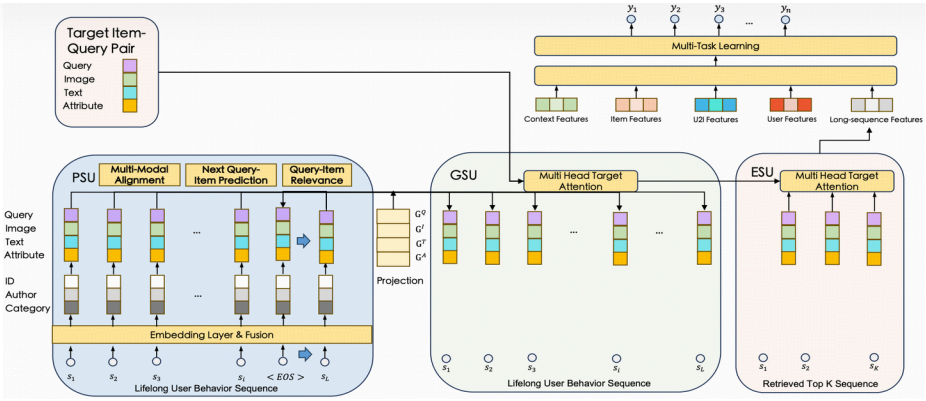


阿里巴巴 2024：利用多模态增强搜索优化长期序列建模

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

Introduction

用户行为建模在现代商业推荐系统中至关重要，涵盖在线电子商务平台如亚马逊、淘宝、支付宝以及内容平台如YouTube、抖音。近年来，随着用户在线购物和观看短视频的时间显著增长，用户历史行为的长度已从几百（10^2）激增至数万以上。

许多近期的研究聚焦于建模用户长期的行为，如ETA、WIN、QIN等。这些模型遵循分层的两阶段架构，首先利用目标内容或目标查询作为触发器，从历史行为中检索出最相关的前K项行为。在第二阶段，它使用target-attention来编码选定的行为作为用户兴趣的表示。 现有的长期行为建模中一个被忽视的问题是ID特性在长期序列中的学习不足问题，如许多在当前训练数据集中找不到的历史内容ID、作者ID等。这些随机初始化的低频ID嵌入无法通过有限的训练数据得到良好的学习，从而损害了target-attention计算的准确性。

现有的长期序列建模的第二个问题是，它无法很好地处理序列中内容的一系列多模态属性，如文本和图像特征。如果不同模态的向量在相同的嵌入空间中未正确对齐，它们的范数值会有所不同。现有的目标内容attention计算使用查询和键的内积，这可能被具有大范数值的模态向量所主导。例如，目标内容只会从历史行为中检索出最相关的视觉上但语义上非常不同的内容，这将影响推荐的在线性能。

为了应对这些问题，我们提出了一种名为搜索增强的多模态兴趣网络和近似检索（SEMINAR）的新模型，用于模拟用户长期的多模态行为。用户的历史行为包括不同类型的异构行为，既涉及浏览内容序列，也包括搜索查询序列。我们将用户的搜索查询序列与浏览内容序列统一为一个查询-内容对的联合序列，这个序列可以灵活地通过目标内容或目标搜索查询在CTR预测任务和个人化搜索排名（PSR）任务中进行检索。

SEMINAR提出了一种预训练搜索单元（PSU）网络，用于学习历史多模态查询-内容对的长期行为序列。它引入了多个预训练任务，旨在解决历史ID特征的学习不足问题和多模态对齐的问题。在后续任务中，target-attention模块从PSU恢复学习到的内容表示，使用预训练的ID嵌入作为初始化，并应用一个投影权重矩阵来获取行为序列的转换表示。

在实时服务中，计算精确的注意力使用多模态向量的内积，其时间复杂度为O(L×M×d)，这会消耗大量时间，其中L表示序列长度，M表示模态数量，d表示嵌入维度。 与现有的近似检索方法，如局部敏感哈希（LSH）和层次化可导航小世界（HNSW）不同，我们利用多模态环境中的产品量化策略，并将多模态内容表示转化为量化码书中的离散整数代码，然后通过子向量中心的内积求和来近似精确注意力计算。在实时服务中，注意力计算等同于预计算的距离表查找和求和操作，这些操作可以高效地执行。

Proposed Model

Problem Formulation

对于搜索行为，用户输入一个查询$q \in \mathcal{Q}$，并与与该查询相关的少数内容进行交互（点击或查看），产生对齐的查询和内容对序列(q_l, i_l)。对于浏览推荐内容的行为，用户浏览一系列内容，没有明确的搜索意图，我们将空查询$q = \emptyset$添加到每个内容中，以获得对齐的查询-内容对$(q_l = \emptyset, i_l)$。最后，我们以时间顺序构造一个统一的对齐查询-内容对序列，长度为L，表示为$\{(q_l, i_l)\}_{l=1:L}$在一些推荐场景中，例如YouTube和抖音的短视频推荐，每个内容都有多模态特征，包括文本（视频标题）、图像和属性（作者和类别）。

我们进一步将浏览的内容序列$\mathcal{B}$划分为M个多模态子序列，包括文本特征序列$\mathcal{T} = \{T_1, T_2, \ldots, T_L\}$，图像特征序列$\mathcal{I} = \{I_1, I_2, \ldots, I_L\}$和属性特征序列$\mathcal{A} = \{A_1, A_2, \ldots, A_L\}$

最后，多模态查询-内容对序列$[\mathcal{Q}, \mathcal{T}, \mathcal{I}, \mathcal{A}] \in \mathbb{R}^{(M+1)\times L\times d}$表示输入到SEMINAR模型中，其中$d$表示对齐表示的维度。

Aligned Lifelong Sequence of Multi-Modal Query-Item Pairs

多模态查询项对的对齐序列通过嵌入层。我们用$[\mathbf{x}_l = (\mathbf{x}_l^{query}, \mathbf{x}_l^{item})]_{l=1:L}$表示历史序列中的查询和项对:

$$\mathbf{x}_l^{query} \in \mathbb{R}^d$$

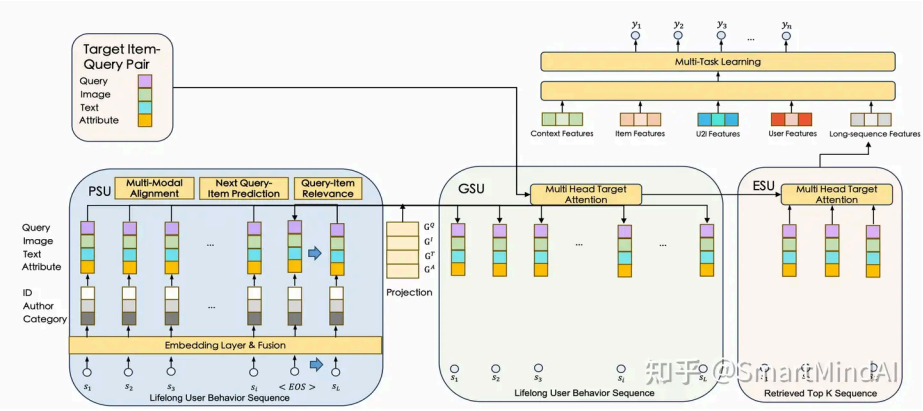
$$\mathbf{x}_l^{item} = (\mathbf{x}_l^{text}, \mathbf{x}_l^{image}, \mathbf{x}_l^{attributes}) \in \mathbb{R}^{M\times d}$$

在CTR预测中，target-attention（TA）是一种结构，它使用目标项从历史行为序列中检索最相关项。我们从目标项扩展到目标查询项对，从历史序列中检索最相关的前K对。我们表示的目标查询项对为

$$\mathbf{x}_t = (\mathbf{x}_t^{query}, \mathbf{x}_t^{text}, \mathbf{x}_t^{image}, \mathbf{x}_t^{attributes})$$

SEMINAR Model Architecture

我们的模型SEMINAR在图中引入了一个新的网络层（PSU），用于使用长期序列的多模态查询项对数据集进行预训练。



Pretraining Search Unit

PSU的输入为查询项对的对齐序列，表示为$[\mathbf{x}_l]_{l=1:L}$。其中L表示对齐序列的长度

$$\mathbf{x}_l = (\mathbf{x}_l^{query}, \mathbf{x}_l^{text}, \mathbf{x}_l^{image}, \mathbf{x}_l^{attributes})$$

代表序列中的第l个行为，包括查询嵌入和多模态嵌入。查询$q \in \mathcal{Q}$通过查询特征编码器$f(.)$进行编码，结果为

$$Q = f(\mathbf{x}_{1:L}^{query}) \in \mathbb{R}^{L\times d_{query}}$$

知乎

$$T = \text{Encoder}_{text}(\mathbf{x}_{1:L}^{text}) \in \mathbb{R}^{L \times d_{text}}$$

并使用视觉transformer对图像特征进行编码，表示为

$$I = \text{Encoder}_{image}(\mathbf{x}_{1:L}^{image}) \in \mathbb{R}^{L \times d_{image}}$$

此外，我们使用函数$g(.)$对属性特征进行编码，属性特征

$$A = g(\mathbf{x}_{1:L}^{attribute}) \in \mathbb{R}^{L \times d_{attribute}}$$

被视为子序列的一个通道，参与对内容序列的多模态对齐。为了将不同通道的表示投影到相同的维度d，我们进一步将它们乘以线性权重矩阵

$$\{W_q, W_t, W_i, W_a\}$$

并得到多模态查询-内容对的堆叠输入序列如下：

$$\mathbf{x} = [QW_q, TW_t, IW_i, AW_a] \in \mathbb{R}^{(M+1) \times L \times d}$$

PSU是设计一个预训练网络来学习长期行为序列中的内容，并且这个预训练网络应该与ETA和TWIN等下游模型的级联两阶段结构共享相同的多头target-attention力结构。下游模型将预训练的查询和内容嵌入作为参数的初始化，并对网络进行微调。

与BERT中的掩码语言模型（MLM）不同，MLM使用上下文窗口中的标记来预测被掩码*的标记，我们使用下一个对预测作为预训练任务来预测正确的最后一个查询和内容对。我们在序列中省略最后一个m为L的查询-内容对，并在序列$\mathbf{x}_{1:L-1}$的末尾加上一个特殊的结束标记$< EOS >$。为了将序列长度*从$L-1$增加到L，我们进一步填充序列。

下一个查询-内容对预测任务被定义为分类任务：

$$y = p(\mathbf{x}_L^{1:M+1} | \mathbf{x}_{1:L-1}^{1:M+1}; \mathbf{x}^{EOS})$$

并带有损失函数*$\mathcal{L}_{next}^{pair}$。正确的最后一个对被赋予正label，而负label则被分配给从负采样查询-内容对中选择的查询-内容对。

$$\mathbf{x} = \lambda \mathbf{x}^{query} + (1-\lambda) \sum_i w_i \mathbf{x}^{item^{(i)}} = \sum_{m \in M+1} \gamma_m \mathbf{x}^{(m)}$$

λ和$(1-\lambda)$分别表示用于合并查询向量和内容向量表示的权重，其中$\lambda \in [0, 1]$。为了简化符号，我们使用单个向量

$[\gamma_m]^{1:M+1} \in \mathbb{R}^{M+1}$ 来表示所有$(M+1)$通道的权重，并且所有权重之和为1。 权重向量*γ_m可以通过门控网络的softmax输出动态地学习。

注意力计算为合并多通道表示的查询和键进行内积运算,最终的注意力分数主要由大范数值$x^{(m)}$和大权重γ_m的模态决定，其他模态的信息容易被忽略。q_t表示目标查询-内容对的表示，即

$$q_t = \sum_i \gamma_i \mathbf{x}_t^{(i)} = \sum_i \gamma_i |\mathbf{x}_t^{(i)}| \hat{\mathbf{x}}_t^{(i)}$$

其中$|\mathbf{x}_t^{(i)}|$表示目标内容第i个通道的向量长度$\hat{\mathbf{x}}_t^{(i)}$是一个单位向量。 同样地，我们可以将第l个历史行为$k_l \in K$表达为

$$k_l = \sum_j \gamma_j \mathbf{x}_l^{(j)} = \sum_i \gamma_j |\mathbf{x}_l^{(j)}| \hat{\mathbf{x}}_l^{(j)}$$

在多头注意力中，第h个头的表示为 $head^{PSU_h} = \text{Attention}_h(q_t, K^{PSU}, V^{PSU})$

注意力分数a_h^{PSU}通过d维向量查询和键的内积运算*，并乘以缩放因子$\frac{1}{\sqrt{d}}$来计算。

$$\alpha_h^{PSU} = \frac{(q_t W_h^{PSU_Q})(K^{PSU} W_h^{PSU_K})^T}{\sqrt{d}}$$

知乎

$\gamma_{ij} = \gamma_i \gamma_j |\mathbf{x}_i^{PSU(i)}||\mathbf{x}_l^{PSU(j)}$

在多模态嵌入空间中学习序列中PSU的多模态嵌入：$[\mathbf{x}_l^{PSU(1:M+1)}]_{l=1}^{L} \in \mathbb{R}^{L\times(M+1)\times d}$ 表示的是PSU的多模态嵌入。而：$\mathbf{x}^{PSU(1:M+1)} = [Q, T, I, A]$则代表了PSU的特定组合。

输入序列的合并表示为：

$K^{PSU} = [\sum_i \gamma_i \mathbf{x}_l^{PSU(i)}]_{l=1}^{L} \in \mathbb{R}^{L\times d}$

第h个头部的查询投影权重矩阵为：$W_h^{PSU_Q} \in \mathbb{R}^{d\times d}$

而键投影权重矩阵为：$W_h^{PSU_K} \in \mathbb{R}^{d\times d}$，$\gamma_{ij}$是序列中查询向量和键向量之间的交叉模态交互权重。$\gamma_{ij}$等于$\gamma_i$和$\gamma_j$的标量乘积，以及查询向量$|\mathbf{x}_i^{PSU(i)}|$和键向量$|\mathbf{x}_l^{PSU(j)}|$的范数值。

多模态对齐是一个至关重要的任务，它在相同的嵌入空间中学习多模态表示。 我们同时训练多模态对齐任务，包括文本-图像、图像-属性、文本-属性，使用N对的交叉熵[+]损失。

$\mathcal{L}_{align} = \sum_{i\in M} \sum_{j\in M\neq i} \mathcal{L}_{\text{CLIP}}(\hat{\mathbf{x}}_{1:L}^{(i)}, \hat{\mathbf{x}}_{1:L}^{(j)}), (i,j) \in \{T, I, A\}$

序列长度 L通常是巨大的，对齐的复杂度为$O(L^2)$。为了降低复杂度，我们进一步将序列分为N_{ch}个片段。每个片段是一个子序列，长度为$L_{sub} = \frac{L}{N_{ch}}$。片段内的对齐损失是多个子序列损失的和，定义为

$\mathcal{L}_{\text{CLIP}}(\hat{\mathbf{x}}_{L_k:L_{k+1}}^{(i)}, \hat{\mathbf{x}}_{L_k:L_{k+1}}^{(j)})$

其复杂度降低至$O(L^2/N_{ch})$。

另外，查询内容相关性预测是典型的搜索任务，通常通过二元分类来预测序列中的正确查询-内容对，从不相关的查询-内容对中预测出正确的对。每个查询和内容的表示为

$[\mathbf{x}_l^{query}; \mathbf{x}_l^{item} = \sum_{m\in M} \gamma_m \mathbf{x}_l^{item^{(m)}}]$

查询-内容二元分类任务的损失为

$\mathcal{L}_{query-item} = \sum \mathcal{L}_{ce}(y_l^{qi}; \mathbf{x}_l^{query}, \sum_{m\in M} \gamma_m \mathbf{x}_l^{item^{(m)}})$

其中y_l^{qi}表示序列中第l对的关联label。正确匹配的查询-内容对被标记为正例，随机选择的不匹配查询-内容对被标记为负例。

预训练搜索单元（PSU）的目标包括以下三部分：下个查询-内容对预测损失$\mathcal{L}_{next}^{pair}$、多模态对齐损失$\mathcal{L}_{align}$以及查询-内容相关性预测损失$\mathcal{L}_{query-item}$。因此

$\mathcal{L}_{PSU} = \mathcal{L}_{next}^{pair} + \mathcal{L}_{align} + \mathcal{L}_{query-item}$

Fine-tuning the projection weight

SEMINAR模型中的通用搜索单元（GSU）与PSU中的结构共享了相同的级联两阶段范式中的多头target-attention力体系结构，即 $head^{GSU_h} = \text{Attention}_h(q_t, K^{GSU}, V^{GSU})$

GSU从PSU恢复预训练的嵌入，并应用特定的投影权重矩阵 $G^{(j)} \in \mathbb{R}^{d\times d}$ 到预训练的嵌入中，以得到在GSU中的投影嵌入$\mathbf{x}^{GSU(j)}$。$E^{PSU(*)}$表示预训练的多模态嵌入。在第一阶段检索后，序列长度从L减小到K。

在第二阶段的ESU中，多头target-attention力体系结构同样与GSU和PSU共享，即

$head^{ESU_h} = \text{Attention}(q_t, K^{ESU}, V^{ESU})$

且每个头有其特定的投影权重矩阵

知乎

$$\alpha_h^{GSU} = \frac{(q_t W_h^{GSU_Q})(K^{GSU} W_h^{GSU_K})^T}{\sqrt{d}}$$

$$\mathbf{x}^{GSU(j)} = \mathbf{x}^{PSU(j)} G_j, \forall j \in M+1$$

比较GSU的注意力α_h^{GSU}与预训练的PSU注意力α_h^{PSU}，我们可以看出，多头target-attention的结构一致，相同。每个头在多头注意力中的查询权重$W_h^{GSU_Q}$和键权重$W_h^{GSU_K}$与$W_h^{PSU_Q}$和$W_h^{PSU_K}$不同。而嵌入投影权重矩阵G_j仅GSU拥有。

在第二步，从GSU中选取的前K个相关查询-内容对被输入到精确搜索单元（ESU）中，ESU中的第h个头部为 $\mathbf{Attention}_h(q_t, K^{ESU}, V^{ESU})$

在ESU中，ESU中的K个最高相关项为$\mathbf{TopK}(K^{GSU})$，其维度为$(M+1) \times K \times D$。ESU中的注意力分数为α_h^{ESU}，ESU中的ID嵌入为$\mathbf{x}^{ESU(j)}$。

$$\alpha_h^{ESU} = \frac{(q_t W_h^{ESU_Q})(K^{ESU} W_h^{ESU_K})^T}{\sqrt{d}}$$

$$\mathbf{x}^{ESU(j)} = \mathbf{x}^{GSU(j)} = \mathbf{x}^{PSU(j)} G^{(j)}, \forall j \in M+1$$

最后，计算用户终生序列表示 $\mathbf{x\_lifelong\_seq}$，为：
$\mathbf{Concat}(\mathbf{head}^{ESU\_1}, \ldots, \mathbf{head}^{ESU\_H}) W^{ESU}$

然后 $\mathbf{x\_lifelong\_seq}$ 与用户、内容、用户-内容交互（u2i）和上下文特征进行拼接，并参与CTR预测。

$$\hat{y}\_i = f\_\theta\_i(\mathbf{x\_lifelong\_seq}, \mathbf{x}\_u, \mathbf{x}\_i, \mathbf{x\_u2i}, \mathbf{x\_context})$$

表示预测值y_i表示实际label值。CTR预测的最终损失为

$$\mathcal{L}\_ctr = \sum\_i \mathcal{L}\_ce(y\_i, \hat{y\_i})$$

Approximate Retrieval of Multi-Modal Query-Item Pair

目标查询项q_t与第l个查询项行为k_l之间的精确注意力分数计算为多个向量的加权内积：

$$\langle w_{q_t} \cdot v_{q_t}, w_{k_l} \cdot v_{k_l} \rangle$$

$$q_t{}^T k_l = (\sum_{i \in M+1} \gamma_i x_t^{(i)})^T (\sum_{j \in M+1} \gamma_j x_l^{(j)}), \forall l \in \{1, 2, \ldots L\}$$

精确计算的时间复杂度为O(LMd)。其中，L表示序列长度，M表示维度为d的多模态嵌入向量的加权求和操作次数。在多模态查询-内容对序列的长期设置中，当L非常大（10\^4）时，计算变得耗时。 快速检索给定输入查询向量q的K个最近向量的一个简单方法是构建一个嵌入索引，例如HNSW，并进行ANN（近似最近邻居）搜索。然而，在多模态查询-内容对序列中检索目标查询-内容对时，构建嵌入索引存在困难。

考虑的另一种分层跨模态策略是检索前K个相关查询-内容对。首先，我们为查询集构建两个单独的向量索引，大小分别为Q和B。在实时检索目标查询-内容对时，我们进行四次向量检索，包括查询到内容、查询到查询、内容到查询和内容到内容。每次检索都保留具有最大内积的前K个内容。第一阶段跨模态检索的过滤器为L → 4K。给定潜在的四个K个内容，我们对这些内容进行精确全面注意力计算以获得最终的前K个内容，过滤器为4K → K。

分层跨模态检索策略*的问题在于，它可能无法达到与精确全面注意力计算相比较的最优解。这是因为最终的内积是所有模态的加权平均。此外，一个模态中排名前K的相关内容（例如，查询到查询相关性）在其他模态（如查询到内容（文本）或查询到内容（图像））中可能具有非常低的相关性，因此总体内积得分并不理想。为了帮助提高召回性能同时兼顾检索速度，关键在于减少查询集$\mathcal{Q}$和内容集$\mathcal{B}$的基数。我们发现产品量化是一种好的近似策略，它将向量分解为N_bit个子向量，并为每个子向量分配最近的中心，从而达到基数的减少。

向量

$$q(\mathbf{x}^{(m)}) = [c^{(m)}\_1, c^{(m)}\_2, \ldots, c^{(m)}\_N\_bit] \in \mathbb{R}^{N\_bit}$$

每个多模态查询-内容对的表示为：

$$[\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)}] \rightarrow [q(\mathbf{x}^{(1)}), \ldots, q(\mathbf{x}^{(M)})] \in \mathbb{R}^{M \times N_{bit}}$$

我们预先计算不同聚类中心的内积，并将值存储在内存中。存储空间复杂度为O(M平方乘以C的平方乘以比特数），其中M表示多模式的大小，C表示聚类的数量，比特数表示代码库中模式m被子向量分裂的数量。

在线服务时，在服务中$q_t{}^T k_l$的内积相当于执行了O(M平方乘以比特数)次距离查找操作。最终得分通过计算这些距离的加权和得出。其中，c\^(i)表示第i个q_t向量的第b个子向量的聚类中心ID，c\^(j)表示第j个x_l向量的第b个子向量的聚类中心ID。

$$q_t{}^T k_l = \sum_i \sum_j \gamma_i \gamma_j \mathbf{x}_t^{(i)} \mathbf{x}_l^{(j)} \approx \sum_i \sum_j \gamma_i \gamma_j \sum_{b \in N_{bit}} \mathrm{dist}(c_b^{(i)}, c_b^{(j)})$$

我们的提出的多模态产品量化策略在实际应用中表现得相当好。我们还比较了不同策略的时间复杂度，例如级联的ANN（HNSW）、局部敏感哈希（LSH）以及我们提出的多模态产品量化（Multi-Modal Product Quantization）近似方法[+]。我们提出的多模态PQ方法的时间复杂度为$O(L \times M^2 \times N_{bit})$

在每次注意力计算中，有$M^2 N_{bit}$个$O(1)$距离查找操作，最终得分是这些距离的累加，远小于对多个向量进行内积计算的时间复杂度$O(L \times M \times d)$。至于两种级联的ANN（HNSW）方法，用于检索查询项对的两个过滤器，第一阶段从L个序列中检索出$M^2 K$个跨模态候选，复杂度为$L \rightarrow M^2 K$，第二阶段从第一阶段中检索出最终的前K个项，复杂度为$M^2 K \rightarrow K$。级联ANN方法的时间复杂度为$O(M^2 \log(L) d + M^2 K d)$，这比我们的产品量化策略更快，但在多个实验中，报告图显示其召回性能可能不如最佳。

因此，我们的多模态产品量化策略在时间和计算效率上都具有优势，尤其是在与级联ANN方法比较时，尽管在某些实验条件下，级联ANN方法的召回性能可能稍逊于我们的策略。

Experiment

数据集 我们对提出的SEMINAR模型在三个数据集上进行评估：两个公开数据集包括亚马逊评论数据集（电影和电视子集和由阿里支付提供的短视频数据集，以及一个工业级数据集------亚马逊的KuaiSAR搜索和推荐数据集。在三个数据集上，用户序列的平均长度分别为2000，1000，100。

Dataset	User	Item	Query	U-I
Amazon Movies & TV	297 K	181 K	-	3,293 K
Alipay Short Video	35,065 K	1,132 K	51 K	62,948 K
KuaiSAR	25,877	6,890,707	453,667	19,664,885

Experimental Results

我们在多领域数据集上的性能比较，包括KuaiSAR数据集上的$\mathbf{NDCG@K}$，亚马逊评论数据集的电影和电视子集上的$\mathbf{NDCG@K}$，以及在蚂蚁金服短视频推荐数据集上的$\mathbf{AUC}$性能，这些结果都在表中呈现。星号表示在每个任务中实现的最佳性能。

我们可以看到，SEMINAR模型在KuaiSAR数据集上的表现最佳，相较于SIM模型，在$\mathbf{NDCG@K} = 5, 10, 50$的性能分别提高了0.0292，0.0308，0.0164，在$\mathbf{NDCG@K} = 5, 10, 50$的性能分别提升了0.0088，0.0082，0.0059在亚马逊数据集上。此外，SEMINAR模型在蚂蚁金服短视频推荐数据集上的$\mathbf{AUC}$性能最佳，相较于多个强大的SOTA基准模型，提升了0.0264。

知乎

QIN	0.2535	0.2672	0.3312	0.3650	0.4038	0.4630	0.7239
ETA	0.2642	0.2756	0.3313	0.3626	0.4008	0.4607	0.7262
TWIN	0.2558	0.2709	0.3294	0.3627	0.4017	0.4605	0.7376
SEMINAR	*0.2816	*0.2969	*0.3457	*0.3661	*0.4041	*0.4636	*0.7523
Absolute Impr.	+0.0292	+0.0308	+0.0164	+0.0088	+0.0082	+0.0059	+0.0264

原文《SEMINAR: Search Enhanced Multi-modal Interest Network and Approximate Retrieval for Lifelong Sequential Recommendation》

发布于 2024-08-08 14:03 · IP 属地北京

推荐系统 工业级推荐系统 序列建模

理性发言，友善互动

发布

还没有评论，发表第一个评论吧

推荐阅读

阿里自主创新的下一代匹配&推荐技术：任意深度学习+树状...

阿里云开发者

阿里巴巴大数据之路-日志采集

01、阿里巴巴的日志采集体系方案包括两大体系：Aplus.JS是Web端（基于浏览器）日志采集技术方案；UserTrack是APP端（无线客户端）日志采集技术方案。 02、浏览器的页面日志采集（Aplus.JS...

居士慧福

阿里巴巴国际站数据分析之有点击无访客

在进行阿里国际站后台数据分析的时候，经常能发现存在点击数远远大于访客数或是有点击数，0访客数的现象,如图产品A：数据点击41，访客30的情况,点击数据比访客数据多了11,换而言之少了11个...

为主的祥助

每天超50亿推广流量、3 展现，阿里妈妈的推荐技

AI科技大... 发表于AI