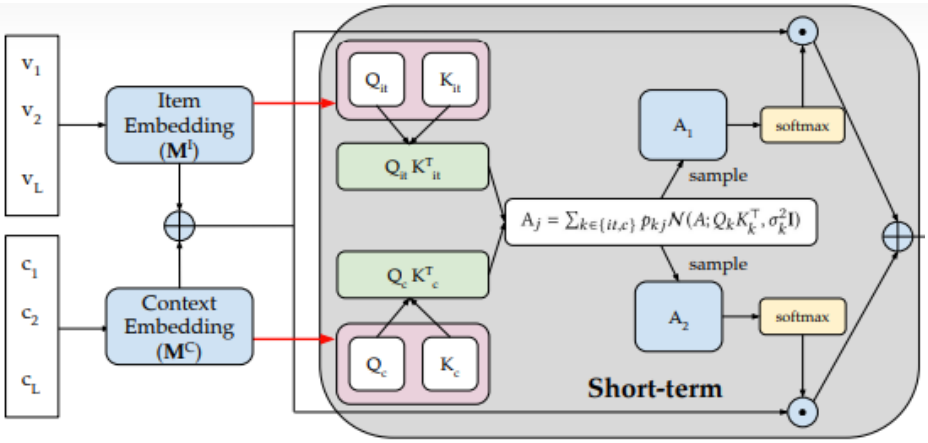
知乎



MOJITO-使用transformer对用户行为序列进行时间场景建模

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

5 人赞同了该文章

Introduction

对于在线平台如电影和音乐流媒体服务来说，内容推荐至关重要。通过模拟用户行为的时间序列，推荐系统可以预测在给定时间最佳的内容，研究者们致力于提升推荐系统的表现，尤其是最近推出的Transformer模型在推荐领域表现出色。然而，现有的一些Transformer系统并未考虑到用户的交互时间因素。

尽管这些系统能处理序列中的相对位置，但在现实生活中，时间信息可能对推荐系统产生关键影响。例如，研究显示，捕捉时间元素如每日的小时能帮助预测用户在音乐流媒体上的下一步行为。另一项研究也发现，在不同的领域，如音乐和电子商务，交互序列中存在很多依赖于时间的模式，包括季节性行为（如12月播放圣诞歌曲）和周期性行为（如每3个月购买一把新的牙刷）。

本研究提出了一种改进的Transformer SR系统MOJITO，通过同时考虑用户-项目交互和时间上下文，学习两种不同的注意力矩阵，并将它们融合到基于高斯混合的时间上下文和item向量表示中，以有效对用户行为进行时间序列建模。我们在多个真实世界的数据集上进行了实验，包括电影推荐和音乐推荐等，结果显示MOJITO在跨时间上下文的相关依赖问题上有更好的表现。

Preliminaries

Problem Formulation

考虑在线服务上的用户集合$\mathcal{U}$和物品集合$\mathcal{V}$。每个用户都有一个交互序列，例如在音乐流媒体服务上的听歌会话。这个序列的元素是$v_1, v_2, \ldots, v_L$，其中i代表序列的索引

$$v_t \in \mathcal{V}, \forall t \in \{1, \ldots, L\}$$

且L是序列的长度。我们可以使用S_{ui}来表示这个交互序列。观察每个S_{ui}的同时，我们也会观测一个名为contextual sequence的c_{ui}。每个c_{ui}是一个元组，将第t^个^S_{ui}与$C \in \mathbb{N}^*$不同类型的time-related contextual information关联起来，例如月份、天数或周日等。

$$\mathbf{c}_t = (c_{t1}, c_{t2}, \ldots, c_{tC})$$

我们将这些值进行排序以考虑时间上的接近性，例如捕捉到一月更接近二月而不是七月。SR的目标是预测用户接下来要与之交互的下一个项v_{L+1}，在这个观察到的时间相关上下文中。

Transformers for Time-Aware SR

C_{ui}表示。时间相关的因素，如当前月份、天数或小时，可能显著影响用户的需求和偏好。因此，我们提出了一种新的时间感知的推荐方法

A Mixture System for Time-Aware SR

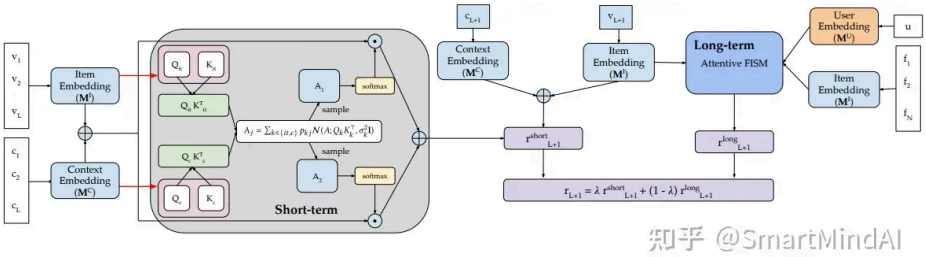


Figure 1: Architecture of MOJITO for time-aware SR using attention mixtures of temporal context and item embeddings.

General Overview

该节介绍MOJITO系统，包括两个组成部分：短期意图模型（受交互和时间背景影响）和长期偏好模型（捕捉长期偏好）。这两个模型分别通过$r_{L+1}^{\text{short}}(v)$和$r_{L+1}^{\text{long}}(v)$评估每个项$v$在扩展观察到的交互序列的可能性。为推荐物品，采用以下组合方式：给定超参数$\lambda \in 0, 1$。

$$r_{L+1}(v) = \lambda r_{L+1}^{\text{short}}(v) + (1-\lambda) r_{L+1}^{\text{long}}(v)$$

Short-Term Representation

我们使用自注意力机制学习时间相关的短期兴趣表示。然而，由于不同注意力头可能学习到相同的模式，从而导致冗余信息。为了解决这个问题，我们在Nguyen等人基础上，通过全局注意力矩阵使得不同的注意力头能够互相交互。而Nguyen等人则使用混合模型从每个头的局部注意力矩阵中采样。在此基础上，我们引入不同的诱导性语义偏置来增强全局矩阵表示的多样性。

Embedding Layer

处理从Section 中获取的S_{ui}和C_{ui}序列。使用

$$\mathbf{M}^I \in \mathbb{R}^{|\mathcal{V}| \times d}$$

表示一个可学习的"物品嵌入矩阵"，每一行是维度为d的向量表示每一个物品。一个序列S_{ui}可以用这个嵌入矩阵表示为

$$\mathbf{E}_{S_{ui}}^I = [\mathbf{m}_{v_1}^I, \mathbf{m}_{v_2}^I, \ldots, \mathbf{m}_{v_L}^I]^\intercal \in \mathbb{R}^{L \times d}$$

$\mathbf{E}_{S_{ui}, C_{ui}}$ is a matrix with dimensions$L \times 2d$, where each row represents the concatenation of two-dimensional vectors$\mathbf{m}_{v_l}^I$ and

$$\mathbf{m}_{c_l}^C$$

for all$l = 1, \ldots, L$.将$\mathbf{E}_{S_{ui}, C_{ui}}$与位置嵌入$\mathbf{P}$结合来增强输入，最终得到$\mathbf{X}^{(0)}$。其中$\mathbf{x}_l^{(0)}$由实例特征和类标签的表示以及位置特征组成，我们希望利用这种增强方式区分在同一物品在不同位置上的影响。

Attention Mixtures

堆叠的B个自我注意力块(SAB)处理输入向量$\mathbf{X}^{(0)}$，每个块的输出是

$$\mathbf{X}^{(b)} = \mathrm{SAB}^{(b)}(\mathbf{X}^{(b-1)})$$

其中$b \in \{1, \ldots, B\}$。SAB包含一个自我注意力层$\mathbf{SAL}(\cdot)$，该层将上下文和项目嵌入表示混合在一起，然后是一个前馈神经网络层$\mathbf{FFL}(\cdot)$。B和H分别表示堆叠的层数和头部的数量。

知乎

对于输入

$\mathbf{X} \in \mathbb{R}^{L \times 2d}$

我们使用输出投影矩阵

$\mathbf{W}^O \in \mathbb{R}^{2Hd \times 2d}$

以及$\mathbf{W}_1$

$\mathbf{W}_2 \in \mathbb{R}^{2d \times 2d}$

和$\mathbf{b}_1$

$\mathbf{b}_2 \in \mathbb{R}^{1 \times 2d}$

来构建FFL网络。此外，我们还定义了每个头的输出

$$\mathbf{X}_j^{\text{Att}} = \text{softmax}(\mathbf{A_j}/\sqrt{d})\mathbf{V}$$

在本工作中，我们通过采样一个高斯混合概率模型得到$\mathbf{A_j}$，并以此自动学习时间上下文和前项交互对序列建模的重要性。

$$\mathbf{A}_j \sim \sum_{k \in \{it,c\}} p_{kj} \mathcal{N}(\mathbf{A}; \mathbf{Q}_k \mathbf{K}_k^\intercal, \sigma_k^2), \sum_{k \in \{it,c\}} p_{kj} = 1, p_{kj} \geq 0,$$

Prediction

$r_{L+1}^{\text{short}}(v) = \mathbf{x}_L^{(B)\intercal} \left[\mathbf{m}_v^I; \mathbf{m}_{c_{L+1}}^C\right]$

$\mathbf{x}_L^{(B)}$是模型的第L个位置在经过B层注意力块后的输出。

Long-Term Representation

长期表示学习：捕获用户偏好中的异质性至关重要。

Attentive FISM

$\hat{\mu}_{u,v}^{(l+1)} = \sum_{i=1}^N w_i f_i^{(l)}$

$\tilde{\mathbf{m}}_u(v) = \mathbf{m}_u + \sum_{f \in \mathcal{F} \setminus \{v\}} \frac{e^{\mathbf{m}_f^\intercal \mathbf{m}_v}}{\sum_{f' \in \mathcal{F} \setminus \{v\}} e^{\mathbf{m}_{f'}^\intercal \mathbf{m}_v}} \mathbf{m}_f \in \mathbb{R}^d$

其中向量

$\mathbf{m}_u \in \mathbb{R}^d$

是在训练过程中由模型学习得到的。

Prediction

$r_{L+1}^{\text{long}}(v) = \mathbf{m}_v^\intercal \, \tilde{\mathbf{m}}_u(v)$

长期偏好定义为与上下文无关。

Training Procedure

的词汇集合中随机选择一个负例。我们的目标是使系统的响应对应于子序列中的关键字具有较高的相关性得分，并对应于其他关键字具有较低的得分。为了实现这一目标，我们使用梯度下降法*来最小化损失函数$\mathcal{L}$，其定义为

$$\lambda\mathcal{L}^{\text{short}} + (1-\lambda)\mathcal{L}^{\text{long}}$$

其中$\lambda \in [0,1]$，该损失函数是短损失$\mathcal{L}^{\text{short}}$和长损失$\mathcal{L}^{\text{long}}$的线性组合*。

$$\mathcal{L}^{x} = -\sum_{s\in\mathcal{S}}\sum_{l=1}^{L}\left[\log\left(\sigma(r^{x}_{l+1}(v_{s,l+1}))\right) + \log\left(1-\sigma(r^{x}_{l+1}(o_{s,l+1}))\right)\right]$$

$F(x) = g(ax+b)$其中$g(a) = \sigma(a)$是sigmoid函数*a是一个常数b是一个变量。为了找到最佳的b值，我们需要使$F(x)$在所有可能的x值处都达到最优。 这是一个复杂的数学问题，需要使用优化算法（如梯度下降法）来解决。

在这个过程中，我们需要定义一个损失函数*$L(F,b)$，该函数衡量了当b取不同值时$F(x)$的性能。我们的目标是找到使得$L(F,b)$最小的b值。 这是一个非常一般的模型，适用于许多实际应用中的问题。具体的函数形式取决于问题的具体特性，包括输入数据的分布、输出数据的需求等。

Experimental Analysis

Experimental Setting

Datasets

我们研究了三个不同领域的实际数据集：电影、书籍和音乐推荐。

1. 电影推荐数据集包含6040个用户的3883部电影评价，每一对评价被视为一个二元交互。
2. 亚马逊收集了100万个用户与96421本交互书籍的数据，其中由109730个用户提交。
3. LFM-1b包含120322名用户的听歌数据，涉及3190371种不同的音乐物品，总听歌次数超过十亿次。

如Sun等人所示，对每个数据集进行了预处理步骤，包括递归过滤数据，直到所有用户（或所有物品）达到至少k^{\text{item}}（或k^{\text{user}}）的交互。k^{\text{item}}和k^{\text{user}}的具体值已在表中给出。

Task

我们通过在一个包含实际缺失项和1K其他未与之互动的"负面"项的评估集中评估SR模型的检索能力。模型需要推荐一个包含10个按相关性分数排序的项目列表。我们使用HR\@10和NDCG\@10指标，HR\@10报告了top-10列表中包含缺失项的比例，NDCG\@10还考虑了缺失项在列表中的排名。

Baselines

比较MOJITO与八种Transformer基准模型（SASRec, BERT4Rec, SSE-PT, FISSA, AttRec, TiSASRec, MeanTime和CARCA）。最后三种基线专门关注SR的时间特性。

Implementation Details

我们使用Adam优化器*在100个周期内训练模型。使用d=64，L=50，批量大小512，B=2，H=2进行MOJITO训练。通过验证集*选择所有其他超参数，并在GitHub仓库中报告了最优值。特别注意，我们测试了学习率0.0002-0.001，N取20-100，以及λ取0.1-1.0的结果。

Results and Discussion

知乎

Non Time-Aware	BERT4Rec	53.75 ± 0.17	75.59 ± 0.18	57.17 ± 0.15	79.90 ± 0.08	56.08 ± 0.04	68.87 ± 0.06
	SSE-PT	55.62 ± 0.24	79.61 ± 0.19	55.00 ± 0.25	79.63 ± 0.11	56.60 ± 0.39	*75.21 ± 0.36*
	AttRec	42.08 ± 0.31	69.23 ± 0.21	44.67 ± 0.20	71.00 ± 0.24	43.81 ± 0.19	64.66 ± 0.38
	FISSA	48.53 ± 0.31	74.16 ± 0.36	58.94 ± 0.11	81.42 ± 0.10	52.40 ± 0.19	68.20 ± 0.23
Time-Aware	TiSASRec	58.09 ± 0.26	*80.86 ± 0.28*	52.16 ± 0.07	78.53 ± 0.22	55.67 ± 0.33	73.25 ± 0.37
	MEANTIME	**59.97 ± 0.18**	79.76 ± 0.15	*58.95 ± 0.11*	82.02 ± 0.11	57.03 ± 0.06	71.60 ± 0.07
	CARCA	38.65 ± 0.14	64.51 ± 0.09	56.82 ± 0.23	*82.89 ± 0.10*	49.56 ± 0.11	66.30 ± 0.33
	MOJITO (ours)	*59.82 ± 0.17*	**82.09 ± 0.22**	**59.94 ± 0.23**	**83.26 ± 0.09**	**60.14 ± 0.19**	**75.72 ± 0.23**

表1总结了所有测试集⁺的表现，并列出了在五次模型训练中计算得出的标准差⁺。总体来说，MOJITO在三个数据集上均展现出竞争力，我们将在此处探讨。

Time-Aware vs Non Time-Aware SR

我们的研究表明，时间敏感的系统在MovieLens和Amazon Book上通常优于非时间敏感的基线。这对于推荐系统在模拟用户偏好与时间背景之间的依赖性非常重要。然而，在LFM-1b上的性能差异较小，可能是因为音乐消费的时间背景较为简单。

相较于书籍或电影消费，音乐消费的参与度更低，导致交互频率增加且时间间隔变短。这使对音乐推荐中的时间背景建模变得更加困难，尤其是考虑到同一时间背景下的音乐曲目数量更多。尽管如此，MOJITO（明确考虑周日周期信息）在LFM-1b上的表现最佳（NDCG 60.14%，HR 75.72%），表明它具有处理倾向于周期模式的音乐消费的粒度模型的能力。

MOJITO vs Other Time-Aware Systems

我们的实验结果显示，使用MOJITO的方法优于时间敏感基准，如亚马逊图书和电影流的用户排名结果也表明这种方法的有效性。

Conclusion

本研究提出一种名为TTSR的时间感知Transformer，利用混合注意力和项嵌入表示来有效处理时间上下文。通过实证研究⁺，该方法成功捕捉了时间上下文对用户偏好的影响，并使得对动作顺序模型的谨慎程度更高。

论文原文《Attention Mixtures for Time-Aware Sequential Recommendation》

编辑于 2024-02-20 16:59 · IP 属地北京

推荐系统 Transformer 场景建模