能会下降，从而证明了AW Loss的有效性。

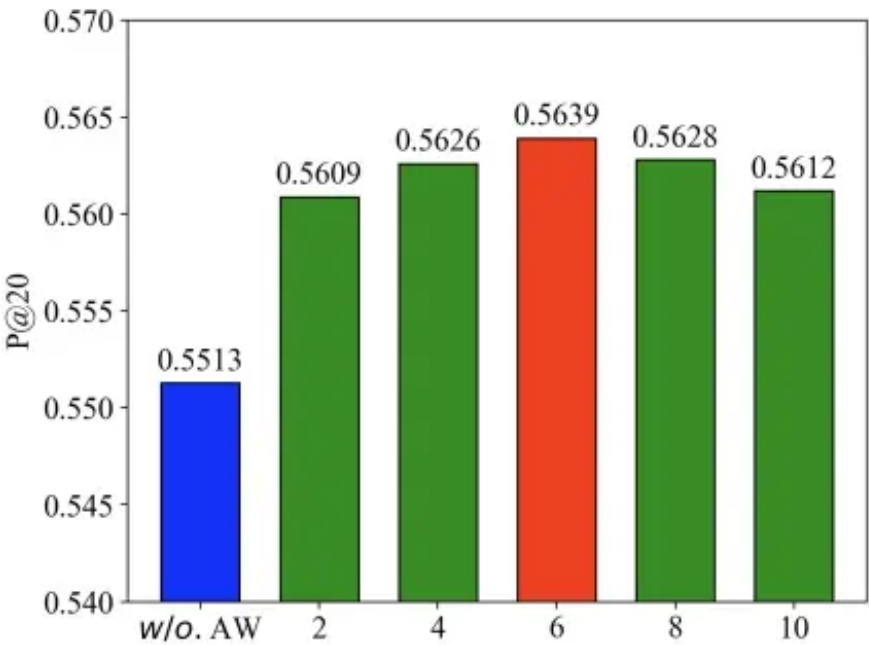


Figure 2: AW Loss Study on RetailRocket.

Efficiency Comparison.

通过比较MTAW训练时间和参数数量与最新最佳实践模型在同一设备上的性能，我们评估了MTAW的效率。结果显示，MTAW比最佳实践方法更有效率，需要更少的参数。在RetailRocket数据集上，MTAW实现了16.59倍的速度提升，同时参数量更少。因此，我们得出结论，MTAW既有效又高效，而兴趣趋势建模是一个潜在的未来工作。

| | Time | #Params | Time | #Params |
|-------------|--------|---------|--------|---------|
| NextItNet | 34m51s | 4.23M | 64m27s | 3.85M |
| SR-GNN | 4m7s | 4.23M | 23m54s | 3.86M |
| GC-SAN | 3m42s | 4.24M | 12m20s | 3.87M |
| GCE-GNN | 2m20s | 4.35M | 16m02s | 3.98M |
| S^2 -DHCN | 32m34s | 4.31M | 77m12m | 3.94M |
| MTAW | 47s | 4.02M | 58s | 3.69M |

Conclusion

本文提出MTAW模型，它使用注意力机制捕捉用户的瞬时兴趣并动态分配采样权重以挖掘兴趣变化趋势。此外，设计了AW损失函数⁺。实验证明了MTAW在两个数据集上的优势。

原文《Mining Interest Trends and Adaptively Assigning Sample Weight for Session-based Recommendation》

发布于 2024-02-01 10:42 · IP 属地北京

清华大学 序列推荐 Transformer

赞同 16 添加评论 分享 喜欢 收藏 申请转载 ...



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

Offline Reinforcement Learning as One Big Sequence Modeling Problem

Michael Janner Qiyang Li Sergey Levine
University of California at Berkeley
{janner, qli}@berkeley.edu svlevine@eecs.berkeley.edu

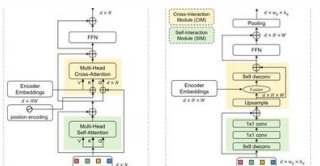
【强化学习 215】Trajectory Transformer

张海抱 发表于强化学习前...



国科大团队提出首个CNN和Transformer双体基网模型...

新智元



华为+清华大学提出DECO | DETR系列用Transformer就...

愿得一人心

Decision Transformer: Reinforce Learning via Sequence Models

Lili Chen^{1,2}, Kevin Lu^{1,2}, Aravind Rajwaran², Kinan Aditya Greer^{2,3}, Michael Laskin¹, Pieter Abbeel¹, Aravind Srinivas¹
¹equal contribution ²equal advising
¹UC Berkeley ²Facebook AI Research ³UCLA ⁴OpenAI ⁵Illinois, kzi@berkeley.edu

【强化学习 214】Decision Transformer

张海抱 发表于强化学习前...