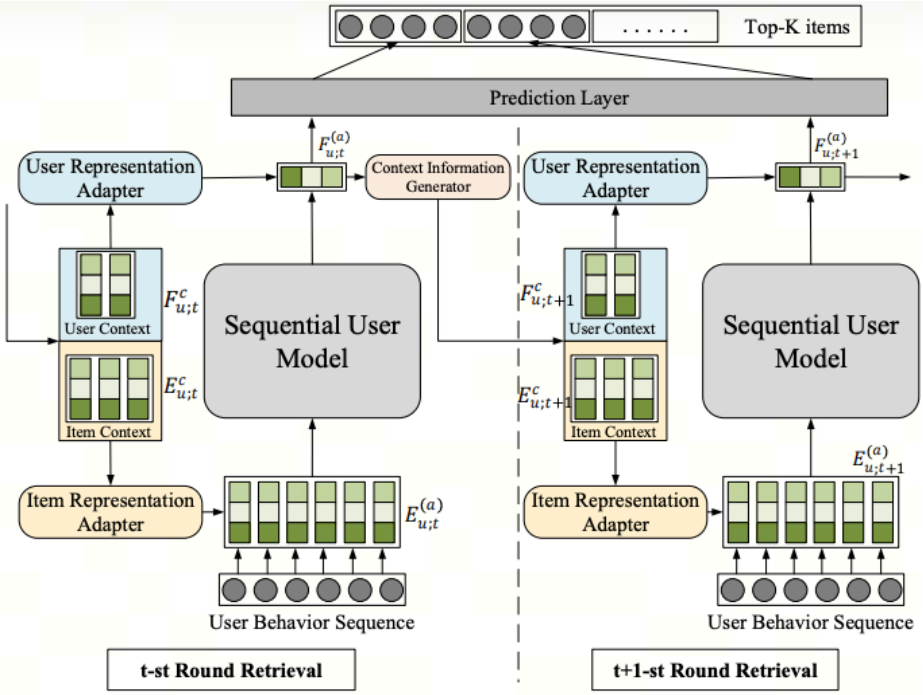


微软2024：Ada-Retrieval - 序列推荐领域的新突破，探索自适应多轮检索新范式

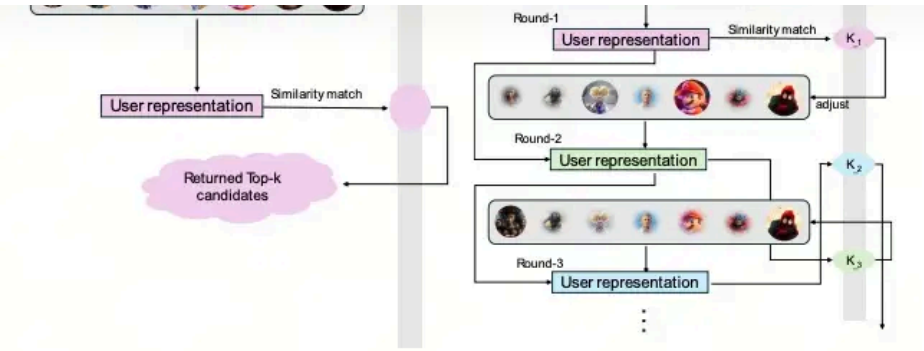
SmartMindAI
专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我
已关注

Introduction

推荐系统通过分析用户历史行为以提供个性化推荐，从而提升用户满意度和活跃度。其中，序列推荐因其捕捉时间序列行为动态并预测未来的能力，受到特别的重视。该领域研究广泛，采用多种基模型，如RNNs（循环神经网络）、CNNs（卷积神经网络）、transformers（变换器）和GNNs（图神经网络），这些模型共同推进了序列推荐技术的进步。

该论文并不旨在提出一个更强大的基模型，而是指出，尽管现有的推荐系统大多采用**单轮推理**方法来选取前k个最相关的项目，这种方法可能无法充分捕捉用户偏好的动态变化，以及适应不断变化的物品多样性。具体来说，给定用户的行为历史等信息，模型首先进行一次前向过程生成用户表示，然后用作查询匹配数据库中的前k个最相似项。然而，这种单轮推理方式可能导致用户表示在搜索物品空间时受限于固定的区域，无法应对用户兴趣的多样化。

知乎



(a) Single-round retrieval (b) Multi-round retrieval

Figure 1: Illustrations of (a) the conventional single-round retrieval paradigm, and (b) our proposed adaptive multi-round retrieval paradigm, in which the final retrieval result is the union of each individual retrieval K_i.

我们提出了一种**多轮推理**的方法，如图(b)所示，将寻找k个物品的目标分为n轮，每轮获取k/n数量的物品。不同的轮次，用户表示的前向传递独立进行。如果前一轮的推荐未充分满足用户需求，用户表示会在下一轮得到调整，使模型能在物品空间的不同区域寻找目标。这与搜索引擎的场景类似，如果当前信息不能准确回答问题，用户会修正他们的查询。因此，多轮推理能够利用前一轮的反馈信息来优化用户表示，避免用户表示被固定在一个静态区域，从而提供更动态和多样化的推荐。商品表示适配器和用户表示适配器。 商品表示适配器包含一个可学习滤波层（LFT）和一个基于检索上下文的注意力（CAT）层，用于根据检索上下文调整用户行为历史中的商品嵌入。这样，用户模型能够通过考虑候选商品空间的反馈，优化下一轮检索。用户表示适配器则包括门控循环单元（GRU）和多层感知器+（MLP）层。GRU层将所有前轮生成的用户表示编码为用户上下文，而MLP层将此上下文与当前用户表示融合，生成适应的用户表示。通过这些组件，Ada-Retrieval能够在检索过程中整合上下文信息，对传统的序列推荐模型进行改进，为商品检索生成逐步细化的用户表示，同时保持轻量级和模型无关的优势。

Preliminaries

Problem Formulation

假设我们有一个用户集合 $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ 和一个商品集合 $\mathcal{I} = \{i_1, i_2, \ldots, i_{|\mathcal{I}|}\}$

其中$u \in \mathcal{U}$代表用户$i \in \mathcal{I}$代表商品。

用户行为可以表示为行为序列 $\mathcal{S} = \{s_1, s_2, \ldots, s_{|\mathcal{U}|}\}$

在序列推荐中，用户的交互行为通常按照时间顺序排列： $s_u = \{i_1, i_2, \ldots, i_n\}$

其中$s_u \in \mathcal{S}$ $u \in \mathcal{U}$。目标是预测用户接下来可能要交互的商品，即$P(i_{n+1}|i_{1:n})$。

Base Sequential Model

(1) 嵌入生成器$\mathbf{EMB}(\cdot)$，负责将用户和商品转化为嵌入向量，如
$\mathbf{emb}_u = \mathbf{EMB}(\mathbf{i}_1, \mathbf{i}_2, \ldots, \mathbf{i}_n)$

(2) 序列理解器$\mathbf{SEL}(\cdot)$，通过处理时间序列捕捉上下文和时间依赖关系；

(3) 预测器$\mathbf{PRED}(\cdot)$，利用编码结果预测用户未来行为的概率，即
$p(i_{n+1}|i_{1:n}) = \mathbf{PRED}(\mathbf{SEL}(\mathbf{emb}_u))$

这个过程如图(a)所示，通过多轮迭代，模型能更好地理解用户动态，提升推荐的准确性和效率。

知乎

在推荐系统中，核心模块包括：(1)$\mathbf{EMB}(\cdot)$，将用户行为序列转化为嵌入，如 $\mathbf{emb}_u = \mathbf{EMB}(\mathbf{i}_1, \mathbf{i}_2, \ldots)$

(2)$\mathbf{SEL}(\cdot)$，如RNNs或Transformer，处理序列并捕获时间依赖和上下文；(3)$\mathbf{PRED}(\cdot)$，利用$\mathbf{enc}_u$预测后续感兴趣商品的概率。通过迭代，模型理解用户动态，生成个性化推荐，提升推荐准确性和用户满意度。使用RNNs或Transformer能捕捉序列中的顺序信息和长期依赖，而Transformer通过自注意力理解上下文关系，帮助模型适应用户兴趣的变化。最终，预测层根据$\mathbf{enc}_u$给出预测，实现连续且个性化的推荐。

$$\mathbf{F_u} = \mathbf{SEL}(\mathbf{E_u})$$

在这个情境中，我们用$\mathbf{F_u}$代表用户在序列中的嵌入，它是用户表示。然后，这个嵌入与目标商品的向量$\mathbf{E_i}$一起传送给预测层$\mathbf{PRED}(\cdot)$。

预测层通过$\mathbf{PRED}(\cdot)$处理这个复合特征，从中学习用户兴趣和商品相关性，生成预测 $\mathbf{H}_u = \mathbf{PRED}(\mathbf{F_u}, \mathbf{E_i})$

$\mathbf{H}_u$是融合了用户特征和商品特征后的表示$\mathbf{PRED}(\cdot)$通过深度神经网络来预测用户对下一个可能感兴趣商品的概率分布：$p(i_{n+1}|i_{1:n}) = p(\mathbf{H}_u)$

这种方法保证了推荐系统能够根据用户的动态行为和当前购物路径，提供个性化的连续推荐，提升推荐质量和用户满意度。

$$\hat{y}_{ui} = \mathbf{PRED}(\mathbf{F_u}, \mathbf{E_i})$$

预测层通常使用点积或余弦相似性✦来预测，尤其在检索场景中。

点积通过计算$\mathbf{H}_u$和$\mathbf{E_i}$的乘积来评估相似性，简单直接：$sim(\mathbf{F_u}, \mathbf{E_i}) = \mathbf{H}_u^\top \mathbf{E_i}$

余弦相似性则通过比较向量的夹角余弦值，反映它们在高维空间中的关系：

$$sim_{cosine}(\mathbf{F_u}, \mathbf{E_i}) = \frac{\mathbf{H}_u \cdot \mathbf{E_i}}{||\mathbf{H}_u||_2 ||\mathbf{E_i}||_2}$$

这两种方法都是衡量潜在兴趣匹配程度的标准，帮助推荐系统理解用户当前兴趣，从而提供更精确的个性化推荐。通过优化这些相似度，推荐系统能更准确地捕捉用户动态，提升推荐质量。

Methodology

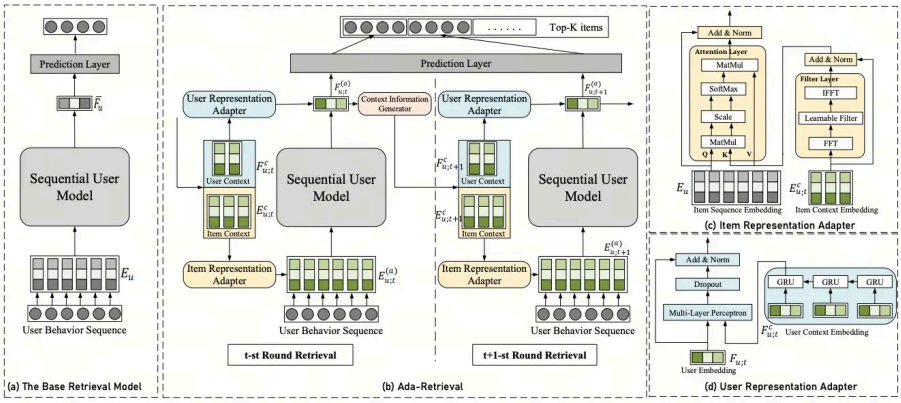


Figure 2: An overview of the traditional retrieval model (a) and our proposed Ada-Retrieval paradigm (b), which consists of two key parts: the item representation adapter (c) and the user representation adapter (d). We use colored elements to indicate the new components in Ada-Retrieval.

我们提出Ada-Retrieval模型，采用自适应多轮检索策略，如图(b)所示。该模型通过创新的适应模块------商品表示适应器(IRA)和用户表示适应器(URA)来整合环境信息，改进了序列推荐。与传统方法相比： 1. 商品表示适应器（IRA）：IRA能动态调整商品嵌入，依据用户当前状态和环境，学习适应用户对不同时间步商品兴趣的变化。 2. 用户表示适应器（URA）：URA关注用户行为随时间的动态，通过考虑历史行为、反馈等，动态更新用户自身嵌入，以反映用户兴趣的演变。

知乎

$$\mathbf{E}_{\mathbf{u}}^{(\mathbf{a})} = \mathbf{IRA}(\mathbf{E}_{\mathbf{u}}; \mathbf{E}_{\mathbf{u}}^{\mathbf{c}})$$

$$\mathbf{F}_{\mathbf{u}}^{(\mathbf{a})} = \mathbf{URA}(\mathbf{F}_{\mathbf{u}}; \mathbf{F}_{\mathbf{u}}^{\mathbf{c}})$$

1. 商品适应层（IRA）：IRA利用交互环境与商品嵌入的结合，动态更新商品表示，适应用户即时兴趣的改变，以提供更精准的推荐。
2. 用户适应层（URA）：URA追踪用户全时段行为并考虑额外反馈，动态调整用户嵌入，捕捉用户兴趣随时间的动态，增强用户模型的动态性。 通过这样的动态交互和特征优化，Ada-Retrieval模型能深化对用户行为的理解，并在推荐过程中持续改进，保证推荐的及时性和准确性，从而提升用户满意度。

Item Representation Adapter

商品适应层（IRA）的核心是根据用户当前的查询情境，对用户过去行为中的商品嵌入进行动态再校准。这个过程如图(c)所示，通过学习适应机制，IRA能捕捉用户对当前浏览商品的即时兴趣变化，从而提供更贴合用户实际需求的个性化推荐。 IRA通过优化商品嵌入，使之与用户行为环境关联更紧密，提升了推荐系统的动态响应能力和适应性。这意味着即便用户兴趣有变，IRA仍能确保推荐内容紧跟用户的新动态，保持推荐的有效性和相关性。

Learnable Filter Layer

为了应对前几轮检索中的不精确或噪声，我们设计了一个可学习的滤波块来精简商品特征。借鉴滤波增强多层感知机（Filter-Enhanced MLP）的思路，通过单个滤波块对提取的上下文商品特征进行处理。对于当前轮的上下文信息$\mathcal{C}_u^t$，首先利用编码器层$\mathbf{EMB}(\cdot)$提取特征，得到$\mathbf{E}_{\mathbf{u}}^{\mathbf{c}}$。 接着，我们运用快速傅里叶变换（FFT，用$\mathcal{F}(\cdot)$表示），对商品维度进行操作，将$\mathbf{E}_{\mathbf{u}}^{\mathbf{c}}$转换到频域，即：$\mathbf{F}_{\mathbf{u}}^{\mathbf{c}} = \mathcal{F}(\mathbf{E}_{\mathbf{u}}^{\mathbf{c}})$

这样做的目的是将复杂、可能受噪声干扰的高维上下文信息转化为频域表示，便于滤波操作，减少噪声影响，提升商品特征的清晰度。这有助于提高推荐的精准性和有效性。

$$\mathbf{X}_{\mathbf{u}}^{\mathbf{c}} = \mathcal{F}(\mathbf{E}_{\mathbf{u}}^{\mathbf{c}})$$

滤波操作通过公式 $\mathbf{Y}_{\mathbf{u}}^{\mathbf{c}} = \mathbf{X}_{\mathbf{u}}^{\mathbf{c}} \times W$

实施，它有助于滤除高频噪声，保留关键频域特征。这个过程是通过学习权重矩阵W来自动完成的，对商品上下文信息进行降噪处理。经过这个步骤，我们得到更干净的特征$\mathbf{Y}_{\mathbf{u}}^{\mathbf{c}}$，这对于提升推荐系统的精度至关重要。 这些处理过的特征随后与用户特征$\mathbf{F}_{\mathbf{u}}^{(\mathbf{a})}$结合，生成最终的推荐结果，以确保推荐的针对性和有效性。

$$\widetilde{\mathbf{X}}_{\mathbf{u}}^{\mathbf{c}} = W \odot \mathbf{X}_{\mathbf{u}}^{\mathbf{c}}$$

在Ada-Retrieval模型中，我们利用⊙操作对商品上下文的频域特征$\widetilde{\mathbf{X}}_{\mathbf{u}}^{\mathbf{c}}$进行滤波处理。通过逆傅立叶变换（IFFT），将此调制后的频谱从频域转换回时域，形成新的序列嵌入 $\mathbf{E}_{\mathbf{u}}^{\mathbf{c,new}}$

这个过程强调了时间序列随时间的变化，因为时域表示能更好地反映用户兴趣的动态。 有了这个更新的序列，它作为输入传递给序列编码层$\mathbf{SEL}(\cdot)$，使得模型能捕捉到更细致的用户行为动态，从而生成更精确的个性化推荐。通过结合频域和时间序列信息，Ada-Retrieval模型能深度理解用户行为，显著提升推荐质量。

$$\widetilde{\mathbf{E}}_{\mathbf{u}}^{\mathbf{c}} = \mathcal{F}^{-1}(\widetilde{\mathbf{X}}_{\mathbf{u}}^{\mathbf{c}})$$

在处理中$\mathcal{F}^{-1}(\cdot)$用于对逆傅立叶变换结果进行逆操作，将复数转为实数。为了预防过拟合，模型加入Dropout层，随机删除部分神经元以降低复杂性；残差连接[+]保持信息流通，保证前后层信息有效传输；归一化操作（如Layer Normalization）通过标准化每个层的输出，保持稳定性，使训练过程更平稳。 通过这些综合策略，Ada-Retrieval模型在处理商品上下文信息时，既能提取频域的抽象信息，又能保留时间序列的行为动态，从而实现对用户行为的深入分析，提供更精准的个性化推荐。

知乎

Context-aware Attention Layer

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

其中： -Q表示 Query（查询），通常是对输入数据的线性变换后的表示； -K表示 Key（键），也是对输入数据的线性变换后的表示，用于与 Query 进行比较； -V表示 Value（值），同样是对输入数据的变换，包含了所需关注的信息； -d_k是一个正则化参数，通常设置为K的维度，用于缩放相乘项，以避免数值不稳定； -$\mathbf{softmax}$操作是对 Query 和 Key 矩阵元素的归一化指数函数，使得结果是对所有可能的 Key 元素加权求和，其值越大，表示对应 Value 的重要性越高； -V的最终输出是这个加权和，反映了Query与Key的相关性，然后与Value进行结合，生成最终的注意力输出。 这个公式在自然语言处理和计算机视觉等领域广泛应用，特别是在Transformer模型中，它帮助模型关注输入的不同部分，以生成更准确和相关的输出。

$$\mathbf{Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V}$$

在这个上下文中$\mathbf{Q}$代表商品序列的特征向量$\mathbf{K}$代表商品上下文的特征向量$\mathbf{V}$也同样是特征向量。为了处理不同来源的特征，我们引入了$\sqrt{d}$来控制内积的规模，以防止数值不稳定。我们定义了一个名为

$$\mathbf{ContextAwareAttention}$$

的注意力机制，它基于上下文计算：

$$\mathbf{ContextAwareAttention(Q, K, V) = softmax}\left(\frac{\mathbf{QK}^T}{\sqrt{d}}\right)\mathbf{V}$$

这个公式中，查询和键的相乘不直接来源于单一源，而是基于商品上下文的特性，这样模型能够更精准地理解用户行为动态，从而在 Ada-Retrieval 中实现更精确的个性化推荐。

$$\widetilde{\mathbf{H}}_{\mathbf{u}}^{\mathbf{c}} = \mathbf{Attention}(\mathbf{E}_{\mathbf{u}}, \mathbf{H}_{\mathbf{u}}^{\mathbf{c}}, \mathbf{H}_{\mathbf{u}}^{\mathbf{c}})$$

NormAtt 操作对特征向量$\mathbf{H}_{\mathbf{u}}^{\mathbf{c}}$应用 Layer Normalization，这是一种归一化方法，随后添加 Dropout 层以防止过拟合。这两种技术的组合有助于保持模型参数的稳定，确保信息在神经网络中顺畅流动。在 Ada-Retrieval 中，这有助于提高推荐的准确性和个性化程度。

$$\mathbf{E}_{\mathbf{u}}^{(\mathbf{a})} = \mathbf{LayerNorm}(\mathbf{E}_{\mathbf{u}} + \mathbf{Dropout}(\widetilde{\mathbf{H}}_{\mathbf{u}}^{\mathbf{c}}))$$

A Case with Sequential User Model

在Ada-Retrieval中，对调整过的商品序列特征$\mathbf{E}_{\mathbf{u}}^{(\mathbf{a})}$，使用Transformer（如SASRec中的TRFM）进行处理，以获取用户特征表示$\mathbf{F}_{\mathbf{u}}$。这里$\mathbf{TRFM}$是SASRec的Transformer模型，用于捕获序列信息。然而，与SASRec不同，Ada-Retrieval不修改TRFM的内部参数，而是利用$\mathbf{E}_{\mathbf{u}}^{(\mathbf{a})}$，即上下文信息动态地调整输入，以提供更个性化的推荐。这种方法保持了模型的结构不变，但通过融合频域和时间序列信息，提高了推荐的精度和个性化程度。

User Representation Adapter

我们构建了用户表示适应器，利用用户上下文信息，目的是生成适应用户当前情境的特征表示，如图(d)所示。这个过程不改变原有的用户特征计算方式，而是动态地调整用户特征$\mathbf{E}_{\mathbf{u}}$，使其能更好地捕捉到具体环境下的需求，从而提升推荐的针对性和有效性。

Gated Recurrent Unit Layer

在推荐系统中，我们利用RNNs，特别是GRUs（Gated Recurrent Units），因其擅长处理变长序列数据，这种方法在该领域取得了显著效果。GRUs通过门控机制解决了传统RNN中的梯度消失问题，允许过去信息逐步但平滑地影响当前状态，而非立即消退。 简而言之，我们利用GRU单元来动态地整合过去的用户特征，这样做的目的是让模型能够随着时间的推移，累计并有效地学习用户

知乎

$$\widetilde{\mathbf{F}}_{\mathbf{u}}^{\mathbf{c}} = \mathrm{GRU}(\mathbf{F}_{\mathbf{u}}^{\mathbf{c}})$$

用户上下文的特征表示$\mathbf{F}_{\mathbf{u}}^{\mathbf{c}}$是通过整合前面几轮处理的用户特征$\mathbf{E}_{\mathbf{u}}^{(\mathbf{a})}$得到的，这是一种对用户行为序列信息的综合体现。为了简化，我们直接使用GRU的最终隐藏状态作为用户上下文的简化版本，记作$\widetilde{\mathbf{F}}_{\mathbf{u}}^{\mathbf{c}}$这包含了过去行为对当前推荐的动态影响，但去除了复杂的上下文特征。这样做的目的是保持模型对用户行为模式的理解，同时保持计算效率。

Multi-Layer Perceptron Layer

在获得了用户上下文简化表示$\widetilde{\mathbf{F}}_{\mathbf{u}}^{\mathbf{c}}$后，它与当前用户特征$\mathbf{F}_{\mathbf{u}}$合并，生成一个融合了全局和局部信息的用户特征。这个组合通过多层感知器（MLP）进行非线性处理，MLP由多层隐藏层组成，使用ReLU激活函数来促进正向信息的传递。经过此步骤，得到的用户上下文特征$\mathbf{H}_{\mathbf{u}}^{\mathbf{ctx}}$作为用户适应器（URA）的输入，进一步提升了Ada-Retrieval模型对用户行为序列的深度理解和个性化推荐效果。

$$\mathbf{F}_{\mathbf{u}}^{(\mathbf{a})} = W_2\,\mathrm{ReLU}(W_1[\widetilde{\mathbf{F}}_{\mathbf{u}}^{\mathbf{c}}; \mathbf{F}_{\mathbf{u}}] + b_1) + b_2$$

Context Information Generator

为累积每轮产生的用户上下文特征，我们采用了堆叠（stacking）策略，将这些信息组合成一个包含多轮信息的上下文向量序列。

$$\mathbf{F}_{\mathbf{u;t}}^{\mathbf{c}} = \mathrm{STACK}(\{\mathbf{F}_{\mathbf{u;1}}^{(\mathbf{a})}, \mathbf{F}_{\mathbf{u;2}}^{(\mathbf{a})}, \cdots, \mathbf{F}_{\mathbf{u;t-1}}^{(\mathbf{a})}\})$$

此外，从候选商品池中，我们依据与当前轮用户特征$\mathbf{F}_{\mathbf{u}}^{(\mathbf{a})}$的相关性，选取排名最前的$k$个商品，提取它们的项ID并加入到项上下文池中。这样做的目的是为了聚焦于最相关的商品，增强Ada-Retrieval模型的针对性和推荐准确性。

$$\mathcal{C}_u^t = \mathcal{C}_u^{t-1} + \text{top-}k\,\underset{i\in\mathcal{I}}{\mathrm{argmax}}\,\mathrm{Sim}(\mathbf{F}_{\mathbf{u;t-1}}^{(\mathbf{a})}; \mathbf{E}_{\mathbf{i}})$$

在这个过程中，我们定义了一个衡量用户和商品特征相似度的函数Sim，通常采用点积来计算。关键点是，基于这个相似度计算的商品上下文，会被用于后续的嵌入查找层，以获取它们对应的特征表示。这样做是为了确保推荐的针对性和准确性，通过聚焦于最相关的商品信息。

Model Prediction and Optimization

在总共进行了T轮迭代后，Ada-Retrieval生成了对应于每个用户的T个上下文用户表示

$$\{\mathbf{F}_{\mathbf{u;1}}^{(\mathbf{a})}, \mathbf{F}_{\mathbf{u;2}}^{(\mathbf{a})}, \ldots, \mathbf{F}_{\mathbf{u;T}}^{(\mathbf{a})}\}$$

这些用户表示与物品嵌入矩阵$\mathbf{E}$通过点积运算来评估候选项的关联性：

$$\hat{y}_{ui} = \mathbf{F}_{\mathbf{u;T}}^{(\mathbf{a})\top}\mathbf{E}$$

这里$\hat{y}_{ui}$代表用户u在第T轮对项目i的实际兴趣评分估计。这个预测值量化了用户在当前互动时刻对每个潜在项潜在兴趣的强度，从而为个性化的推荐提供了依据。

$$\hat{y}_{ui;t} = \mathbf{E}_{\mathbf{i}}^{\mathbf{T}}\mathbf{F}_{\mathbf{u;t}}^{(\mathbf{a})}$$

在每次迭代的T轮结束后，每用户生成T个用户特征表示

$$\mathbf{F}_{\mathbf{u;1}}^{(\mathbf{a})}, \mathbf{F}_{\mathbf{u;2}}^{(\mathbf{a})}, \ldots, \mathbf{F}_{\mathbf{u;T}}^{(\mathbf{a})}$$

然后，通过与物品嵌入矩阵$\mathbf{E}$的点积，计算与每个项的预测兴趣分数

$$\hat{y}_{ui} = \mathbf{F}_{\mathbf{u;T}}^{(\mathbf{a})\top}\mathbf{E}$$

知乎

$$L_t = -\sum_{u,i} \log(\hat{y}_{ui}) \cdot 1[i \in \text{top}_k(u)]$$

这里$1[\cdot]$是指示函数，当项i位于用户u的前k个最相关项时为1，否则为0。通过最大化这个损失，模型试图更准确地预测用户偏好，从而提升推荐的精确度。

$$\mathcal{L}_t = -\sum_{u\in\mathcal{U},i\in\mathcal{I}} y_{ui;t} \log(\sigma(\hat{y}_{ui;t})) + (1 - y_{ui;t}) \log(1 - \sigma(\hat{y}_{ui;t}))$$

为了更注重对正面商品的早期预测，我们引入了衰减因子λ，它对每轮训练的损失进行加权，调整了总损失。优化目标函数变为：

$$L_{t,\text{weighted}} = \lambda_t \cdot L_t = \lambda_t \cdot \left(-\sum_{u,i} \log(\hat{y}_{ui}) \cdot 1\left[i \in \text{top}_k(u)\right]\right)$$

其中λ_t是根据轮次动态调整的权重，它的值会随着训练的进行逐渐减小，这样可以确保早期预测的准确性得到更多重视。通过这种方式，模型在训练过程中能够更早地聚焦于用户的真实需求，从而提升推荐的准确性。

$$\mathcal{L} = \sum_{t=1}^{T} \lambda^t \mathcal{L}_t$$

为提升训练效率，我们采取分步的两阶段策略：首先，对基础序列模型进行基础预训练；接着，对Ada-Retrieval进行微调，利用预训练模型作为起点。在第二阶段，我们同时更新Ada-Retrieval参数Θ和基础模型参数Φ。Ada-Retrieval具有兼容性，能与GRU和GNN等检索模型集成，保持其结构但性能增强，不适用于依赖用户-项目交互矩阵的矩阵分解（MF）模型，因其优化目标和方法与Ada-Retrieval有异。

Experimental Settings

Datasets.

为了检验方法的泛化能力，我们在三个公开来源的数据集------Beauty、Sports和Yelp上评估模型。Beauty和Sports数据源自Amazon产品评论，包含用户对商品的反馈；而Yelp是基于业务推荐的扩展项目序列，来源于2019年后的交易记录。为保持一致性，我们按用户或会话对数据进行分组，并按时间顺序整理。为确保数据质量，我们删除了互动次数少于5次的记录。

Dataset	Beauty	Sports	Yelp
# Sequences	22,363	25,598	30,431
# Items	12,101	18,357	20,033
# Actions	198,502	296,337	316,354
# Sparsity	99.93%	99.95%	99.95%

Table 1: Statistics of datasets after preprocessing.

Evaluation Settings.

为了全面评估模型，我们采取一种特殊的交叉验证策略，对每个用户的行为序列使用留一法进行划分，形成训练集、验证集和测试集。区别于常规的抽样方法，我们的方法涵盖所有未被用户实际访问过的商品作为潜在选项。主要的评价指标包括精确率指标，如top-k点击率（Hit Ratio@k），以及top-k归一化折现累积增益（Normalized Discounted Cumulative Gain@k，NDCG@k），用来度量模型在预测用户兴趣上的准确性和召回率。

Implementation Details.

32GB内存。训练参数包括使用Adam优化器，初始学习率0.001，批大小1024。数据处理设定：序列长度限制为50，嵌入维数为64，最大训练轮数为200。针对Ada-Retrieval，我们调整的参数T和λ分别在3到8及0.1到0.9的范围内，步长为1和0.2。实验重复五次，以平均值和标准差展示结果。若在连续10个验证集的HR\@50指标下滑，我们会采用早停策略来控制训练过程。

Main Results with Various Backbone Models

Backbones.

为了全面评估Ada-Retrieval的有效性，我们将其与几种典型的序列推荐模型进行比较，包括GRU4Rec、SASRec、NextItNet、SRGNN和FMLPRec。这些模型涵盖了不同的结构，如循环神经网络（RNN）、卷积神经网络（CNN）、图神经网络（GNN）以及多层感知机（MLP），以展现其多样性和代表性。

Results.

Datasets	Models	GRU4Rec		SASRec		NextItNet		SRGNN		FMLPRec	
		HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
Beauty	Base	13.126	4.574	17.110	6.506	12.539	4.064	12.411	4.242	17.935	6.876
	Ada-Retrieval	14.175	4.915	17.741	7.062	12.948	4.288	13.274	4.375	18.531	7.265
	Improv.	**↑7.99%**	**↑7.46%**	**↑3.69%**	**↑8.55%**	**↑3.27%**	**↑5.50%**	**↑6.96%**	**↑3.12%**	**↑3.32%**	**↑5.66%**
Sports	Base	7.644	2.447	10.924	4.046	7.939	2.563	7.704	2.504	11.607	4.238
	Ada-Retrieval	8.226	2.683	11.352	4.234	8.425	2.663	8.551	2.731	11.903	4.444
	Improv.	**↑7.61%**	**↑9.67%**	**↑3.92%**	**↑4.65%**	**↑6.12%**	**↑3.90%**	**↑11.00%**	**↑9.07%**	**↑2.55%**	**↑4.85%**
Yelp	Base	9.252	2.765	12.062	3.770	9.828	2.971	10.501	3.141	13.013	4.029
	Ada-Retrieval	10.415	2.985	12.637	3.852	11.224	3.316	11.688	3.454	13.430	4.157
	Improv.	**↑12.57%**	**↑7.98%**	**↑4.77%**	**↑2.19%**	**↑14.20%**	**↑11.60%**	**↑11.31%**	**↑9.96%**	**↑3.20%**	**↑3.18%**

Table 2: Top-50 performance comparison of five backbone models and Ada-Retrieval on three datasets. All the metrics in the table are percentage numbers with '%' omitted.

在三个数据集上，我们对基础序列模型及对应的Ada-Retrieval模型进行了训练，结果显示，无论在哪种情况和指标下，Ada-Retrieval都明显超越了所有基础模型。具体来说，与GRU4Rec相比，Ada-Retrieval（GRU4Rec）在NDCG\@50上的提升平均达8.37%，而对于SASRec，提升幅度为5.57%。不论模型基于RNN（如GRU4Rec）、Transformer（如SASRec）、CNN（如NextItNet）、GNN（如SRGNN）还是MLP（如FMLPRec），Ada-Retrieval都能无缝融入并提升性能。重要的是，它能适应各种结构，通过附加到任何基线模型的适应模块，保持原架构不变，展现出强大的适应性和插件式特性。

Comparison with Multi-Interest Models

Baselines.

我们运用多轮自适应学习，创建多维用户表示，这种方法虽非特指某一研究领域，但在生成多用户视角上与多兴趣感知用户建模有相似。对比中，我们把Ada-Retrieval与几种多兴趣检索模型做基线研究，包括**DNN（YouTube DNN）**，**MIND**，**ComiRec**和**SINE**。

Results.

Methods	Beauty		Sports		Yelp	
	HR	NDCG	HR	NDCG	HR	NDCG
DNN	13.705	4.726	8.798	2.890	11.241	3.317
MIND	14.045	5.002	8.888	2.918	11.320	3.443
ComiRec	14.394	5.232	9.270	3.250	11.479	3.523
SINE	13.191	4.325	9.087	2.978	12.091	3.724
Ada.	**17.741**	**7.062**	**11.352**	**4.234**	**12.637**	**3.852**

Table 3: Top-50 performance comparison of several baselines and Ada-Retrieval (SASRec) on three datasets.

知乎

性和深度，体现了其智能化。

编辑于 2024-05-22 11:02 · IP 属地北京

微软（Microsoft） 搜索引擎 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

Linux命令行如何使用图形化软件

wangt... 发表于基因学院

SQL SERVER 2012/2014 序列号密钥

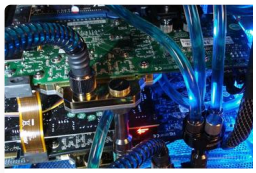
SQL SERVER 2012/2014 序列号密钥 MICROSOFT SQL SERVER 2012 DEVELOPER 版 序列号: YQWTX-G8T4R-QW4XX-BVH62-GP68Y MICROSOFT SQL SERVER 2012 ENTERPRISE SERVER/CAL...

林老师之家

CMake识别操作系统平台及Linux发行版本

识别操作系统平台以下代码可以识别 Windows、Linux、Macos三种类型的操作系统平台： IF (CMAKE_SYSTEM_NAME MATCHES "Linux") MESSAGE(STATUS "curren...

guoti... 发表于技术闲谈



如何在 Linux 中查看系统制造商、型号和序列号

Linux... 发表于