

2024美团技术前沿：运用LLM模型建模商品转移概率，提升序列推荐算法效果



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

46 人赞同了该文章

Introduction

在推荐系统领域，序列推荐是个热门议题，通过分析用户历史行为预测未来喜好。早期多数基于马氏链⁺建模，后来深度学习兴起，RNN、CNN和Transformer等深度神经网络⁺被采用以捕捉交互行为。然而，这些大多依赖于无明显表达的物品相似度计算，忽视了显式关系对用户决策的影响。近期，关系感知序列推荐崭露头角，通过明确考虑物品间关系提高了推荐性能，但尚未完全克服应用局限。

针对当前推荐系统⁺中的关系型推荐面临的“关系稀疏”和“实体稀疏”问题，本研究提出了一种创新的Language Knowledge-based Latent Relation Discovery (LRD)方法，用于提升序列推荐，它通过利用大型语言模型⁺(LLM)来自动发现和理解物品间非结构化的潜在关系。这种方法克服了传统关系型推荐的关系稀疏和实体稀疏问题，因为它不依赖预设规则，而是基于LLM的语义表示自动生成关系。

具体操作上，我们设计了一个自我监督学习框架，通过LLM提取物品的语义信息⁺，然后通过关系抽取模型预测潜在关系。接着，重构单个物品模型以增强关系的准确性。LRD的优点包括：

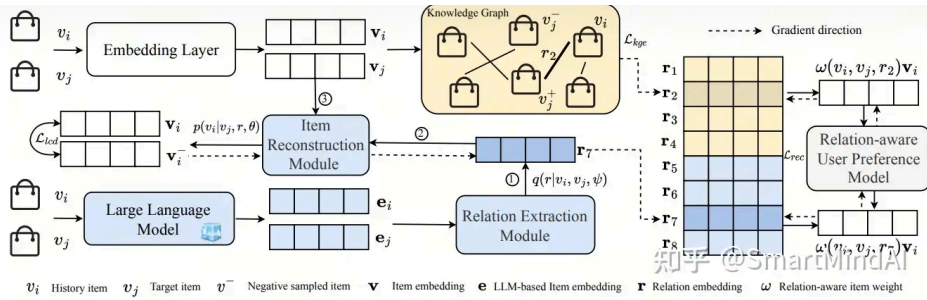
1. 模型能自主发现关系，捕捉用户复杂偏好，提升推荐质量。
2. 优化了关系感知推荐的目标，使模型趋向于发现更有价值的关系。
3. 通过分析预测关系，提高了模型的可解释性。

实验结果在多个公开数据集上证实了LRD的有效性，它显著超越了最先进的关系感知推荐模型，证明了这种方法在提升推荐性能上的优越性。

Problem Statement

我们构建了LRD模型，其中：- \mathcal{U} 代表用户群体 \mathcal{S}_u 是用户 u 的历史购买或浏览记录，按时间顺序排列。- 商品集合 \mathcal{V} 包括每一个商品实例，每个商品可能有其相关商品 v_{-i} ，关系 r 定义了它们之间的互动。- \mathcal{R} 由预定义关系 \mathcal{R}_{def} 和潜在关系 \mathcal{R}_{latent} 组成，前者存储在知识图谱 \mathcal{G} 中，包含用户行为数据中的直接关系。

目标是通过分析用户历史交互 \mathcal{S}_u ，为用户下一个可能感兴趣的物品提供排序列表。然而，传统的序列推荐仅利用 \mathcal{R}_{def} ，而我们的LRD方法则引入了 \mathcal{R}_{latent} ，这意味着它不仅考虑显式的关系，还探索潜在的、未明确定义的用户行为模式。



Framework Overview

- 1) 文本理解部分：利用LLM提取物品的语义信息，以此预测它们之间的潜在联系。
- 2) 结构生成阶段：利用预测关系和目标物品的表示，重构物品的表示，以增强模型对用户兴趣的理解。

我们将LRD融入关系感知推荐系统，通过预测的潜在关系扩展已有的项关系嵌入，以优化用户偏好模型。

Latent Relation Discovery (LRD)

Optimization Objective

我们通过预测潜在关系来达到目标，这是一种无监督学习过程，因为这类关系不在预设的有标签数据集集中。为此，我们采用了DVAE的自监督学习策略。我们假设所有潜在关系遵循一个均匀分布 $p_u(r)$ ，并建立伪似然损失函数：

$$\mathcal{L}_{\text{Pseudo}} = - \sum_{(i,j,r) \in \mathcal{D}} p_u(r) \log p_\theta(r|i, j)$$

其中 \mathcal{D} 代表数据集 θ 是模型参数 $p_\theta(r|i, j)$ 表示模型预测在物品 (i, j) 下潜在关系 r 的概率。这个损失函数⁺促使模型学习到的潜在关系分布尽可能接近真实分布 $p_u(r)$ ，从而实现对潜在关系的有效发现。

在这个具体场景中 v_i 和 v_{-i} 代表一对具体的物品 v_{-i} 是它们的配对，即除了 v_i 的所有其他物品。 $p(v_i|v_{-i}, r, \theta)$ 表示在已知其他物品和关系 r 条件下，物品 v_i 出现的概率 θ 是模型的参数集合。

我们使用自监督学习的策略，通过构建伪似然损失函数 $\mathcal{L}(\theta)$ 来估计潜在关系的分布。为了做到这一点，我们引入了一个变分后验分布⁺ $q(r|v_i, v_{-i}, \psi)$ ，其中 ψ 是附加的参数。Jensen不等式允许我们将伪似然函数⁺下界化，以便更精确地估计模型的性能。因此，优化目标可以改写为：

$$\mathcal{L}(\theta) \geq - \sum_{(i,j,r) \in \mathcal{D}} q(r|v_i, v_{-i}, \psi) \log p_\theta(r|i, j)$$

通过这个优化，模型能够学习到如何根据已有的 (v_i, v_{-i}) 对和关系 r ，推断出潜在关系 r 的概率分布⁺，进而提升对潜在关系的识别和预测能力。

$$\mathcal{L}(\theta) > \mathcal{L}(\theta, \psi) = \sum_{i=1}^2 \sum_{r \in \mathcal{R}} q(r|v_i, v_{-i}, \psi) \log p(v_i|v_{-i}, r, \theta) + \alpha H[q(r|v_i, v_{-i}, \psi)],$$

在这个情境中 $q(r|v_i, v_{-i}, \psi)$ 是用于预测特定物品对关联关系的模型，它利用 v_i 和 v_{-i} （排除 v_i 的其他物品）来推断关系，而 $p(v_i|v_{-i}, r, \theta)$ 则是个别物品表示的重构模型，它基于预测的关系重构物品的特征，以反映潜在的偏好。 ψ 和 θ 分别是关系提取和重构模型的参数集。

通过优化目标，模型试图最大化重构真实物品概率以提高预测准确性，同时通过加入熵项 H 作为正则化项，保持预测关系的多样性，防止过拟合⁺。参数 α 控制着正则化的力度，影响模型的复杂性和泛化能力。

Relation Extraction

则有限。因此，我们借鉴人类通过语言理解和描述关系的能力，利用大型语言模型（LLM）来挖掘物品间的潜在关系。LLM能捕捉丰富的世界知识⁺和语义，所以我们将每个物品的文本令牌序列输入LLM，获得其语言知识表示，如表达式所示：

$$\text{Item Representation} = LLM(w_1, w_2, w_3, \dots, w_{N_v})$$

这里 N_v 表示物品的令牌数量 w_i 是对应的文本元素，通过LLM生成的表示能够捕捉到物品之间的语义关联。然后，这些表示被用于关系提取模型，以期更好地理解并预测物品间的潜在关系。

$$\mathbf{e} = W_1(LLM([w_1, w_2, w_3, \dots, w_{N_v}])) + b_1,$$

在这个上下文里 $LLM(\cdot)$ 是利用特定LLM对隐藏状态进行池化操作，以获得物品的语义表示。LLM的输出通过投影层(W_1 和 b_1)降维到与推荐模型匹配的维度。我们称这些由LLM生成的表示为语言知识表示，它们包含了超出预定义关系集的信息。关系提取模型 $q(r|v_i, v_{-i}, \psi)$ 利用这些语言知识，预测物品对 \mathbf{e}_i 和 \mathbf{e}_{-i} 在关系集 \mathcal{R} 上的关系。模型通常采用线性分类器⁺，如公式所示：

$$\hat{r} = \text{softmax}(W_2[\mathbf{e}_i; \mathbf{e}_{-i}] + b_2)$$

其中 W_2 和 b_2 是分类器的权重和偏置，softmax函数计算每个关系类别的概率分布 \hat{r} 是预测的关系类别。这种方法允许模型根据LLM的丰富知识来推断潜在关系，弥补了仅依赖预定义关系的不足。

$$q(r|v_i, v_{-i}, \psi) = \text{SoftMax}(W_2[\mathbf{e}_i; \mathbf{e}_{-i}] + b_2),$$

在这个设置中 W_2 和 b_2 是线性分类器的参数，它们将LLM生成的融合向量 $\mathbf{e}_i \otimes \mathbf{e}_{-i}$

（通过拼接操作 \mathbf{e}_i 和 \mathbf{e}_{-i} ）映射到关系集 \mathcal{R} 的类别空间。线性分类器通过这样的组合提供每个关系类别的预测概率：

$$\hat{r} = \text{softmax}(W_2[\mathbf{e}_i; \mathbf{e}_{-i}] + b_2)$$

softmax函数确保输出概率分布在0到1之间且总和为1，便于理解和后续决策。

Relational Item Reconstruction

1. 特征提取⁺：从关系 \hat{r} 中抽取与 v_{-i} 相关的特征，比如关系相关属性或上下文信息。
2. 重建映射：建立一个根据 \hat{r} 转换特征的映射函数，生成 v_{-i} 的潜在表示 \hat{v}_{-i} 。
3. 重构：利用潜在表示，通过解码机制如神经网络生成重构的物品表示 \tilde{v}_{-i} 。
4. 评价：通过评估 \tilde{v}_{-i} 与真实 v_{-i} 的差异，如误差或相似度，来衡量模型的重建性能。

通过这种方式，项重建模型能从已知关系中推断出未观测到的细节，从而提高关系提取模型的稳健性和精度。

$$p(v_i|v_{-i}, r, \theta) = \frac{\exp(\phi(v_i, v_{-i}, r))}{\sum_{v'_i \in \mathcal{V}} \exp(\phi(v'_i, v_{-i}, r))},$$

在这个上下文中，评估函数 $\phi(v_i, v_{-i}, r, \theta)$ 是用来量化两个物品 (v_i, v_{-i}) 和关系 r 之间关联强度的指标，它可以根据不同的评分模型来定义。我们选择DistMult作为通用模型，其数学形式为：

$$\phi(v_i, v_{-i}, r, \theta) = \mathbf{v}_i^\top \mathbf{R}_r \mathbf{v}_{-i}$$

其中 \mathbf{v}_i 和 \mathbf{v}_{-i} 是物品的嵌入向量 \mathbf{R}_r 是关系矩阵 θ 是模型参数。DistMult通过矩阵乘积来模拟关系的直接作用，估算在特定关系 r 下，两个物品的相似度或相关性。这样，通过调整 θ ，我们可以根据模型学习到的关系规则来预测和理解不同情境下的物品关系。

$$\phi(v_i, v_{-i}, r) = \mathbf{v}_i^\top \text{diag}(\mathbf{r}) \mathbf{v}_{-i},$$

$$\log p(v_i|v_{-i}, r, \theta) \approx -\frac{1}{k} \sum_{n=1}^k \text{sim}(\mathbf{v}_i, \mathbf{v}_{j_n}) + \text{const}$$

其中 $\text{sim}(\cdot, \cdot)$ 衡量向量相似性 \mathbf{v}_{j_n} 是从 \mathcal{V} 中随机选取的负样本 k 是负样本数量，常数项保证了概率的标准化。这种方法降低了计算复杂度，同时保持了对关系影响的估计能力。

$$\theta^* = \arg \max_{\theta} \left(-\frac{1}{k} \sum_{n=1}^k \sigma(\mathbf{v}_i^\top \mathbf{v}_{j_n}) + \text{const} \right)$$

这里 $\mathbf{v}_i^\top \mathbf{v}_{j_n}$ 是**正样本**⁺ (\mathbf{v}_i) 与负样本 (\mathbf{v}_{j_n}) 的内积, **sigmoid函数**⁺ 确保输出值在0到1之间, 表示它们的相似程度。通过最大化这个估计, 我们优化了模型, 使其在给定关系 \mathbf{r} 下更准确地判断和重构物品, 因为这最大化了相关性概率。

$$\begin{aligned} \mathcal{L}(\theta, \psi) = & \sum_{i=1}^2 \sum_{\mathbf{r} \in \mathcal{R}} q(\mathbf{r} | \mathbf{v}_i, \mathbf{v}_{-i}, \psi) [\log \sigma(\phi(\mathbf{v}_i, \mathbf{v}_{-i}, \mathbf{r}, \theta)) \\ & + \log \sigma(-\phi(\mathbf{v}_i^-, \mathbf{v}_{-i}, \mathbf{r}, \theta))] + \alpha H[q(\mathbf{r} | \mathbf{v}_i, \mathbf{v}_{-i}, \psi)]. \end{aligned}$$

LRD-based Sequential Recommendation

Relation-aware Sequential Recommendation

对于用户 u 与其目标项 v_j , 我们利用用户过去的交互记录

$$S_u = \{v_1, v_2, \dots, v_{N_u}\}$$

来估计其对 v_j 的兴趣。考虑到每个历史项与目标项间的关系, 我们引入关系嵌入 $r_{v_i v_j}$, 这增强了推荐的洞察能力。用户对 v_j 的偏好分数定义为:

$$\text{pref}(u, v_j) = f(S_u, (r_{v_1 v_j}, r_{v_2 v_j}, \dots, r_{v_{N_u} v_j}), \text{动态交互})$$

其中 $f(\cdot)$ 是一个融合函数, 它将用户历史、每个交互关系以及动态交互信息整合起来。这样, 通过关系嵌入, 我们可以量化不同历史项对目标项的影响力, 从而更精确地揭示用户潜在的喜好。

$$\mathbf{y}_{u,j} = (\mathbf{u} + \mathbf{m}_{u,j}) \mathbf{v}_j^T + b_j,$$

$$\mathbf{m}_{u,j} = \text{AGG}([\mathbf{s}_{u,j,r_1}; \mathbf{s}_{u,j,r_2}; \dots; \mathbf{s}_{u,j,r_{|\mathcal{R}|}}]),$$

在该模型中, AGG是一个用于聚合信息的函数, 可以选择平均、最大或注意力等方法。关系集合 \mathcal{R} 包含预定义和LLM生成的关系。对于用户 u 和目标项 v_j , 当考虑关系 r 时, 用户的历史行为序列 $\mathbf{s}_{u,j,r}$ 通过计算所有与 r 相关联的用户交互 $\mathbf{s}_{u,i}^{(r)}$ 的AGG结果来获取。 $\mathbf{s}_{u,i}^{(r)}$ 是用户在 r 关联下所有交互的特征表示, AGG操作将这些交互的特征整合成用户在特定关系下的整体偏好表示。

$$\mathbf{s}_{u,j,r} = \sum_{v_i \in S_u} \omega(v_i, v_j, r) \mathbf{v}_i,$$

$$w(v_i, v_j, r) = \frac{\phi(v_i, v_j, r, \theta)}{\sum_{r' \in \mathcal{R}} \phi(v_i, v_j, r', \theta)}$$

这里 \mathcal{R} 是关系集合 $\phi(\cdot)$ 对每个关系进行加权 θ 确定其权重。归一化过程确保了在不同关系间比较时的公正性, 因为所有可能的关系对 v_i 和 v_j 的相对重要性都被考虑进来。

$$\omega(v_i, v_j, r) = \frac{\exp(\phi(v_i, v_j, r))}{\sum_{v'_i \in \mathcal{V}/S_u} \exp(\phi(v'_i, v_j, r))},$$

在这个上下文中 $\phi(v_i, v_j, r)$ 代表的是对物品 (v_i, v_j) 在关系 r 下的评分, 它与前文提到的用于物品重建模型(对应方程(...))中的评分逻辑)保持一致。这种统一性使得在后续的优化过程中, 能够无缝地整合这些评分信息, 方便进行联合优化。

Joint Learning

$$L(u, v_j; \theta) = - \sum_{\mathbf{r} \in \mathcal{R}} \sum_{(u, v_i) \in S_u \times \mathcal{V}, (u, v_j) \notin S_u} \log \sigma(\text{pref}(u, v_j) - \text{pref}(u, v_i) + \alpha r_{uv})$$

其中 \mathcal{V} 表示所有项 (u, v_i) 代表用户 u 与项 v_i 的交互 **pref** 基于用户历史和 r_{uv} 计算出用户对 v_j 的预测评分。BPR通过对比用户对未接触 v_j 与已接触过但评价较低的 v_i 的偏好差异, 促使模型优先推荐高

$$\mathcal{L}_{rec} = - \sum_{u \in \mathcal{U}} \sum_{j=2}^{N_u} \log \sigma(y_{u,j} - y_{u,j-}).$$

为结合潜在关系的发现与推荐任务，我们提出联合优化策略。在公式中，首先定义了关系发现任务的损失 $\mathcal{L}_{discovery}$ ，它依赖于历史项 $\mathbf{v}_i, \mathbf{v}_j$ 、关系 r_{ij} 和模型参数 θ 。推荐任务的损失 $\mathcal{L}_{recommendation}$ 则基于用户 \mathbf{u} 和目标序列 $\mathbf{m}_{u,j}$ 。目标函数*调整为：

$$\min_{\theta, \{\alpha_n\}} \mathcal{L}_{discovery}(\{\mathbf{v}_i, \mathbf{v}_j, r_{ij}\}; \theta) + \mathcal{L}_{recommendation}(\mathbf{u}, \mathbf{m}_{u,j}; \theta)$$

这样，模型在优化过程中既要学习从潜在关系中提取信息以提升推荐，又受用户实际行为的反馈引导，从而实现对相关性和准确性的双重提升。

$$\begin{aligned} \mathcal{L}_{lrd} = - \sum_{u \in \mathcal{U}} \sum_{j=2}^{N_u} \sum_{i=1}^{j-1} \sum_{r \in \mathcal{R}} q(r | v_i, v_j, \psi) [& \log \sigma(\phi(v_i, v_j, r, \theta) \\ & + \log \sigma(-\phi(v_i^-, v_j, r, \theta))) \\ & + \alpha H[q(r | v_i, v_j, \psi)]. \end{aligned}$$

$$\min_{\theta} \mathcal{L}_{KG}(\{\mathbf{e}_v, \mathbf{e}_r\}) = \sum_{(v_i, r, v_j) \in \mathcal{T}} \text{Loss}_{KG}(\mathbf{e}_{v_i}, \mathbf{e}_r, \mathbf{e}_{v_j}; \theta)$$

其中 \mathbf{e}_v 和 \mathbf{e}_r 是物品和关系的嵌入 Loss_{KG} 是针对知识图谱节点的损失，通过 θ 进行优化。这样，模型不仅能学习潜在关系，还能利用已知结构，从而提升推荐的精准度和对知识图谱内容的理解。

$$\mathcal{L}_{kge} = - \sum_{(v_i, v_{-i}, r) \in \mathcal{T}} \log \sigma(\phi(v_i, v_{-i}, r) - \phi(v_i^-, v_{-i}^-, r)).$$

在联合优化中，我们结合了序列推荐和知识图谱嵌入两个任务，目标函数为：

$$\min_{\theta} L = \mathcal{L}_{seq}(\theta) + \mathcal{L}_{KG}(\{\mathbf{e}_v, \mathbf{e}_r\})$$

其中 \mathcal{L}_{seq} 代表关系感知序列推荐的BPR损失，处理用户行为数据，优化用户体验 \mathcal{L}_{KG} 关注知识图谱中的嵌入，通过调整 θ 利用预定义关系，增强对知识的理解和应用。通过同时优化这两个目标，系统能够更好地融合用户行为和领域知识，提供更精准且相关的目标推荐。

$$\mathcal{L} = \mathcal{L}_{rec} + \gamma \mathcal{L}_{kge} + \lambda \mathcal{L}_{lrd},$$

在该模型中，参数 γ 和 λ 分别控制了知识图谱嵌入任务 (\mathcal{L}_{KG}) 和潜在关系发现任务 ($\mathcal{L}_{discovery}$) 的权衡。 γ 调整了对知识图谱信息在推荐决策中的作用，而 λ 则关乎从用户互动数据中发现的关系的重要性。通过调整这两个系数，模型能在保证推荐精确性的同时，灵活地融合这两种来源的信息，提升推荐的有效性和对领域知识的洞察能力。

Experimental Settings

Datasets

我们在三个来源广泛的公开数据集上进行了实证研究，以评估模型*在各类推荐环境下的关系发现能力。这些数据集包含多样化的用户行为和内容，旨在全面检验模型在处理复杂关系时的表现。我们不仅关注提升推荐的准确性，还着重于验证模型能否深入理解和解析用户需求背后的隐含关系，从而提供更为精确和贴近用户兴趣的推荐内容。

Datasets		MovieLens	Offices	Electronics
User-Item Interactions	#user	943	4,905	192,403
	#item	1,349	2,420	63,001
	#inter.	99,287	53,258	1,682,498
	density	7.805%	0.448%	0.014%
Item Relations	#relation	2	4	4
	#triplets	886K	778K	2,148M

在所有数据集的处理过程中，我们遵循既定的标准，对用户和项目进行了筛选，只保留了交互次数超过5次的观测点。这是为了确保数据的可靠性和模型的训练有效性。接下来，我们对处理后的数据集进行了统计描述，如表所示。这些统计信息反映了原始数据的概况，为后续的LRD模型训练和性能评估提供了基础。

Baselines

在我们的实验设置中，我们选取了多组不同的序列推荐模型作为参照，以全面评估LRD模型的性能。这样做的目的是通过对比，明确LRD在提升推荐质量方面的独特贡献，以及它如何通过调整参数 γ 和 λ ，在充分利用知识图谱信息和用户交互数据中的潜在关系的同时，优化推荐策略，从而量化这一贡献。

Evaluation Metrics

为了评估模型表现，我们采用了HR@K（Top-K准确率）和nDCG@K两种标准，其中K分别为5和10。采用留一交叉验证策略，我们构建数据集，每轮选取用户的一个最新交互作为测试样本，前一个交互作为验证样本，剩余部分作为训练数据。在预测时，我们依照既定步骤，将目标项与99个随机选择的非目标项进行排序。实验结果将以五次重复实验的平均值呈现，以消除偶然性影响。

Performance Comparison (RQ1)

我们对比了两种基于LRD增强的关系感知序列推荐模型，即RCF~LRD~和KDA~LRD~。整体实验结果见表中呈现。

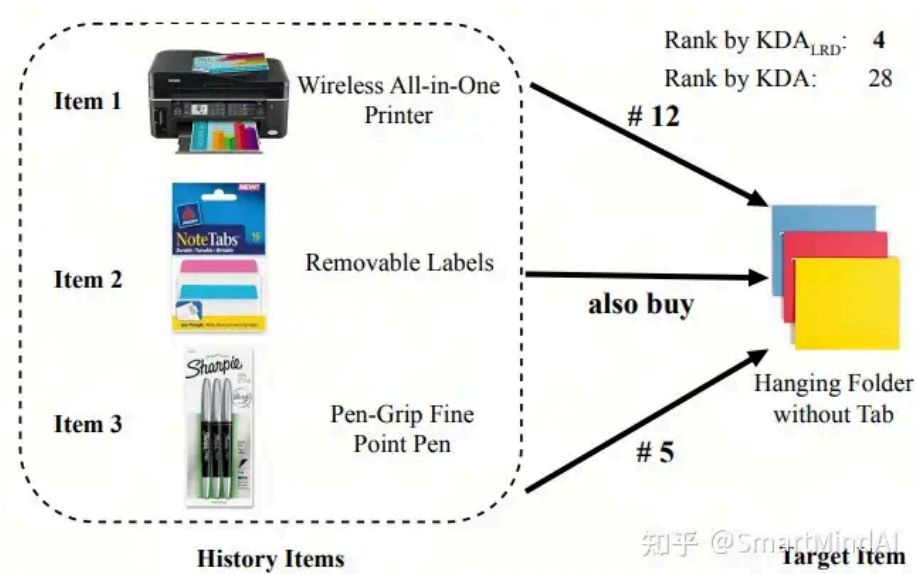
Table 3: Overall performance of different models. The best performances are denoted in bold fonts. “H@K” is short for “HR@K” and “N@K” is short for “NDCG@K”, respectively. The subscript “LRD” denotes the model is enhanced by LRD. “Improv.” means the relative improvement of the LRD-based model over the corresponding vanilla model. The superscripts † and ‡ indicate $p \leq 0.05$ and $p \leq 0.01$ for the paired t-test of the LRD-based model vs. vanilla model.

Datasets	MovieLens				Office				Electronics			
	H@5	H@10	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10	N@5	N@10
Caser	0.5217	0.6872	0.3571	0.4107	0.3095	0.4762	0.1993	0.2530	0.4620	0.5865	0.3435	0.3838
GRU4Rec	0.5101	0.6723	0.3451	0.3976	0.3295	0.4856	0.2164	0.2670	0.4699	0.5994	0.3487	0.3906
SASRec	0.5186	0.6829	0.3712	0.4242	0.4027	0.5439	0.2751	0.3210	0.4805	0.6083	0.3587	0.4000
TiSASRec	0.5313	0.6882	0.3812	0.4322	0.4014	0.5433	0.2745	0.3209	0.5114	0.6329	0.3860	0.4253
RCF	0.5101	0.6660	0.3635	0.4137	0.4145	0.5696	0.2911	0.3413	0.5790	0.7004	0.4475	0.4868
RCF _{LRD}	0.5398 [‡]	0.6882 [‡]	0.3886 [‡]	0.4365 [‡]	0.4381 [‡]	0.5761 [‡]	0.3127 [‡]	0.3573 [‡]	0.5828 [†]	0.7035 [†]	0.4510 [†]	0.4901 [†]
Improv.	+5.82%	+3.33%	+6.91%	+5.51%	+5.69%	+1.14%	+7.42%	+4.69%	+0.66%	+0.44%	+0.78%	+0.68%
KDA	0.5748	0.7381	0.4182	0.4711	0.4453	0.6145	0.3127	0.3676	0.6008	0.7194	0.4665	0.5049
KDA _{LRD}	0.6066 [‡]	0.7434 [‡]	0.4420 [‡]	0.4867 [‡]	0.4826 [‡]	0.6302 [‡]	0.3403 [‡]	0.3881 [‡]	0.6503 [‡]	0.7255 [‡]	0.4780 [‡]	0.5143 [‡]
Improv.	+5.53%	+0.72%	+5.69%	+3.31%	+8.38%	+2.55%	+8.83%	+5.58%	+1.71%	+1.40%	+2.04%	+1.86%

对于基线方法，有几点观察值得注意。首先，传统的序列推荐模型主要依赖用户交互历史中的隐含偏好来预测下一个物品，这使得它们高度依赖丰富的交互数据。在相对密集的MovieLens数据集上，传统方法与关系感知方法的性能差距并不显著。然而，在高密度的Amazon数据集上，传统方法的表现远逊于关系感知方法。其次，将物品关系纳入序列推荐中，RCF和KDA显著优于非关系感知的序列推荐模型，表明明确建模历史项与目标项之间的关系能更有效地捕捉用户的偏好。再者，TiSASRec和KDA通过结合SASRec和RCF分别处理时间信息的模块，实现了显著性能提升，这表明在用户交互序列建模中考虑时间因素能有效提升模型性能。

性。特别是在MovieLens这类具有较少预定义关系的数据集上，RCF的性能并不比传统方法有显著优势。然而，通过我们的LRD方法，我们能够有效增强关系感知模型，捕捉潜在的物品关系。此外，KDA~LRD~的性能提升进一步表明，由LRD发现的潜在关系也具有随时间演变的特性。

Case Study



Hyper-parameter Sensitivity

在探索潜在关系数量（Num_latent）和联合优化中潜在关系发现任务系数（ λ ）的影响时，我们首先关注了Num_latent的变化。实验结果显示，模型在不同Num_latent下表现出显著性能波动，这强调了适度的潜在关系数量对模型性能至关重要，过多或过少都会导致效果下降。同时，无论在哪种数据集（如MovieLens和Office），模型对Num_latent的总体适应性有相似趋势，但最优值却各不相同，这验证了不同推荐环境对关系理解的要求可能不同。

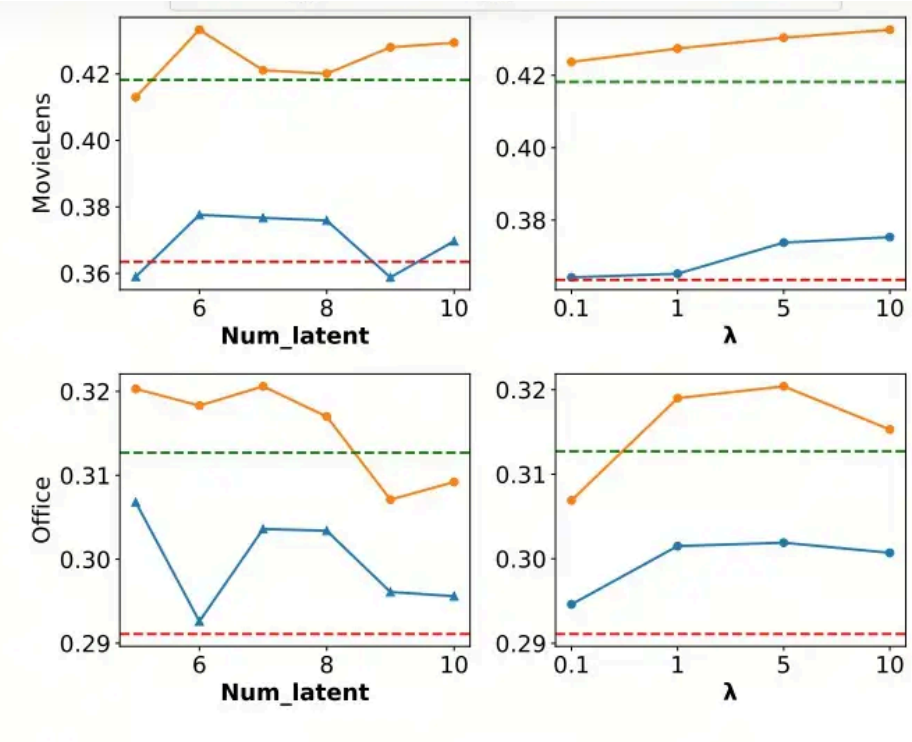


Figure 6: nDCG@5 comparison w.r.t. the number of latent relations and the coefficient of the latent relation discovery task, i.e., λ .

接下来，我们考察了 λ 的调节。当我们保持Num_latent不变，观察不同 λ 下的平均性能时，我们发现当 λ 达到最优值时，模型性能稳定提升。这表明潜在关系发现任务对于模型的提升作用是普遍且显著的，不论在何种数据集上都能提升推荐性能。综合来看，合理选择Num_latent和 λ 参数对于优化关系嵌入的模型至关重要，可以有效地改善推荐系统的效能。

原文《Sequential Recommendation with Latent Relations based on Large Language Model》

发布于 2024-05-10 10:39 · IP 属地北京

美团 LLM（大型语言模型） 序列推荐

▲ 赞同 46 ▼ ● 添加评论 ↗ 分享 ❤ 喜欢 ★ 收藏 📄 申请转载 ...



理性发言，友善互动



发布



还没有评论，发表第一个评论吧