

百度2023探索：运用LLM技术强化图模型架构



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

1 人赞同了该文章

Introduction

推荐系统在缓解信息过载方面起着关键作用。有效的推荐系统需精确理解用户偏好，包括分析历史互动模式及整合丰富属性信息，如内容属性、用户生成内容、多模态特征等。多样化的信息能提供独特的偏好描述方式，通过利用这样的属性信息，模型可以以个性化的方式进行推荐。然而，尽管基于NN的推荐系统取得了显著的进步，这些方法通常面临着数据稀缺和处理属性信息的问题。

稀疏的隐式反馈信号：数据稀疏性和冷启动问题限制了协同偏好捕捉。一些研究在协同过滤⁺ (CF) 中 (如NGCF、LightCGN) 使用强大的图神经网络⁺ (GNN)，但由于缺乏足够的监督信号而受到限制。有一些研究使用对比学习添加自监督信号 (如SGL、SimGCL)。

然而，最近的方法不同于一般的CF，更关注属性信息作为辅助建议器。例如，MMGCN和GRCN将项末尾内容引入GNNs来发现高阶内容感知关系。LATTICE利用辅助内容进行数据增强，通过建立i-i关系。最近的研究 (如MMSSL、MICRO) 通过引入多个内容增强视图之间的最大互信息最大化来解决稀疏性问题。然而，在推荐系统中解决稀疏性策略可能有限制，尤其是对于多模态内容。这是因为复杂性和缺少属性信息对CF的相关性引入了偏差。因此，确保在将属性信息融入CF时准确捕获现实的用户偏好变得至关重要，以避免次优结果。

LLMs是在大量真实世界的知识上训练的，允许它们理解用户的偏好并提供有价值的完成信息，即使对于受隐私保护的用户档案也是如此。

大语言模型的全面词汇库将嵌入在一个单一的向量空间中，桥接异构特征之间的鸿沟，并促进编码计算。这种融合防止了特征在不同的向量空间中的分散，并提供了更精确的结果。现实中将大型语言模型作为有效的推荐数据增强器存在技术挑战需要解决。

- 如何利用明确的隐式反馈信号提高大型语言模型推理用户-内容交互模式的能力？
- 如何保证使用大型语言模型增强的内容的可靠性，避免引入可能损害结果的噪声？

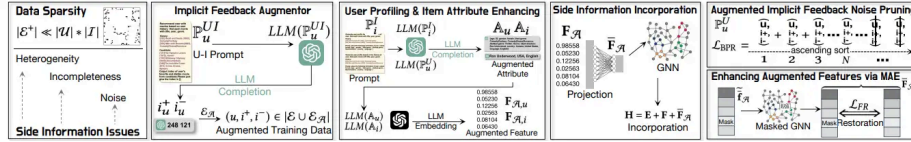
大型语言模型可以有效增强推荐系统以解决稀疏性问题和改善不完整边信息。但是，实现这一方法需要解决一些挑战。为此，我们提出了一种新的框架[LLMRec]来解决这些挑战。

模；iii) 用户特征化。

对于第一部分，我们设计了一种基于大型语言模型的贝叶斯个性化排名采样算法，用于从文本内容中挖掘用户可能喜欢或不喜欢的内容。

对于第二部分，我们在节点属性生成过程中为每个用户/项创建额外的属性，但必须注意到增强的边和节点特征都可能包含噪声。为了解决这个问题，我们提出了一个去噪数据鲁棒化机制，结合了噪音边缘删除和MAE 两种方法来确保增强数据的质量。

Preliminary



本文介绍了一种基于图嵌入的推荐系统。该系统通过使用协同过滤方法来学习用户的兴趣相关向量。然后，研究人员使用图神经网络来模拟用户和内容的复杂关系。最终，该系统被划分为两个阶段：首先，生成用户的兴趣相关向量；其次，根据用户和内容的兴趣相关向量进行推荐。

$$\mathbf{E}^* = \arg \max_{\mathbf{E}} p(\mathbf{E} | \mathcal{E}^+)$$

该系统首先从 \mathcal{E}^+ 中提取出u-i关系，然后根据这些关系进行u-i预测。为了处理数据稀疏性，作者引入了节点特征，并将其作为一个特征图的形式出现。然后，作者通过最大似然估计法⁺求解最大概率参数，其中包含u-i关系、节点特征和特征编码器。

$$\Theta^* = \arg \max_{\Theta} p(\Theta | \mathbf{F}, \mathcal{E}^+)$$

推荐编码器 f_{Θ} 会输出最终的表示 \mathbf{h} ，这个表示包含了来自 \mathbf{E} 的合作伙伴关系信号以及来自 \mathbf{F} 的边信息。 \mathbf{h} 的计算方式是将 \mathbf{f} 和 \mathcal{E}^+ 输入到 f_{Θ} 中。

虽然加入边信息使推荐系统有了显著的进步，但引入低质量的边信息可能会削弱稀疏交互 \mathcal{E}^+ 的效果。因此，研究人员开发了一种新的方法来解决这个问题，那就是基于LLM的u-i交互边增强 \mathcal{E}_A ，以及基于LLM的节点特征生成 \mathbf{F}_A 。优化目标是为了增强交互特征图。

$$\Theta^* = \arg \max_{\Theta} p(\Theta | \{\mathbf{F}, \mathbf{F}_A\}, \{\mathcal{E}^+, \mathcal{E}_A\})$$

推荐系统 f_{Θ} 将原始和增强的数据作为输入，其中包括边 $\{\mathcal{E}^+, \mathcal{E}_A\}$

和节点特征 $\{\mathbf{F}, \mathbf{F}_A\}$

然后输出高质量的表示 \mathbf{h} 来预测用户的偏好分数 $\hat{y}_{u,i}$ 。最后通过排序用户 u 与物品 i 互动的可能性来完成推荐任务。

Methodology

Q1: 如何使LLMs能够预测u-i交互边？ **Q2:** 如何使LLMs能够生成有价值的内容？ **Q3:** 如何将增强内容纳入原始图内容中？ **Q4:** 如何使模型对增强的数据保持鲁棒性⁺？

LLMs as Implicit Feedback Augmentor (Q1)

(1)这种增强方法使推荐器能够充分利用整个数据集中的内容；(2)这种增强方法能够直观地反映用户的偏好。将用户-项目交互边缘进行增强并在训练数据中包含它的过程可以用以下公式表示：

$$\hat{\mathcal{E}} = \{(u_i, v_j), \hat{y}_{u_i, v_j}\}$$

$$i_u^+, i_u^- = LLM(\mathbb{P}_u^{UI}); \quad \mathcal{E}_{BPR} = \mathcal{E} \cup \mathcal{E}_A$$

来确保选择出的用户-内容交互边具有代表性。iv) 输出格式描述：输出结果将以图的形式展示，图(a)给出了其基本结构。

我们在研究中使用LLMs采样器，以一定程度上减少原始隐式反馈中存在的噪声和非互动项问题。

(i) 噪声：我们定义为不是由用户真实意愿引起的交互，例如偶然点击或受流行度偏好的影响的实例。(ii) 非互动项：我们定义为可能不反映用户真实的喜好，但通常被认为是负样本。

通过将LLMs用作隐式反馈增强器，模型可以通过LLMs强大的推理能力和知识支持获取更有意义、更具信息量的样本。具体来说，我们将模型与LLMs结合在一起，以便在收集到的数据中找到更多有意义的信息。

LLM-based Side Information Augmentation

User Profiling & Item Attribute Enhancing (Q2)

- 使用从数据集交互和属性信息中提取的提示，我们将LLM用于细化用户和物品信息。这使得LLM生成了原始数据集中没有的部分用户和物品属性。具体的例子可以在图(b)和(c)中看到。
- 我们使用LLM作为编码器来增强用户的物品信息，并将其以特征形式编码，作为推荐系统的输入。这样可以利用LLM提供高效的语义理解，从而更好地描述用户的交互偏好并去偏差化物品属性。
- 正式的来说，我们的属性信息增强是这样的：

$$\begin{cases} \text{user: } \mathbf{A}_u = LLM(\mathbb{P}_u^U) & \longrightarrow & \mathbf{f}_{\mathcal{A},u} = LLM(\mathbf{A}_u) \\ \text{item: } \mathbf{A}_i = LLM(\mathbb{P}_i^I) & \longrightarrow & \mathbf{f}_{\mathcal{A},i} = LLM(\mathbf{A}_i) \end{cases}$$

其中 $\mathbf{f}_{\mathcal{A},u}$ 和 $\mathbf{f}_{\mathcal{A},i}$

是以LLM的隐藏维度 d_{LLM} 维度表示的LLM加强的用户/物品特性。文本提示 \mathbb{P}_u^U 和 \mathbb{P}_i^I 分别用于改进用户的属性和物品的属性。 \mathbf{A}_u 和 \mathbf{A}_i 表示生成的文本属性，这些属性将使用LLM的嵌入功能编码为特性 $\mathbf{F}_{\mathcal{A},u}$ 和 $\mathbf{F}_{\mathcal{A},i}$

Side Information Incorporation (Q3)

在获取用户/物品的增强属性信息后，需要采取有效的结合方法。

LLMRec 包括一个标准步骤：（1）增强语义投影，（2）协作上下文注入，以及（3）特性融入。

让我们深入研究每一项：1. 增强语义投影：通过投影用户/物品的增强属性信息到标准向量空间中，可以构建出更丰富的向量表示。这些向量表示不仅保留了用户/物品的原始特征，还包含了增强的属性信息。2. 协作上下文注入：通过将模型的其他部分与用户/物品的增强属性信息结合起来，可以进一步丰富用户/物品的表示。这种方法不仅可以提高模型的准确性，还可以防止模型受到增强属性信息的影响。3. 特征融入：通过将增强的属性信息融合到模型的其他特性中，可以使模型更好地理解和处理用户/物品的特性和行为。这种方法可以帮助模型更好地预测用户对物品的兴趣和喜好。

- 我们使用线性层和dropout来减少LLM增强的语义特征的维度，并将此类增强特征映射到自己的空间中。这个过程可以用

$$\bar{\mathbf{F}}_{\mathcal{A}} = \text{Linear}(\mathbf{F}_{\mathcal{A}})$$

表示，其中

$$\mathbf{f}_{\mathcal{A}} \in \mathbb{R}^{1 \times d_{LLM}}$$

是输入特征

$$\bar{\mathbf{f}}_{\mathcal{A}} \in \mathbb{R}^{1 \times d}$$

知乎

- 为了更好地利用协同过滤推荐系统中的用户-物品交互数据，我们可以采用多种策略进行上下文注入，从而获得更具代表性的用户和物品的表示。一种常见的上下文注入方法是基于用户的加权平均，即给定用户 u 和物品 i 之间的所有互动，我们将其权重乘以用户 u 的特征，然后取其平均值作为 u 关于 i 的上下文表示。这种方法可以有效地捕捉用户和物品之间的真实关系。

为了在增强特征 $\bar{\mathbf{f}}_{\mathcal{A},u}$ 和 $\bar{\mathbf{f}}_{\mathcal{A},i}$ 中注入高阶协作连接，LLMRec。

$\bar{\mathbf{F}}_{\mathcal{A}}$ 是输入向量 ($\mathbf{x}_u \mathbf{x}_i$) 的额外组合。

$$\mathbf{h}_u = \mathbf{e}_u + \omega_1 \cdot \sum_{k \in \mathcal{M} \cup \mathcal{A}_u} \frac{\bar{\mathbf{f}}_u^k}{\|\bar{\mathbf{f}}_u^k\|_2}; \quad \mathbf{h}_i = \mathbf{e}_i + \omega_1 \cdot \sum_{k \in \mathcal{M} \cup \mathcal{A}_i} \frac{\bar{\mathbf{f}}_i^k}{\|\bar{\mathbf{f}}_i^k\|_2}$$

该预测表示 \mathbf{h}_u 和 \mathbf{h}_i 是在 $\mathbb{R}^{1 \times d}$ 中，并且有用户档案 \mathbf{A}_u ，去偏好的项属性 \mathbf{A}_i ，以及原始多模态属性信息 \mathcal{M} 。具体的特征类型是 \mathbf{f}^k 。特征矢量通过使用聚合权重 ω_1 和 L_2 规范化来调整，以消除分布差距，并确保额外特征在推荐引擎编码中的有效性。

Training with Denoised Robustification (Q4)

我们将介绍如何通过模型将增强数据集集成到优化中。我们还引入了两种质量约束机制来增强边缘和节点特征：i) 噪声用户-内容交互修剪，ii) MAE (均方误差⁺) 基于的特征增强。

Augmented Optimization with Noise Pruning

我们在联合集合

$$\mathcal{E} \cup \mathcal{E}_{\mathcal{A}}$$

上训练我们的推荐系统，其中 \mathcal{E} 是原始训练集 $\mathcal{E}_{\mathcal{A}}$ 是从 LLM 中增强的集合。目标是通过优化带有增强监督信号的 BPR \mathcal{L}_{BPR} 损失来改善推荐系统的表现。这个损失函数旨在利用融入的 LLM 增强的用户偏好。

在联合训练集中选择 (u, i^+, i^-) 三元组。对于正负样本对预测得分，我们通过最终表示 \mathbf{h} 的内积获得，即

$$\hat{y}_{u,i^+} = \mathbf{h}_u \cdot \mathbf{h}_{i^+}, \hat{y}_{u,i^-} = \mathbf{h}_u \cdot \mathbf{h}_{i^-}$$

在总共由 LLM 生成的训练数据子集

$$\{LLM(\mathbb{P}_u) | u \in \mathcal{U}\}$$

中，我们会随机抽取一部分作为增强数据集 $\mathcal{E}_{\mathcal{A}}$ 。这是因为如果太多伪标签⁺被包含在增强数据集中，可能会导致结果的准确性下降。为了控制增强数据集的数量 $|\mathcal{E}_{\mathcal{A}}|$

我们会使用批大小 B 和率 ω_3 。正则化项 $|\Theta|^2$ 以 ω_2 为权值，可以用来防止过拟合。最后，我们使用 sigmoid 激活函数 $\sigma(\cdot)$ 引入非线性。

$$\sum_{(u,i^+,i^-)}^{(1-\omega_4) \cdot |\mathcal{E} \cup \mathcal{E}_{\mathcal{A}}|} - \text{SortAscend}(\log(\sigma(\hat{y}_{u,i^+} - \hat{y}_{u,i^-}))) [0:N] + \omega_2 \cdot \|\Theta\|^2$$

我们定义了一个函数 $\text{SortAscend}(\cdot)[0:N]$ ，该函数会按照从大到小的顺序对输入的值进行排序，并且只选择前 N 个最大的值。保留的数目 N 可以通过计算

$$N = (1 - \omega_4) \cdot |\mathcal{E} \cup \mathcal{E}_{\mathcal{A}}|$$

来确定，其中 ω_4 是一个比例因子。这种做法能够控制损失样本的删除，强化相关信号并同时减少噪声。这样做的好处是可以避免不可靠反向传播梯度的影响，从而让优化过程更加稳定和有效。

Enhancing Augmented Semantic Features via MAE

节点集合 \mathcal{V} 中的子集

$$\tilde{\mathcal{V}} \subset \mathcal{V}$$

并将它们的特征使用掩码标记符 $MASK$ （例如，学习型矢量或均值池化）来进行遮罩。

$$\tilde{\mathbf{f}}_{\mathcal{A}} = \begin{cases} \mathbf{f}_{[MASK]} & v \in \tilde{\mathcal{V}} \\ \bar{\mathbf{f}}_{\mathcal{A}} & v \notin \tilde{\mathcal{V}} \end{cases}$$

经过遮罩操作后的增强特征被表示为 $\tilde{\mathbf{f}}_{\mathcal{A}}$

如果节点被选择（ $\tilde{\mathcal{V}} \subset \mathcal{V}$ ），则将其替换为掩码标记符 $\mathbf{f}_{[MASK]}$ ；否则，它对应于原始增强特征 $\bar{\mathbf{f}}_{\mathcal{A}}$

为了加强特征编码器，我们引入了特征恢复损失 \mathcal{L}_{FR} ，该损失函数通过比较遮罩属性矩阵 $\tilde{\mathbf{f}}_{\mathcal{A},i}$

与原始增强特征矩阵 $\bar{\mathbf{f}}_{\mathcal{A}}$

并乘以一个缩放因子 γ 。特征恢复损失函数 \mathcal{L}_{FR} 如下所示：

$$\mathcal{L}_{FR} = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \left(\frac{\gamma \|\tilde{\mathbf{f}}_{\mathcal{A},i} - \bar{\mathbf{f}}_{\mathcal{A},i}\|^2}{\|\bar{\mathbf{f}}_{\mathcal{A},i}\|^2} \right)$$

$$\mathcal{L}_{FR} = \frac{1}{|\mathcal{V}|} \sum_{v \in \tilde{\mathcal{V}}} \left(1 - \frac{\tilde{\mathbf{f}}_{\mathcal{A}} \cdot \bar{\mathbf{f}}_{\mathcal{A}}}{\|\tilde{\mathbf{f}}_{\mathcal{A}}\| \cdot \|\bar{\mathbf{f}}_{\mathcal{A}}\|} \right) \gamma$$

最终优化目标是噪声剪枝的BPR损失 \mathcal{L}_{BPR} 和特征恢复（FR）损失 \mathcal{L}_{FR} 的加权和。

$$\min_{\theta, \psi} \mathcal{L}_{total} = \lambda_1 \mathcal{L}_{BPR} + \lambda_2 \mathcal{L}_{FR}$$

In-Depth Analysis of our LLMRec

LLM-based Augmentation Facilitates Optimization

Time Complexity.

我们分析时间复杂性。在语义特征的投影中，其时间复杂度为

$$\mathcal{O}(|\mathcal{U} \cup \mathcal{I}| \times d_{LLM} \times d)$$

基于图的协同上下文学习的GNN编码器需要的时间为

$$\mathcal{O}(L \times |\mathcal{E}^+| \times d)$$

BPR损失函数计算的时间复杂度为

$$\mathcal{O}(d \times |\mathcal{E} \cup \mathcal{E}_{\mathcal{A}}|)$$

而特征重建损失的时间复杂度为

$$\mathcal{O}(d \times |\tilde{\mathcal{V}}|)$$

其中 $|\tilde{\mathcal{V}}|$ 表示被掩码节点的数量。

Experimental Settings

Datasets.

MovieLens：我们使用了由ML-10M衍生的MovieLens数据集。该数据集的属性信息包括电影标题、年份和类型，以文本形式提供。视觉内容则来自网络爬虫获取的电影海报。

Netflix：我们通过网络爬虫收集了Netflix的数据集。其中的隐式反馈和基本属性来源于Kaggle上的Netflix Prize Data。

对于这两种数据集，我们都使用CLIP-ViT来编码视觉特征。基于LLM的数据增强技术。此研究使用OpenAI API访问LLMs进行数据增强。OpenAI平台文档提供了详细的信息。生成的模糊的隐式反馈使用"gpt-3.5-turbo-0613"聊天完成模型。收集的内容属性，如导演、国家和地区以及语言，使用的是同一种模型。

基于"gpt-3.5-turbo-16k"模型的用户轮廓，包括年龄、性别、喜欢的类型、不喜欢的类型、喜欢的导演、国家和地区。嵌入是通过"text-embedding-ada-002"模型实现的。在两个数据集上，增强策略的近似费用为15.65美元、20.40美元和3.12美元。

Implementation Details

实验使用24GB英伟达RTX 3090 GPU上的PyTorch进行实现。使用AdamW优化器，并使用不同的学习率范围 $[5e^{-5}, 1e^{-3}]$ 和 $[2.5e^{-4}, 9.5e^{-4}]$ 对Netflix和MovieLens进行训练。使用LLM的参数，我们选择温度从更大的值{0.0, 0.6, 0.8, 1}来控制生成文本的随机性，设置小值{0.0, 0.1, 0.4, 1}来鼓励可能的选择。选择stream为false以确保响应的完整性。

Evaluation Protocols

我们评估了在top-K内容推荐任务中的三个常见度量-----召回 (R@k)、归一化折扣累积增益 (N@k) 和精度 (P@k)。为避免测试采样可能带来的潜在偏见，我们采用了全排名策略。我们将结果从五次独立运行中报告，K设置为10、20和50（对于全排名是合理的）。我们计算与最佳表现基准的p-值进行统计显著性分析。

Baseline Description

为了全面比较，我们选择了四组不同的基线方法。

- i)通用协同过滤方法：MF-BPR、NGCF和LightGCN。
- ii)利用属性信息的方法：VBPR、MMGCN和GRCN。
- iii)数据增强方法：LATTICE。
- iv)自监督学习方法：CLCRec、MMSSL和MICRO。

Performance Comparison (RQ1)

Baseline	Netflix							MovieLens						
	R@10	N@10	R@20	N@20	R@50	N@50	P@20	R@10	N@10	R@20	N@20	R@50	N@50	P@20
General Collaborative Filtering Methods														
MF-BPR	0.0282	0.0140	0.0542	0.0205	0.0932	0.0281	0.0027	0.1890	0.0815	0.2564	0.0985	0.3442	0.1161	0.0128
NGCF	0.0347	0.0161	0.0699	0.0235	0.1092	0.0336	0.0032	0.2084	0.0886	0.2926	0.1100	0.4262	0.1362	0.0146
LightGCN	0.0352	0.0160	0.0701	0.0238	0.1125	0.0339	0.0032	0.1994	0.0837	0.2660	0.1005	0.3692	0.1209	0.0133
Recommenders with Side Information														
VBPR	0.0325	0.0142	0.0553	0.0199	0.1024	0.0291	0.0028	0.2144	0.0929	0.2980	0.1142	0.4076	0.1361	0.0149
MMGCN	0.0363	0.0174	0.0699	0.0249	0.1164	0.0342	0.0033	0.2314	0.1097	0.2856	0.1233	0.4282	0.1514	0.0147
GRCN	0.0379	0.0192	0.0706	0.0257	0.1148	0.0358	0.0035	0.2384	0.1040	0.3130	0.1236	0.4532	0.1516	0.0150
Data Augmentation Methods														
LATTICE	0.0433	0.0181	0.0737	0.0259	0.1301	0.0370	0.0036	0.2116	0.0955	0.3454	0.1268	0.4667	0.1479	0.0167
MICRO	0.0466	0.0196	0.0764	0.0271	0.1306	0.0378	0.0038	0.2150	0.1131	0.3461	0.1468	0.4898	0.1743	0.0175
Self-supervised Methods														
CLCRec	0.0428	0.0217	0.0607	0.0262	0.0981	0.0335	0.0030	0.2266	0.0971	0.3164	0.1198	0.4488	0.1459	0.0158
MMSSL	0.0455	0.0224	0.0743	0.0287	0.1257	0.0383	0.0037	0.2482	0.1113	0.3354	0.1310	0.4814	0.1616	0.0170
LLMRec	0.0531	0.0272	0.0829	0.0347	0.1382	0.0456	0.0041	0.2603	0.1250	0.3643	0.1628	0.5281	0.1991	0.0186
p-value	$2.9e^{-4}$	$3.0e^{-3}$	$9.4e^{-5}$	$1.5e^{-3}$	$2.8e^{-5}$	$2.2e^{-3}$	$3.4e^{-5}$	$2.8e^{-3}$	$1.6e^{-2}$	$3.3e^{-3}$	$4.1e^{-3}$	$1.9e^{-3}$	$1.8e^{-3}$	$1.8e^{-3}$
Improv.	13.95%	21.43%	8.51%	20.91%	5.82%	19.06%	7.89%	4.88%	10.52%	5.26%	10.90%	7.82%	9.06%	6.29%

表1比较了我们的模型与其他基线方法之间的性能。

框架的有效性。

- 属性信息的集成显著增强了推荐系统的能力。像MMSSL和MICRO这样的方法在有效使用多种属性信息模态和图神经网络方面表现突出。相比之下，仅依赖于有限内容的方法，如VBPR仅使用视觉特征，或者基于CF的架构如NGCF，没有属性信息，会产生显著降低的结果。这强调了有价值内容的重要性，因为仅依赖于ID对应的记录无法捕捉完整的u-i关系。
- 现有的方法，如LATTICE和MICRO等也利用属性信息进行数据增强，相比我们的模型显示了有限的改进。这可以归因于两个主要因素：(1)将属性信息与同质的关系（例如i-i或u-u）进行增强可能引入噪音，从而损害用户的偏好精确度。(2)这些方法通常不直接对u-i交互数据进行增强。
- 自监督模型，如MMSSL，MICRO等，在通过SSL信号解决稀疏问题方面显示出有前景的结果。但是，它们并没有超越我们的模型的性能，可能是因为它们增强的自监督信号可能与目标任务（即建模u-i交互）不一致。相反，我们直接建立BPR三元组来解决训练数据的稀缺性。

Ablation and Effectiveness Analyses (RQ2)

我们对提出的模型进行了功能验证，并在表中展示了结果。

Effectiveness of Data Augmentation Strategies.

	Metrics	R@10	N@10	R@20	N@20	R@50	N@50	P@20
Aug.	w/o-u-i	0.0477	0.0239	0.0791	0.0317	0.1376	0.0432	0.0037
	w/o-u	0.0423	0.0196	0.0656	0.0255	0.1192	0.0360	0.0033
	w/o-u&i	0.0309	0.0127	0.0602	0.0202	0.1051	0.0289	0.0030
Q.C.	w/o-prune	0.0504	0.0258	0.0786	0.0328	0.1363	0.0447	0.0039
	w/o-QC	0.0488	0.0244	0.0786	0.0318	0.1279	0.0416	0.0038
	LLMRec	0.0531	0.0272	0.0829	0.0347	0.1352	0.0456	0.0041

- (1). 不考虑用户-物品交互：禁用LLM增强的隐式反馈 $\mathcal{E}_{\mathcal{A}}$ 会导致显著下降。这表明，我们的模型通过包括上下文知识增加了潜在的监督信号，从而更好地理解用户的偏好。
- (2). 不考虑用户
- (3). 当我们将用户和内容的增强属性信息都删除时（ $\mathbf{F}_{\mathcal{A},u}, \mathbf{F}_{\mathcal{A},i,1}$ ），推荐准确率会降低。这一发现表明，基于LLM的增强属性信息为推荐系统提供了有价值的增强数据，帮助获得质量性和信息性的表示。

Impact of the Denoised Data Robustification.

- (4). 删除噪声修剪的结果会导致较差的性能。这表明删除过程中删除噪音隐式反馈信号有助于防止不正确的梯度下降过程。
- (5). 当同时移除隐式反馈的限制和语义特征的质量限制时（即移除噪声修剪和MAE），性能会恶化。这表明我们的去噪数据鲁棒化机制通过集成噪声修剪和语义特征增强而带来的好处。

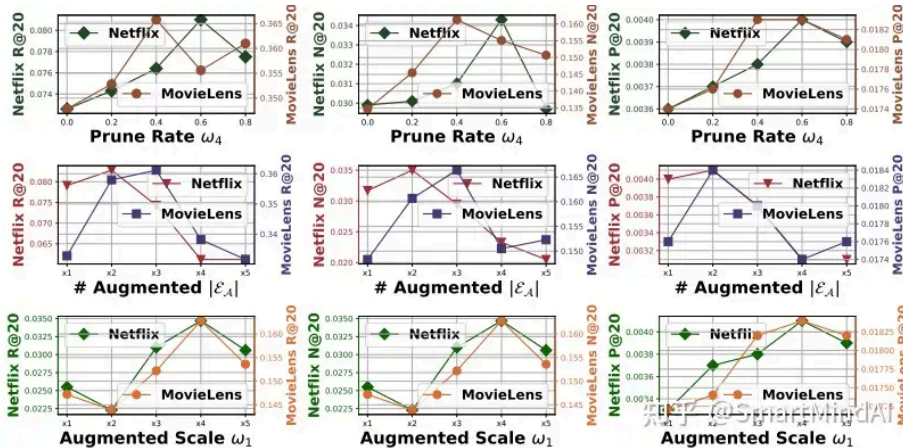
Hyperparameter Analysis (RQ3)

Parameters Affecting Augmented Data Quality.

- (6). LLM温度参数 τ 会影响文本的随机性。较高的值（>1.0）增加多样性和平创造性，较低的值（<0.1）导致更多的专注。我们使用 τ 从{0, 0.6, 0.8, 1}。如表所示，增大 τ 起初改善了大多数指标，随后下降。
- (7). LLM上采样参数 p 可以根据top-p参数选择令牌。较低的 p 值优先考虑可能的令牌，而较高的 p 值鼓励多样性。我们使用 p 从{0, 0.1, 0.4, 1}，并且较小的 p 值往往能得到更好的结果，可能是因为避免了未列出的候选选择。更高的 p 值会导致浪费的令牌，因为重复的LLM推理。
- (8). 我们使用 C 来限制LLM基于推荐项的候选数量。由于成本限制，我们探索{3, 10, 30}。表格显示 $C = 10$ 可以获得最佳结果。小值限制选择，大值增加推荐难度。

Sensitivity of Recommenders to the Augmented Data.

- (10). LLMRec ω_3 和批次大小 B 来控制每个批次的增强BPR训练数据样本的数量。 ω_3 在Netflix上设置为{0.0, 0.1, 0.2, 0.3, 0.4}, 在MovieLens上设置为{0.0, 0.2, 0.4, 0.6, 0.8}。当 ω_3 为零或过大时, 可能会导致不理想的实验结果。增加多样性和随机性可以使梯度下降更加稳健。
- (11). LLMRec ω_2 来控制特征幅度, 其值在Netflix上设置为{0.0, 0.8, 1.6, 2.4, 3.2}, 在MovieLens上设置为{0.0, 0.1, 0.2, 0.3, 0.4}。优化的结果取决于数据, 当 ω_2 太小或太大会导致不理想的实验结果, 如图(c)所示。



Model-agnostic Property (RQ4)

		Method	LATTICE	MICRO	MMSSL
Aug.	R@20		0.0821 \uparrow 11.40%	0.0835 \uparrow 9.29%	0.0833 \uparrow 11.11%
	N@20		0.0287 \uparrow 10.81%	0.0301 \uparrow 11.07%	0.0313 \uparrow 9.06%
	P@20		0.0039 \uparrow 8.33%	0.0041 \uparrow 7.89%	0.0041 \uparrow 10.81%

(12). 我们在Netflix上进行了模型无关性的实验, 以验证我们数据增强方法的应用性。具体来说, 我们将增强的隐式反馈 \mathcal{E}_A 和特征 $\mathbf{F}_{A,u}, \mathbf{F}_{A,i}$

引入了基线MICO, MMSSL和LATTICE。如表所示, 我们的LLM基于的数据提高了所有模型的性能, 证明了它们的有效性和可重用性。有些结果没有超过我们的模型, 可能是因为: i) 缺乏质量约束机制来调节增强数据的稳定性和质量, 以及ii) 在相同的向量空间中没有考虑到协同信号的建模, 正如Sec. 所述。

Cost/Improvement Conversion Rate (RQ5)

	Cost(USD)	R@10		N@10	
		Imp.(%)	CIR(%)	Imp.(%)	CIR(%)
U	10.92	25.53	233.79	38.78	355.13
I	1.96	2.31	117.86	1.12	57.14
U-I	8.26	11.32	137.05	13.81	167.19
LLMAug	21.14	13.95	65.99	21.43	101.37

(13).为了评估我们的增强策略的成本效益, 我们计算了CIR, 如表所示。CIR与三个数据增强策略的消减版本以及最佳基线 (来自表1和表2) 进行了比较。隐式反馈增强器的成本是指GPT-3.5 turbo 4K的价格。属性信息增强器的成本包括完成 (使用GPT-3.5 turbo 4K或16K) 和嵌入 (使用text-embedding-ada-002)。我们利用HuggingFace API工具进行分词和计数。表中的结果显示'U' (LLM用户特征) 是最具成本效益的策略, 总体投资是值得的。

发布于 2024-03-14 14:29 · IP 属地北京

LLM (大型语言模型) 推荐系统 图模型



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



LLM建模了什么，为什么需要RAG

远洋之帆 发表于大模型原理

2023关于LLM的总结

- 1、如果只做单轮的场景，完全可以只做文本向量的检索，要不要放LLM不影响单轮。如果想兼顾其它通用知识的回答，可以做RAG。
- 2、如果要做多轮的场景，可以判断一下这个场景，是不是属于很常...

算法观察者

面向LLM应用的 中间层平台产品【2024.8】

0、前言本文是 谈小微应用开发平台（包括Agent构建平台）【2024.8】的姊妹篇，这两篇虽然是从两个不同的方向讨论的两种不同的平台，但它们无论在技术上还是未来发展上重合点都不少。我...
孔某人 发表于孔某人的低...

一窥廉价LLM时代的Age略框架设计【2024.7】

TL;DR介绍一种新的设计思路量的prompt撬动尽可能多的作量。1、前提条件1.1、高LLM的到来虽说现在各家LLM商都很难做到持续的快速提升从整个生态来说每个季度整...
孔某人 发表于孔某...