

2024腾讯：ID驱动的预训练大模型，指引推荐系统发展新方向



SmartMindAI

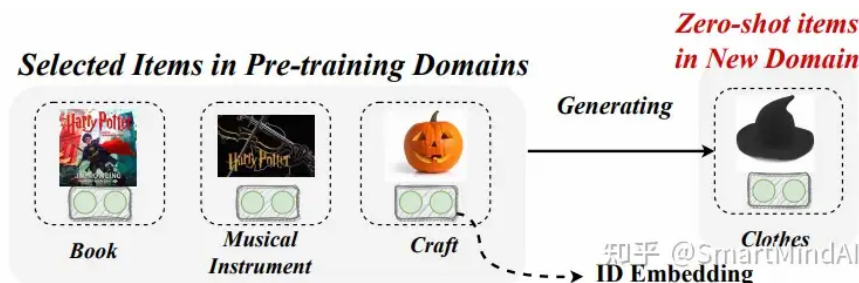
专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我

已关注

36 人赞同了该文章

Introduction

传统的顺序推荐（SR）系统通过分析用户历史行为序列，预测用户未来喜好，常用于网站推荐。它主要利用ID嵌入，将商品独特ID转化为密集的**嵌入向量**来表示商品。



然而，当面对新领域迁移时，这种方法存在挑战。本研究提出ID中心推荐预训练范式（IDP）来解决这一问题，通过在预训练阶段整合ID和行为信息，利用ID嵌入的语义，用CDIM在源领域寻找相似商品，以生成适应新领域的项目嵌入。实验证明，IDP模型在冷启动和暖启动场景下表现优越。

Methodology

Problem Statement

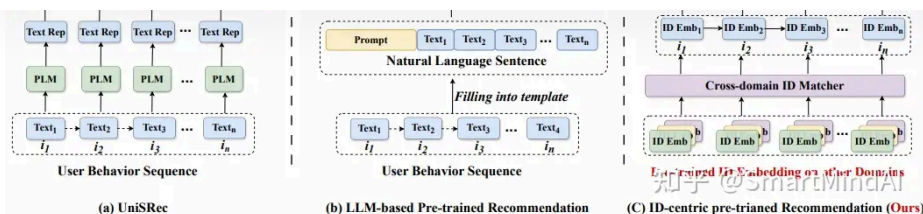
目标: $\arg \max_{v \in V} P(v|s_u, S) = \arg \max_v P(v|u, S)$ （根据 s_u 预测这表示找到与用户当前行为序列最相关，即最有可能继续交互的商品。

$$v_i^* = \arg \max_{v_i \in V} P(v_{|s_u|+1} = v_i | s_u)$$

Overview

本文中，我们提出IDP，一个基于ID的预训练推荐模型。

知乎



与之对比，IDP不通过文本信息替代ID，而是直接利用ID嵌入中蕴含的行为信息。我们强调ID在推荐中的重要性，IDP提供了一种构建预训练模型⁺的新思路。我们通过实验展示了IDP相对于其他方法的独特之处，并通过图表展示了它们的差异。

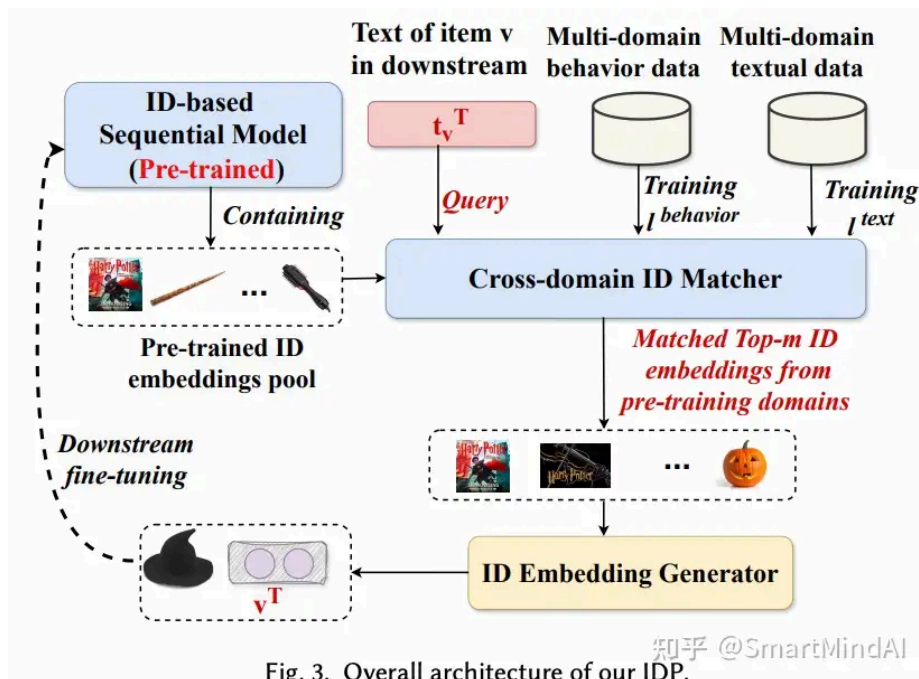


Fig. 3. Overall architecture of our IDP.

在IDP模型架构图中，有三个核心组件：

1. ID为基础的序列模型预训练：我们对ID驱动序列模型进行多领域预训练，这是我们的起点。
2. 跨领域ID匹配器（CDIM）：此模块通过联合行为信息，利用文本，实现对预训练物品嵌入与下游项目间关系的建模，是链接两者的关键桥梁。
3. 下游微调：这个阶段针对具体推荐任务，对IDP模型进行调整，以适应和提升下游领域的推荐性能。

Pre-trained ID-based Sequential Model

在这个阶段，我们利用IDP框架，基于ID的序列模型SASRec进行预训练。SASRec作为基础模型，它的功能是将用户行为序列 s_u 转化为一系列的特征向量 $h_1, h_2, \dots, h_{|s_u|}$ 。通过学习模型参数 θ 。这个过程不局限于SASRec，模型的灵活性使得它可以适用于其他类似序列模型。预训练结果 $[h_1, h_2, \dots, h_{|s_u|}]$ 随后被用作后续步骤，如跨领域ID匹配和下游微调的输入。

$$e_u = f_{seq}(s_u | \Theta, \mathbf{E}, \mathbf{P}), \text{ where } f_{seq} \text{ is SASRec,}$$

我们初始化模型时 $h_{u,i}^0$ 代表用户 u 行为项 v_i^u 的ID嵌入。沿袭传统推荐，我们用BPR（二元交叉熵⁺损失）来优化IDP预训练。具体地，损失函数⁺为：

$$\mathcal{L}_{BPR} = \sum_{(u,i,j) \in \mathcal{D}} [m_{ui}(\log(1 + \exp(-y_{ui}(h_{u,i} - h_{u,j}))) + (1 - m_{ui})(\log(1 + \exp(y_{ui}(h_{u,i} - h_{u,j}))))]$$

其中 \mathcal{D} 是训练数据 m_{ui} 是真实反馈（1表示偏好项，-1表示不偏好项） y_{ui} 是偏置（通常为1或0） $h_{u,i}$ 和 $h_{u,j}$ 是用户行为项的ID嵌入。通过最小化这个BPR损失，IDP模型学会依据ID信息生成更符合用户兴趣的推荐。

我们在多领域数据集上对IDP模型进行初始训练，直接结合所有领域的交互数据，不进行用户和项目对齐处理，保持原始的ID信息。

Cross-domain ID Matcher

(1) 文本相似性：我们考察了IDP在理解文本信息方面的表现，比如用户评论、产品描述等，通过比较ID下的内容关联度来度量推荐的相关性。

(2) 行为相似性：我们分析了IDP在捕捉用户行为习惯上的效能，如用户购买历史、浏览轨迹等，通过识别用户行为模式的相似性来优化推荐。

Textual similarity⁺

实际上，评估文本内容的相似性就像比较两段话，我们运用SimCSE来处理这一任务。尽管过去常将BERT用于文本编码⁺，但BERT并非专为句子匹配设计。因此，我们选择了SimCSE，它通过对抗性学习优化BERT，生成高质量的句子嵌入，显著提升了在句子匹配任务上的表现。具体来说，对于项目 v_i 及其对应的文本信息 c_{v_i} ，我们首先添加指示符 $[CLS]$ ，然后将 c_{v_i} 与 $[CLS]$ 拼接，输入SimCSE进行处理得到嵌入：

$$\text{Emb}_{v_i} = \text{SimCSE}([CLS] \oplus c_{v_i})$$

SimCSE通过这种方式学习到句子的向量表示，确保在衡量文本相似性时具备优势，同时弥补了BERT原始设计在该任务上的不足，增强了CDIM的性能。

$$t_{v_i} = \text{SimCSE}(\{[CLS], w_1, w_2, \dots, w_{|c_{v_i}|}\}).$$

在SimCSE中 t_{v_i} 是 $[CLS]$ 令牌最终表示。然而，由于推荐语料库⁺可能与SimCSE训练库存在差异，我们还在推荐数据上微调SimCSE。为此，我们使用对比学习，输入相同文本两次，得到编码器生成的 t_{v_i} 和 t'_{v_i} 作为正样本⁺。对比学习的目标是通过优化模型参数 θ ，使得原向量距离尽可能大，同时保持相似度大于某个阈值 m 。这有助于SimCSE更好地适应推荐环境，从而提升IDP的整体表现。

$$l_{v_i}^{\text{text}} = \frac{e^{\text{sim}(t_{v_i}, t'_{v_i})/\tau}}{e^{\text{sim}(t_{v_i}, t'_{v_i})/\tau} + \sum_k^{N-1} e^{\text{sim}(t_{v_i}, t_{v_k})/\tau}}.$$

$$\text{Contrastive Learning: } \min_{\theta} \left(\|t_i - t'_i\|^2 - \frac{\text{sim}(t_i, t'_i)}{\tau} + m \right)$$

其中 $\text{sim}(\cdot)$ 衡量两个向量的相似度 τ 是一个温度参数 m 是一个正数。目标是减小原始向量的欧氏距离，同时保持它们的相似度大于 m 。这种方法增强了SimCSE在处理推荐数据时的对抗性学习，不仅捕获文本的通用相似性，还能聚焦于行为相关特征，从而提升IDP预训练模型的精确度。

Behavior-involved tuning

在推荐系统⁺中，我们重视行为信息的相似性，因此在SimCSE的学习过程中融入了协同信号。我们首先对IDP模型在多领域预训练数据上进行联合预训练，使用预训练的序列模型。对于每个项目 v_i ，找到与其最相似的项 $V_s^{v_i}$ ，然后通过预训练嵌入的相似性作为相似性指标。对比学习是通过调整SimCSE来强化这一过程。具体而言，我们构建正负样本对。正样本是 v_i 与预训相似项 $V_s^{v_i}$ 的嵌入，负样本是 v_i 与其他不在 $V_s^{v_i}$ 内的项目。对比学习的目标是：

$$\text{对比学习: } \min_{\theta} \sum_{j=1}^k \left(\|\text{Emb}_i - \text{Emb}_{s,j}^{v_i}\|^2 - \frac{\text{sim}(\text{Emb}_i, \text{Emb}_{s,j}^{v_i})}{\tau} + m \right)$$

其中 Emb_i 是 v_i 的嵌入 $\text{Emb}_{s,j}^{v_i}$ 是其最相似项的嵌入 τ 是温度参数 m 是正数。这个策略旨在通过对比训练，让SimCSE能更准确地学习和理解行为模式，从而优化IDP模型的推荐效果。

$$v_j \in V_s, \quad \angle k$$

我们运用多任务学习来优化SimCSE，确保ID嵌入既能体现语义关联（连接不同领域项目），又能理解行为特性（适应新领域推荐）。损失函数设计如下

$$L = \sum_{v_i \in V} l_{v_i}^{behavior} + l_{v_i}^{text}.$$

Generate ID Embedding in New Domains

在这个部分，我们利用预训练⁺的项目ID嵌入来生成特定下游场景下的项目ID嵌入。通过CDIM，我们利用文本信息来量化项目间的相似性，因为相似项目往往在推荐中有相似的行为模式，这些模式储存在ID中。具体操作上，对于下游领域的项目，我们通过其文本信息 C_{item} 作为桥梁，连接到预训练的项目ID $ID_{pretrained}$ ，通过函数 f 来融合两者信息：

$$ID_{downstream} = f(ID_{pretrained}, C_{item})$$

这里 f 是一个映射，它通过比较和整合预训练ID和文本信息，生成定制化的ID嵌入，既保留了语义关联，又考虑了行为特性。这种方法使得IDP在下游推荐任务中的应用效果得以提升。

$$V_{v_i}^T = TOP([sim(t_{v_i}^T, t_1^S), sim(t_{v_i}^T, t_2^S), \dots, sim(t_{v_i}^T, t_{|V^S|}^S)], m),$$

在这个阶段，我们通过 $TOP(\cdot, \cdot)$ 函数找到 $V_{v_i}^T$ ，即CDIM判断下 m 个最相似项目。由于计算效率高，我们在“Sec.”部分讨论了模型的复杂性。接着，我们利用这 m 个ID嵌入，为目标域项目 v_i^T 生成个性化的ID嵌入：

$$ID_{target} = \sum_{j=1}^m \frac{\alpha_{ij}}{\sum_{k=1}^m \alpha_{ik}} \times ID_{top_j}$$

其中 α_{ij} 是归一化相似性 ID_{top_j} 是对应项目 j 的ID。通过加权平均，我们确保IDP模型在目标领域推荐时能更精准地聚焦于行为相似和相关性。

$$score_{v_i, v_j} = sim(t_{v_i}^T, t_{v_j}^S) / \sum_k^m sim(t_{v_i}^T, t_{v_k}^S),$$

$$e_{v_i}^T = \sum_k^m score_{i,k} * e_{v_k}^S,$$

在这个场景中 e_k^S 代表预训练领域项目 v_k^S 的ID嵌入 e_i^T 是我们为特定目标领域项目 v_i^T 生成的个性化ID嵌入。我们通过寻找与 v_i^T 最相似的 m 个预训练领域项目，即 $V_{v_i}^T = \{e_1^S, e_2^S, \dots, e_m^S\}$

并用它们的相似性来分配权重。权重 α_{ij} 通过计算每个点与 v_i^T 的余弦相似性⁺除以所有相似点的总和来确定，确保了注意力的公平分配： $\alpha_{ij} = \frac{sim(e_i^T, e_j^S)}{\sum_{k=1}^m sim(e_i^T, e_k^S)}$

最后，通过加权求和⁺，我们形成目标领域的ID嵌入： $ID_{target} = \sum_{j=1}^m \alpha_{ij} \times e_j^S$

这个方法确保了IDP模型在目标领域推荐时能更好地融合预训练知识和领域行为特性，从而提升其在目标领域任务中的性能。

Applied to downstream domain

在下游领域，我们采取直接应用策略，直接将预训练模型及相应的ID嵌入运用于目标环境。不需要对模型进行任何形式的调整，只需利用生成的ID嵌入作为输入，进行推荐或相关任务的预测。这一操作展示了模型在新领域的即时适用性。

$$e_u^T = f_{seq}(s_u^T | \Theta, E^T, P),$$

其中 f_{seq} 代表预训练的序列模型，它基于参数集 Θ ，包含了嵌入层和序列编码器的所有组件。当转移到下游领域时，我们利用不变的 Θ ，不需对模型进行针对新领域的微调。这样做是为了利用预

- 在处理下游数据时，我们不仅可能对整个预训练的序列编码器进行微调（

$$\Theta_{finetuned} = \arg \min_{\Theta} \mathcal{L}_{downstream}(f_{seq}(x; \Theta), y)$$

），还可以针对生成的ID嵌入进行优化。这里 $\Theta_{finetuned}$ 是经过调整以适应新任务的参数集合。 $\mathcal{L}_{downstream}$ 是针对特定下游任务的损失函数。 $f_{seq}(x; \Theta)$ 表示使用预训练模型和 Θ 处理输入序列的输出 y 是实际期望的结果。通过这种方式，我们既能利用ID嵌入的已有优点，又能在新数据上对序列模型进行局部优化，从而提升模型在目标领域的表现。

$$\begin{aligned} \mathbf{e}_u^T &= f_{seq}(s_u^T | \Theta^*, \mathbf{E}^{*T}, \mathbf{P}^*), \\ \Theta^*, \mathbf{E}^{*T}, \mathbf{P}^* &= \arg \max_{\Theta, \mathbf{E}^T, \mathbf{P}} P(v_{|s_u|+1} | s_u, \Theta), u \in U^T, v \in V^T, \end{aligned}$$

其中 Θ^* 代表了经过充分优化以适应下游领域的最优参数集。通过迭代优化 $\mathcal{L}_{downstream}$ ，我们旨在找到一个最优解，使模型在处理目标领域数据时能达到最佳性能。这个过程结合了预训练ID嵌入的通用知识与针对目标领域的特定训练，确保了模型既保持了泛化能力，又能显著提升在实际应用中的表现。因此 Θ^* 集成了两者的优势，实现了动态适应和性能提升。

- 当然，我们还可以考虑对预训练的编码器进行替换，比如构建一个专为目标领域定制的新编码器（不一定沿用原始预训练模型的架构）。在下游任务上，我们会对新生成的ID嵌入进行单独的微调。这种策略能够更精确地适应特定领域，因为编码器设计可能更加贴合目标数据的特性。然而，值得注意的是，这会增加**计算成本**和训练时间，因为我们需要额外的训练过程。

$$\begin{aligned} \mathbf{e}_u^T &= f_{seq}^{new}(s_u^T | \vartheta^*, \mathbf{E}^{*T}, \mathbf{P}^*), \\ \vartheta^*, \mathbf{E}^{*T}, \mathbf{P}^* &= \arg \max_{\vartheta, \mathbf{E}^T, \mathbf{P}} P(v_{|s_u|+1} | s_u, \vartheta), u \in U^T, v \in V^T, \end{aligned}$$

在处理下游数据时，我们有时会选择重构一个专属的编码器

$$\vartheta^* = \arg \min_{\vartheta} \mathcal{L}_{downstream}(f_{new_seq}(x; \vartheta, \mathbf{ID}_{generated}), y)$$

，而不是基于预训练模型。这里的 f_{new_seq} 代表基于新编码器的序列建模过程。 $\mathbf{ID}_{generated}$ 是预先微调过的ID嵌入。尽管如此，我们仍然使用预训练的ID嵌入作为初始化，以保留部分通用知识。 $\mathcal{L}_{downstream}$ 是针对特定任务的损失 y 是实际的输出。这样做旨在优化模型对目标领域数据的适应性，提升泛化能力和预测精度。然而，这通常伴随着更高的**计算复杂性**和训练时间，因为需要重新训练新的编码器。

Base Model

在本文中，我们以SASRec（采用Transformer结构的典型序列模型）为基础进行探索。SASRec以其出色的性能通过堆叠**Transformer层**来解析用户行为序列。它由嵌入层、多头注意力模块和位置-wise前馈网络三个子模块构成。

关键点在于，我们的IDP方法设计具有高度灵活性，不拘泥于特定的序列模型类型。这意味着只要序列模型拥有嵌入层和相似架构，IDP就能轻易适用。这种设计赋予了模型广泛的通用性和适应性，使其能够轻易应对各种不同的序列预测任务。

Embedding Layer

在嵌入层中，高维一维的one-hot编码项ID通过查找项集矩阵 $\mathbf{E} \in \mathbb{R}^{V \times d}$

（其中 V 代表项的总数 d 是嵌入的低维维度）转换成低维连续表示。同时，位置嵌入表 $\mathbf{P} \in \mathbb{R}^{T \times d}$

它对应序列中的每个位置，将位置信息映射到同样维度。对于用户行为序列

$$s_u = \{v_1^u, v_2^u, \dots, v_{|s_u|}^u\}$$

我们计算其输入表示：

这里 $\text{Emb}(\cdot)$ 代表每个项的嵌入 $\text{PosEmb}(s_u)$ 是基于序列中每个位置累加位置嵌入，即位置嵌入按顺序加到对应的项嵌入上。

$$\mathbf{h}_u^0 = \{\mathbf{h}_{u,1}^0, \{\mathbf{h}_{u,2}^0, \dots, \mathbf{h}_{u,|s_u|}^0\},$$

$$\mathbf{h}_{u,i}^0 = \mathbf{e}_{v_i} + \mathbf{p}_i,$$

$$\text{Input Representation}_u = \text{InputEncoding}(s_u) = \mathbf{h}_{u,1}^0 + \mathbf{p}_1 + \mathbf{e}_2^u + \dots + \mathbf{p}_{|s_u|} + \mathbf{e}_{|s_u|}^u$$

这里 InputEncoding 是将每个位置的项嵌入和位置嵌入相加的过程，形成了序列中顺序信息与项特征的整合，用于后续的序列分析或预测任务。

Self-attention Block

在嵌入层之后，我们加入了自注意力块（Self-Attention Block），模仿Transformer来捕捉序列中项目间的动态关系和用户偏好。我们使用多头自注意力机制⁺，这样可以从多个独立的子空间中分别提取信息，避免了全局视角下的单一解释。具体操作是，首先对输入进行 k 维子空间的线性变换，然后对每个子空间进行自注意力操作。随后，通过线性组合不同头部的输出，将这些子空间中的信息整合起来。这个过程可以用数学公式表示为：

$$\text{Attention}_{ij}^{(q,k,v)} = \text{softmax}\left(\frac{Q_i^T Q_j}{\sqrt{d_k}}\right) Q_k^T P_j$$

其中， $Q_k, K_v, V \in \mathbb{R}^{n \times d_k}$

是分别对应查询（query）、键（key）和值（value）的子空间投影； $-n$ 是序列长度； $-d_k$ 是子空间维度； $-P_j$ 是位置嵌入后的项嵌入 j 表示当前位置； softmax 是对角线上的softmax函数，用来归一化注意力权重。

$$MH(\mathbf{H}^l) = \text{concat}(\text{head}_1; \text{head}_2; \dots; \text{head}_k) W^O$$

$$\text{head}_i = \text{Attention}(\mathbf{H}^l W_i^Q, \mathbf{H}^l W_i^K, \mathbf{H}^l W_i^V)$$

在模型中，使用了自注意力块处理输入，其中通过

$$W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d \times d \times \frac{d}{h}} \text{ 和 } W^O \in \mathbb{R}^{d \times d}$$

这样的矩阵，将原始 d 维嵌入分解到子空间维度 $\frac{d}{h}$ 。这里的 h 是子空间数量。点积注意力（Scaled Dot-Product Attention）是计算步骤，其中查询向量 $Q_i = W_i^Q \cdot E_i$ ，键向量 $K_j = W_i^K \cdot E_j$ ，值向量 $V_j = E_j$ 和 E_j 是输入项的嵌入。点积后，应用softmax函数进行归一化，生成注意力得分。缩放因子 $\sqrt{d/h}$ 防止了除以零问题。每个位置的注意力得分通过加权求和得到输出，作为该位置的最终表示。

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

在这个场景中 K, Q, V 分别代表处理过的嵌入作为查询、键和值。 K, Q, V 的生成源于对原始嵌入的变换。 $\sqrt{d/h}$ 是一个常数，其中 d 是输入维度 h 是头的数量，用于优化计算效率并防止因大数值问题导致的溢出。

这个因子的存在是为了应对点积可能导致的数值过大问题。即使输入特征中有重复或高度相关的元素，通过将点积除以 $\sqrt{d/h}$ ，可以保持结果在合理范围内，确保了计算的稳定性和准确性。

Point-Wise Feed-Forward Network

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

其中 x 是来自自注意力层的输出 W_1 和 W_2 是权重矩阵 b_1 和 b_2 是偏置项。ReLU激活函数⁺确保只有正值传递到下一层，避免负数影响后续计算，保持了非线性特性。PPN增强了模型对复杂非线性关系的理解，从而提升模型的表达能力和预测表现。

在模型设计中，PPN使用 W_1, b_1, W_2, b_2 这组可学习参数，通过位置-wise前馈网络处理输入，以增强非线性交互。这里，常见的激活函数是GELU，它能提供非线性的增益，帮助模型理解更复杂的内在关系。PPN的计算流程是先通过ReLU激活 $xW_1 + b_1$ ，接着与 W_2 和 b_2 相乘，再加入ReLU的门控，确保正向信息传递⁺，这样它能拓展模型的表达能力，捕捉到非线性特征，显著提升预测效果。

Stack more Blocks

在实际应用中，增加模型深度有利于学习复杂模式，但过深可能导致训练难度增大。为此，我们采用了残差连接（Residual Connections）、dropout和层归一化（Layer Normalization）等方法。残差连接允许信息轻松绕过深层，解决了梯度消失问题⁺，促进了信息的高效流动。dropout随机舍弃部分神经元，有助于防止过拟合，保持模型的泛化能力。层归一化则对每层输出进行标准化，稳定了内部参数，加速了训练过程。这样，我们的模型能有效处理深层次学习任务，同时保持优秀的训练表现和泛化能力。

$$LayerNorm(x + Dropout(sbulayer(x)))$$

在模型中，我们采用 SequenceModeling 进行序列数据处理，首先对输入 x 进行 InputEncoding，捕捉序列的原始特征。然后，叠加一个自注意力子层(sublayer)以理解局部依赖关系。接下来，位置-wise 前馈网络(FFN)对自注意力层的输出进行非线性扩展，提升模型的深度学习能力。最后，通过结合这些步骤，模型能深入解析序列信息，以适应复杂模式。通过残差连接、dropout和层归一化，我们保证了训练的稳定性 and 模型的泛化能力。

$$\begin{aligned} H^0 &= E + P \\ H^l &= Trm(H^{l-1}) \\ Trm(H^{l-1}) &= LayerNorm(A^{l-1} + Dropout(PPFN(A^{k-1}))) \\ A^{l-1} &= LayerNorm(H^{l-1}, Dropout(MH(H^{l-1}))) \end{aligned}$$

在处理用户行为序列时，我们利用多层自注意力结构，随着时间步 t 生成一个动态行为表示。具体地，我们取行为序列末尾的时间步输出，即 $e_u = h_{u, |s_u|}^L$

其中 $|s_u|$ 是序列长度， L 是层数。这个 e_u 综合了用户在序列中的行为模式和潜在偏好，通过观察整个序列的交互过程来学习得到，因此包含了全面且动态的行为特征。

Dataset

下游数据集：针对每个预训练类别，我们收集了九个亚马逊⁺评论相关的具体下游数据集进行测试，涵盖了诸如商品评价、图书、音乐、电子书、电影和电视节目等多个子领域。这样做的目的是全面评估IDP模型在不同细分市场中的推荐性能，确保其泛化能力和针对性。通过比较预训练和下游数据集的结果，我们可以更准确地衡量IDP在实际应用中的表现和优势。

Pre training	1,125,233	729,305	13,920,393	19.09
Home	599,247	315,932	6,284,657	19.89
CD	82,891	109,840	1,266,516	11.53
Grocery	102,274	84,303	1,105,163	13.11
Kindle	118,061	140,592	2,188,322	15.57
Movie	222,760	78,638	3,075,735	39.11
Downstream				
Arts	1,576,189	302,370	2,726,742	9.02
Prime	247,640	10,812	447,359	41.38
Instruments	903,060	112,132	1,469,965	13.11
Office	3,402,597	306,612	5,382,474	17.55

Universality of IDP (RQ2)

1. 预训练模型泛化测试：我们在多个ID预训练模型如SASRec基础上应用IDP，结果显示其显著超越了纯ID模型，验证了IDP能有效地利用和增强ID预训练的普遍适用性。
2. 多领域序列模型验证：IDP不仅能优化推荐，还在其他序列任务中展现出显著提升，证明了其对不同应用场景的广泛适应性和通用性。
3. 跨平台验证：IDP的普适性不受平台限制，无论在哪种环境下，都能保持其优越性能，验证了其跨平台的通用性。

The universality of IDP on the pre-training base model.

我们通过实验验证了IDP的普适性，具体表现为：

Dataset	Model	HR@5	NDCG@5
Instruments	BERT4Rec	0.5286	0.4017
	IDP ^{BERT4Rec}	0.5563	0.4438
Arts	BERT4Rec	0.5091	0.3837
	IDP ^{BERT4Rec}	0.5098	0.3968
Office	BERT4Rec	0.6089	0.4681
	IDP ^{BERT4Rec}	0.6029	0.4687
Prime	BERT4Rec	0.2747	0.1861
	IDP ^{BERT4Rec}	0.3466	0.2422

1. 模型替换验证：即使将IDP与BERT4Rec等不同ID型序列模型结合，IDP在所有数据集上仍保持优秀表现，证明其不受特定模型限制，能够从预训练中提取和利用知识。

综上，IDP的通用性体现在其对多种基模型的适用性和对小数据集的有效改进，进一步证实了其在推荐领域的广泛应用。



The universality of IDP on the downstream sequential model.

在实际推荐系统的多样实践中，根据业务需求和架构差异，团队常会采用不同序列模型。传统方法受限于单一体系，无法充分利用预训练推荐模型的多功能性。因此，我们提出ID中心预训练模型，以提高灵活性。

Dataset	Model	HR@5	NDCG@5
Instruments	BERT4Rec	0.5286	0.4017
	IDP ^{B&S}	0.5428	0.4299
Arts	BERT4Rec	0.5091	0.3837
	IDP ^{B&S}	0.5149	0.3999
Office	BERT4Rec	0.6089	0.4681
	IDP ^{B&S}	0.6130	0.4773
Prime	BERT4Rec	0.2747	0.1861
	IDP ^{B&S}	0.3444	0.2406

实验中，我们选取SASRec作为ID嵌入的基础，对其进行训练，然后在下游应用中，使用BERT4Rec进行微调，尽管两者在预训练阶段并不一致。结果显示，IDP在不依赖预训练序列编码器的情况下，仍能显著提升下游性能，体现了其出色的灵活性。这证实了IDP可以在不同序列模型间通用，适应各种环境，易于在实际部署中推广，增强了其广泛的应用价值。

The universality of IDP on the cross-platform scenarios

为了验证IDP的广泛应用性，我们进行了跨平台的更复杂实验。首先，我们用IDP在亚马逊数据集上预训练，然后在MovieLens-1M数据集进行微调。实验数据显示，IDP在跨越这两个显著不同的平台（一个是电商，另一个是电影评分）上，都能显著提升推荐性能。这一结果尤其引人注目，因为用户行为在两个平台间存在明显差异。这进一步确认了IDP不受平台特性的局限，具有广泛的适用性。这为构建一个能在多种推荐场景中通用的模型提供了有力支持。

GRU4Rec	0.3374	0.4794	0.5253	0.5808	0.6919	0.4945
SASRec	0.3566	0.4948	0.5362	0.5927	0.6930	0.5082
BERT4Rec	0.3258	0.4668	0.5117	0.5682	0.6765	0.4821
DACDR	0.1444	0.2278	0.2649	0.2891	0.3791	0.2675
PterRec	0.2187	0.3475	0.3961	0.4417	0.5599	0.3763
FDSA	0.3313	0.4667	0.5107	0.5644	0.6710	0.4840
Recformer	0.1332	0.2256	0.2685	0.2942	0.3987	0.2629
UniSRec _{ID+t}	0.2902	0.4328	0.4810	0.5361	0.6531	0.4513
IDP	0.3606	0.4966	0.5381	0.5935	<u>0.6942</u>	0.5105
IDP _{ID+t}	<u>0.3566</u>	0.4904	<u>0.5363</u>	0.5874	0.6987	0.5062
Impro.	1.10%	0.36%	0.35%	0.16%	0.80%	0.43%

原文《ID-centric pre-training for recommendation》

发布于 2024-05-31 13:44 · IP 属地北京

推荐系统 腾讯 大模型



理性发言，友善互动

1 条评论

默认 最新



刘东泽

上线了？怎么动态更新

06-04 · 广东

回复 喜欢

推荐阅读

IDM下载注册教程

1、下载IDM1) 打开[IDM官网](Internet Download Manager: 最快的下载加速器)，点击“试用IDM 30 天免费试用”，进行IDM安装包下载。2) 按照安装流程安装即可。3) 打开浏览器，IDM会自动添...

yimengsama

王者荣耀背后的实时大数据平台用了什么黑科技？

简介：实时方面主要是补足我们对游戏运营的体验，比如说在游戏里玩完一局或者做完一个任务后，立马就能得到相应的奖励，或者下一步的玩法指引。对用户来说，这种及时的刺激和干预，对于他...

阿里云云栖号

如何构建IDP（内部开发者平台）？

什么是IDP？在讲IDP之前，先讲下平台工程。平台工程是软件工程中比较火的一个话题。Gartner预测，到2026年，80%的软件工程组织将建立平台工程团队，作为应用程序交付的通用组件、服务以及工...

iyacntrl



基于SDF的摇杆移动(腾讯开发精粹笔记)

晨星