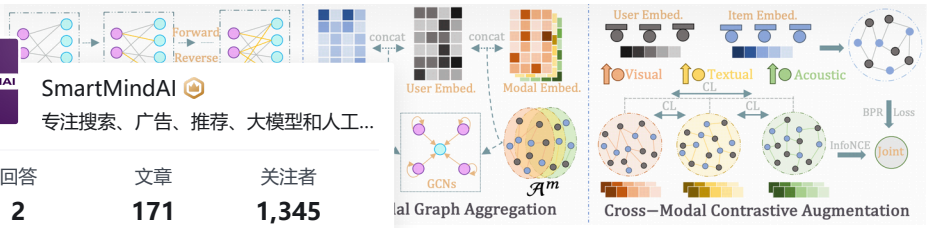


赋能推荐系统创新

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

多模态推荐系统[+]在电子商务和内容分发应用中至关重要，旨在通过处理视觉、听觉和文本等模态内容，捕捉用户在精细模态层面的偏好。多种研究路径整合多模态内容到推荐系统中，如VBPR、ACF、MMGCN、GRCN和LATTICE，其利用图神经网络[+]（GNN）处理模态信息。然而，大多数现有的多模态推荐系统在训练过程中依赖于高质量的标记数据，但在实际应用的推荐场景中，这类数据往往稀少，限制了模型生成准确代表复杂偏好的嵌入表示。

为应对数据稀疏性问题，有研究尝试从无标注数据中生成监督信号。它们通常在增强过程中忽略了对多模态特性的深入考虑，导致在模态感知用户偏好的表现上受限。因此，需要一种适应性的模态感知增强方法，以实现更精准的自监督学习。这种方法应确保从无标注数据中生成的监督信号与多模态推荐任务相关，有效整合多模态上下文信息与相关协作信号，以学习用户偏好。

考虑到现有解决方案的局限性和挑战，我们提出了一种名为多模态图扩散模型[+]的推荐新方法（DiffMM）。受到近期在图像合成任务中扩散模型（DMs）进展的启发，我们的方法专注于通过利用扩散模型的生成能力，生成一个模态感知的用户-项目图。这使得多模态知识能够有效地被引入到用户-项目交互的建模中。具体来说，我们采用分步的破坏过程，逐步向初始的用户-项目交互图中引入随机噪声。然后，通过一个反过程，我们迭代地恢复被破坏的图，该图在T步中累积了噪声，以获得原始的用户-项目图结构。为了进一步指导反过程并生成一个模态感知的用户-项目图，我们引入了一个简单而有效的模态感知信号注入机制。通过生成模态感知的用户-项目图，我们引入了一种模态感知的图神经范式来进行多模态图聚合。这使我们能够有效地捕捉与不同模态相关联的用户偏好。此外，我们提出了一种跨模态对比学习框架，探讨了不同模态中用户-项目交互模式的一致性，进一步增强了推荐系统在多模态上下文学习方面的能力。

本文贡献:

(i) 提出多模态图扩散模型，解决现有推荐方法局限。

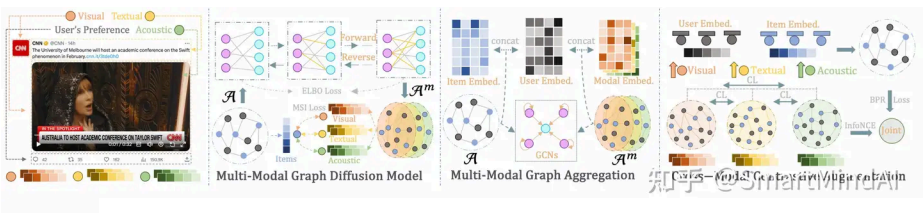
(ii) 利用扩散模型生成能力，构建模态感知用户-项目图，实现多模态知识转移。

(iii) 引入模态感知信号注入机制与图神经范式，增强不同模态偏好捕捉。

(iv) 设计跨模态对比学习框架，提升多模态上下文学习效率与推荐系统性能。

Methodology

Preliminaries



多模态特征协作图

在基于图神经网络（GNN）的协作过滤技术的成功应用基础上，我们的模型 Model 有效地利用了图结构数据来驱动一个全面的多模态推荐系统。我们用图 $\mathcal{G}=(u,i)|u\in\mathcal{U},i\in\mathcal{I}$，其中$\mathcal{U}$和$\mathcal{I}$分别代表用户集合和物品集合。边$(u,i)$表示用户$u$与物品$i$的交互行为。为了丰富交互图$\mathcal{G}$的模态，每个物品$i$都具有特定模态的特征向量

$$\hat{\mathbf{F}}_i=\hat{\mathbf{f}}_i^1,\ldots,\hat{\mathbf{f}}_i^m,\ldots,\hat{\mathbf{f}}_i^{|\mathcal{M}|}$$

每个向量$\hat{\mathbf{f}}_i^m$在维度d_m上包含了物品i的模态m特征，其中模态的范围为从 1 到$\mathcal{M}$。这些特征属于模态集合$\mathcal{M}$，而维度d_m表示了模态m特征的深度。

我们的目标是设计一个多模态推荐系统，该系统能够高效地识别用户与项目间的关联，并通过学习函数f来实现这一目标，同时利用项目的多模态特征。系统旨在估计用户u采用项目i的概率，预测值$\hat{y}_{ui}$由函数f根据多模态交互+网络$\mathcal{G}^{\mathcal{M}}$计算得出。

Multi-Modal Graph Diffusion Model

受到扩散模型在生成输出中保持关键数据模式成功启发，我们的模型框架提出了一种新颖的方法用于多模态推荐系统。具体而言，我们引入了一个多模态图扩散模块，用于生成包含多模态数据的用户-项目交互图，从而增强用户与项目协作的信号的建模。我们的框架专注于解决多模态推荐系统中无关或噪声模态特征的负面影响。

为了实现这一目标，我们使用了感知模态的去噪扩散概率模型，将用户与项目协作的信号与多模态数据统一起来。具体而言，我们逐步添加噪声到原始用户-项目图中的交互，并通过概率扩散过程+进行迭代学习，以恢复原始交互。这种迭代去噪训练在生成用户-项目交互图的同时，有效地融入了多模态信息，同时减少了噪声模态特征的负面影响。

此外，为了实现模态感知的图生成，我们开发了一种新颖的模态感知信号注入机制，指导了交互恢复的过程。这个机制在将多模态信息有效融入用户-项目交互图生成中扮演了关键角色。通过利用扩散模型的能力，我们模型框架提供了增强多模态推荐器的稳健和有效解决方案。

Probabilistic Diffusion Paradigm with Interactions

我们的图扩散范式提出两种关键过程针对用户-项目交互。前向过程引入高斯噪声破坏原始图，模拟嘈杂影响。反向过程则专注于学习并修复破坏的图连接结构，以恢复原始的用户与项目之间的交互。

正向图扩散过程关注用户u与项目集$\mathcal{I}$的交互序列 $\mathbf{a}_u=[\mathbf{a}_u^0,\mathbf{a}_u^1,\cdots]$，其中$\mathbf{a}_u^i$为1或0表示用户$u$是否与第$i$个项目交互。以$\mathbf{a}_u$为起点，在$T$步内，逐步通过引入高斯噪声，构建了从$\boldsymbol{\alpha}_1$到$\boldsymbol{\alpha}_T$的序列。每一步转换$\boldsymbol{\alpha}_t$到$\boldsymbol{\alpha}_{t-1}$都包含高斯噪声的参数化。

$$q(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1})=\mathcal{N}(\boldsymbol{\alpha}_t;\sqrt{1-\beta_t}\boldsymbol{\alpha}_{t-1},\beta_t\mathbf{\emph{I}})$$

扩散过程包含t差分步骤，表示为t属于$\{1,\ldots,T\}$。随着T趋向于无穷大时$\boldsymbol{\alpha}_T$收敛到标准高斯分布。通过重参数技巧和独立高斯噪声的加性性质，我们可以直接从$\boldsymbol{\alpha}_0$得到$\boldsymbol{\alpha}_t$。形式上，表达如下：

$$q(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_0)=\mathcal{N}(\boldsymbol{\alpha}_t;\sqrt{\bar{\gamma}_t}\boldsymbol{\alpha}_0,(1-\bar{\gamma}_t)\mathbf{\emph{I}})$$

为了表示t时刻的平均噪声率，我们引入了参数$\gamma_t=1-\beta_t$和 $\bar{\gamma}_t=\prod_{t'=1}^{t}\gamma_{t'}$，将$\boldsymbol{\alpha}_t$重新定义为在t时刻$\boldsymbol{\alpha}$等于从$\boldsymbol{\alpha}_0$中抽取的噪声的平方根与$\boldsymbol{\epsilon}$的平方根的线性组合，其中$\boldsymbol{\epsilon}$服从标准多维正态分布。

采用线性噪声调节器控制$1-\bar{\gamma}_t$中的噪声量，实现从第1个到第T个$\boldsymbol{\alpha}$的噪声调节。

$$1-\bar{\gamma}_t=s\cdot\left[\gamma_{min}+\frac{t-1}{T-1}(\gamma_{max}-\gamma_{min})\right],t\in\{1,\cdots,T\}$$

$s\in[0,1]$属于区间$0,1$，调节着噪声的尺度；而γ_{min}和γ_{max}（均在区间（0, 1）内），则是添加噪声的上下

目标是在反向步骤中从$\boldsymbol{\alpha}_t$中去除噪声，恢复原始状态$\boldsymbol{\alpha}_{t-1}$。这使多模态扩散能够有效地捕捉生成过程中的细微变化。从$\boldsymbol{\alpha}_T$开始，逐步执行去噪转换，逐步恢复用户与项目的交互。

$$p_\theta(\boldsymbol{\alpha}_{t-1}|\boldsymbol{\alpha}_t) = \mathcal{N}(\boldsymbol{\alpha}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{\alpha}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{\alpha}_t, t))$$

$\boldsymbol{\mu}_\theta(\boldsymbol{\alpha}_t, t)$ 和 $\boldsymbol{\Sigma}_\theta(\boldsymbol{\alpha}_t, t)$ 分别是预测的高斯分布的均值和协方差。这两个参数是由具有可学习参数θ的两个神经网络输出的。

Modality-aware Optimization for Graph Diffusion

通过图扩散训练优化模型，旨在最大化用户对项目交互观察$\boldsymbol{\alpha}_0$的负对数似然的证据下界（ELBO），以引导反向图扩散过程。

$$\mathcal{L}_{elbo} = \mathbb{E}_{q(\boldsymbol{\alpha}_0)}[-\log p_\theta(\boldsymbol{\alpha}_0)] \leq \sum_{t=0}^{T} \mathbb{E}_q[\mathcal{L}_t], t \in \{0, \cdots, T\}$$

有三种不同情况，$\mathcal{L}_t$，分别是.

$$\mathcal{L}_t = \begin{cases} -\log p_\theta(\boldsymbol{\alpha}_0|\boldsymbol{\alpha}_1), & t = 0 \\ D_{KL}(q(\boldsymbol{\alpha}_T|\boldsymbol{\alpha}_0)||p(\boldsymbol{\alpha}_T)), & t = T \\ D_{KL}(q(\boldsymbol{\alpha}_{t-1}|\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_0)||p_\theta(\boldsymbol{\alpha}_{t-1}|\boldsymbol{\alpha}_t), & t \in \{1, 2, \cdots, T-1\} \end{cases}$$

其中$\mathcal{L}_0$是基于$\boldsymbol{\alpha}_0$的负重构损失$\mathcal{L}_T$是一个在优化过程中无需考虑的常数$\mathcal{L}_t$（$t \in \{1, 2, \cdots, T-1\}$）通过正则化 $p_\theta(\boldsymbol{\alpha}_{t-1}|\boldsymbol{\alpha}_t)$ 使其与真实转移步骤 $q(\boldsymbol{\alpha}_{t-1}|\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_0)$对齐。

优化图扩散，设计神经网络在反向过程中执行去噪操作以提升性能。目标如式t所示，通过KL散度[+]迫使 $p_\theta(\boldsymbol{\alpha}_{t-1}|\boldsymbol{\alpha}_t)$，接近 $q(\boldsymbol{\alpha}_{t-1}|\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_0)$，利用贝叶斯规则[+] $q(\boldsymbol{\alpha}_{t-1}|\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_0)$ 可表示为：

条件分布[+]

$$q(\boldsymbol{\alpha}_{t-1}|\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_0)$$

在$\boldsymbol{\alpha}_t$对于$\boldsymbol{\alpha}_{t-1}$和$\boldsymbol{\alpha}_0$的优化图扩散过程中。

$$q(\boldsymbol{\alpha}_{t-1}|\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_0) \propto \mathcal{N}(\boldsymbol{\alpha}_{t-1}; \tilde{\boldsymbol{\mu}}(\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_0, t), \sigma^2(t)\mathbf{I})$$

$$\tilde{\boldsymbol{\mu}}(\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_0, t) = \frac{\sqrt{\gamma_t}(1-\bar{\gamma}_{t-1})}{1-\bar{\gamma}_t}\boldsymbol{\alpha}_t + \frac{\sqrt{\bar{\gamma}_{t-1}}(1-\gamma_t)}{1-\bar{\gamma}_t}\boldsymbol{\alpha}_0,$$
$$\sigma^2(t) = \frac{(1-\gamma_t)(1-\bar{\gamma}_{t-1})}{1-\bar{\gamma}_t}.$$

在这里，

$$\tilde{\boldsymbol{\mu}}(\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_0, t)$$

和$\sigma^2(t)\mathbf{I}$分别表示在给定$\boldsymbol{\alpha}_t$、$\boldsymbol{\alpha}_0$和时间t下

$$q(\boldsymbol{\alpha}_{t-1}|\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_0)$$

的均值和协方差。我们直接设定

$$\sum_\theta(\boldsymbol{\alpha}_t, t) = \sigma^2(t)\mathbf{I}$$

之后t时刻的损失函数$\mathcal{L}_t$如下：

$$\mathcal{L}_t = \frac{1}{2\sigma^2(t)}[||\boldsymbol{\mu}_\theta(\boldsymbol{\alpha}_t, t) - \tilde{\boldsymbol{\mu}}(\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_0, t)||_2^2],$$

这被

$$\boldsymbol{\mu}_\theta(\boldsymbol{\alpha}_t, t)$$

促使接近

$$\tilde{\boldsymbol{\mu}}(\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_0, t)$$

通过遵循等式，同样地我们可以分解

$$\boldsymbol{\mu}_\theta(\boldsymbol{\alpha}_t, t)$$

以保持风格一致。

$$\boldsymbol{\mu}_\theta(\boldsymbol{\alpha}_t, t) = \frac{\sqrt{\gamma_t}(1-\bar{\gamma}_{t-1})}{1-\bar{\gamma}_t}\boldsymbol{\alpha}_t + \frac{\sqrt{\bar{\gamma}_{t-1}}(1-\gamma_t)}{1-\bar{\gamma}_t}\hat{\boldsymbol{\alpha}}_\theta(\boldsymbol{\alpha}_t, t)$$

根据$\boldsymbol{\alpha}_t$和t预测的$\boldsymbol{\alpha}_0$，我们有： 通过将等式1和等式2代入等式3，得到：

$$\hat{\boldsymbol{\alpha}}_\theta(\boldsymbol{\alpha}_t, t) = \text{预测的}\boldsymbol{\alpha}_0$$

等式$\mathbf{1}$ 等式$\mathbf{2}$ 方程$\mathbf{3}$

$$\mathcal{L}_t = \frac{1}{2}\left(\frac{\bar{\gamma}_{t-1}}{1-\bar{\gamma}_{t-1}} - \frac{\bar{\gamma}_t}{1-\bar{\gamma}_t}\right)||\hat{\boldsymbol{\alpha}}_\theta(\boldsymbol{\alpha}_t, t) - \boldsymbol{\alpha}_0||_2^2$$

$$\hat{\boldsymbol{\alpha}}_\theta(\boldsymbol{\alpha}_t, t)$$

是根据$\boldsymbol{\alpha}_t$和时间步长t预测的$\boldsymbol{\alpha}_0$。我们使用神经网络实现这一预测功能。具体地，通过多层感知器⁺（MLP）实例化

$$\hat{\boldsymbol{\alpha}}_\theta(\cdot)$$

接受$\boldsymbol{\alpha}_t$和时间步长t的输入，以预测$\boldsymbol{\alpha}_0$。 对于$\mathcal{L}_0$的计算方式：

$$\mathcal{L}_0 = \text{某种函数}(\boldsymbol{\alpha}_t, t)$$

$$\mathcal{L}_0 = ||\hat{\boldsymbol{\alpha}}_\theta(\boldsymbol{\alpha}_1, 1) - \boldsymbol{\alpha}_0||_2^2,$$

通过未加权的

$$\|\hat{\boldsymbol{\alpha}}_\theta(\boldsymbol{\alpha}_1, 1) - \boldsymbol{\alpha}_0\|_2^2$$

来估算高斯对数似然度 $\log p(\boldsymbol{\alpha}_0|\boldsymbol{\alpha}_1)$

在时间步长t的均匀采样中，取值于$\{1, 2, \cdots, T\}$，同时降低计算成本，以减少计算开销。

$$\mathcal{L}_{elbo} = \mathbb{E}_{t\sim\mathcal{U}(1,T)}\mathbb{E}_{q(\boldsymbol{\alpha}_0)}[||\hat{\boldsymbol{\alpha}}_\theta(\boldsymbol{\alpha}_t, t) - \boldsymbol{\alpha}_0||_2^2]$$

模态感知信号注入

多模态图扩散旨在通过模态感知的用户项图增强推荐系统。为此，我们引入模态感知信号注入（MSI）机制，生成与模态相对应的多个用户项图。具体操作包括对齐并聚合项目模态特征$\mathbf{e}_m^i$（请参阅第），以及预测的模态感知用户项交互概率$\hat{\boldsymbol{\alpha}}_0$。同时，我们聚合项目id嵌入$\mathbf{e}^i$与观察到的用户项交互$\boldsymbol{\alpha}_0$。最终，我们计算了上述两个聚合嵌入之间的均方误差损失，并将其与$\mathcal{L}_{elbo}$结合进行优化。

对于模态m，MSI的均方误差损失$\mathcal{L}_{msi}^m$如下：

$$\mathcal{L}_{msi}^m = ||\hat{\boldsymbol{\alpha}}_0 \cdot \mathbf{e}_m^i - \boldsymbol{\alpha}_0 \cdot \mathbf{e}^i||_2^2$$

此loss function增强了扩散模块的模态信息丰富度，并提升了模型性能。

Inference of Multi-Modal Graph Diffusion Model.

我们设计了一种简单推理策略，与扩散训练结合用于预测用户-项目交互，不同于其他Diffusion Models在反向生成阶段使用随机高斯噪声，策略首先在正向过程对$\boldsymbol{\alpha}_0$进行T^l步破坏，然后设置

$\hat{\boldsymbol{\alpha}}_T = \boldsymbol{\alpha}_{T'}$ 执行反向去噪操作，共T步，过程中忽略方差。反向去噪阶段使用 $\hat{\boldsymbol{\alpha}}_{t-1} = \mu_\theta(\hat{\boldsymbol{\alpha}}_t, t)$

进行确定性推理。最后，利用$\hat{\boldsymbol{\alpha}}_0$重建用户-项目图结构。对于用户$u$： $\hat{\mathbf{a}}_u = \hat{\boldsymbol{\alpha}}_0$

包含前k个排名最高的交互值。从$\hat{\mathbf{a}}_u^i$中选择k个最高值，并在用户u与项目$i \in \mathcal{I}$之间建立交互。对于模态m的结果用户-项目图，表示为$\mathcal{A}^m$。

Cross-Modal Contrastive Augmentation

在多模态推荐场景中，用户在视觉、文本和声学等项目模态的交互中表现出一致性。例如，一段短视频的视觉和声学特征共同吸引用户。因此，用户偏好可能在视觉和声学方面交织复杂。 为提升推荐系统性能，我们设计了两种模态感知对比学习方法。一种以不同模态视图为锚点，另一种则以主要模视为锚点。

Modality-aware Contrastive View.

本部分介绍生成特定模态的用户/项目嵌入的方法，采用基于GNN的表示学习。通过模态感知生成模块处理原始特征向量$\hat{\mathbf{f}}^m$，获得对齐的项目模态特征$\mathbf{e}_m^i$。这些特征通过模态感知用户-项目图$\mathcal{A}^m$上的信息聚合整合到统一维度空间。在跨模对比学习中，这些维度对齐的项目模态特征用于生成模态感知的用户/项目嵌入。

$$\mathbf{e}_m^i = Norm(Trans(\hat{\mathbf{f}}^m)), m \in \mathcal{M}$$

$Norm(\cdot)$归一化操作与$Trans(\cdot)$多层感知器（MLP）进行从d_m到d的映射转换。接着，通过应用用户嵌入$\mathbf{E}^u$和项目模态特征$\mathbf{E}_m^i$，我们进行信息聚合，得到模态感知聚合嵌入$\mathbf{z}^m$。

$$\mathbf{z}_u^m = \bar{\mathcal{A}}_{u,*}^m \mathbf{E}^u, \ \mathbf{z}_i^m = \bar{\mathcal{A}}_{*,i}^m \mathbf{E}_m^i, \ \bar{\mathcal{A}}_{u,i}^m = \mathcal{A}_{u,i}^m / \sqrt{|\mathcal{N}_u^m||\mathcal{N}_i^m|}$$

在原始交互图$\mathcal{A}$中，生成的模态感知图$\mathcal{A}^m$的归一化邻接关系位于实数矩阵$U \times I$中。在$\mathcal{A}$中进一步探索了多模态信息意识下的高阶协作效应。

$$\mathbf{Z}_{l+1}^m = \bar{\mathcal{A}} \cdot \mathbf{Z}_l^m, \quad \mathbf{Z}_0^m = \mathbf{Z}^m$$

在多层GNN中，特定层的嵌入通过求和池化聚合产生输出：

$$\bar{\mathbf{Z}}^m = \sum_{l=0}^{L} \mathbf{Z}_l^m$$

其中L表示图层数量。$\mathbf{Z}_l^m$与$\mathbf{Z}_{l+1}^m$分别代表第l层和第$(l+1)$层的嵌入。同时$\bar{\mathcal{A}}^m$是$\mathcal{A}^m$的规范化邻接矩阵[+]。

Modality-aware Contrastive Augmentation.

使用模态意识的对比视图，我们采用了两种不同的对比方法。其中一种方法利用不同模态的视图作为锚点，而另一种则使用主要视图作为锚点。

模态视图作为锚点。

基于不同模态下的用户行为相关性，我们将不同模态的嵌入视为视图（例如，模态m_1和m_2的嵌入），并使用InfoNCE损失来最大化两种模态视图之间的互信息。 此外，我们使用不同用户之间的嵌入作为负对（例如，用户u和用户v）。 正式地，第一种对比学习损失如下定义：

$$\mathcal{L}_{cl}^{user} = \sum_{m_1 \in \mathcal{M}} \sum_{m_2 \in \mathcal{M}} \sum_{u \in \mathcal{U}} -\log \frac{\exp(s(\bar{\mathbf{z}}_u^{m_1}, \bar{\mathbf{z}}_u^{m_2})/\tau)}{\sum_{v \in \mathcal{U}} \exp(s(\bar{\mathbf{z}}_u^{m_1}, \bar{\mathbf{z}}_v^{m_2})/\tau)}$$

其中$s(\cdot)$表示相似性函数τ是温度系数。这个对比损失函数旨在最大化正向对的共识，并最小化负向对的共识。

我们的第二种对比学习方法是通过利用不同模态下的用户行为模式来指导和提升目标推荐任务的学习。 为了实现这一目标，我们采用主任务的嵌入$\bar{\mathbf{H}}$，并使用InfoNCE损失与各种模态视图进行互信息的最大

$$\mathcal{L}_{cl}^{user} = \sum_{m \in \mathcal{M}} \sum_{u \in \mathcal{U}} -\log \frac{\exp(s(\bar{\mathbf{h}}_u, \bar{\mathbf{z}}_u^m)/\tau)}{\sum_{v \in \mathcal{U}} \exp(s(\bar{\mathbf{h}}_u, \bar{\mathbf{z}}_v^m)/\tau)}$$

采用类似方法，计算用户维度与项目维度的对比学习损失，分别为$\mathcal{L}_{cl}^{user}$和$\mathcal{L}_{cl}^{item}$。通过整合这两项损失，得到跨模态对比学习的整体目标函数，即综合对比学习损失等于用户维度和项目维度的对比学习损失之和：

$$\mathcal{L}_{cl} = \mathcal{L}_{cl}^{user} + \mathcal{L}_{cl}^{item}$$

Multi-Modal Graph Aggregation

在d维空间生成用于预测的最终用户表示$\bar{\mathbf{h}}_u$和项目表示$\bar{\mathbf{h}}_i$。首先，聚合所有模态感知嵌入$\hat{\mathbf{f}}^m$和对应图$\mathcal{A}^m$。然后，通过在原始用户-项目交互图$\mathcal{A}$上进行消息传递，探索高阶协作信号。

具体地，使用特定公式从$\hat{\mathbf{f}}^m$中提取每个模态对齐的项目特征$\mathbf{e}_m^i$，确保一致性。接着，对$\bar{\mathcal{A}}$和$\bar{\mathcal{A}}^m$进行聚合，获取每个模态的表示$\hat{\mathbf{z}}^m$。

$$\hat{\mathbf{z}}_u^m = \bar{\mathcal{A}}_{u,*} \cdot \mathbf{E}^u + \bar{\mathcal{A}}_{u,*} \cdot (\bar{\mathcal{A}}_{u,*} \cdot \mathbf{E}^u) + \bar{\mathcal{A}}_{u,*}^m \cdot \mathbf{E}^u,$$
$$\hat{\mathbf{z}}_i^m = \bar{\mathcal{A}}_{i,*} \cdot \mathbf{E}_m^i + \bar{\mathcal{A}}_{i,*} \cdot (\bar{\mathcal{A}}_{i,*} \cdot \mathbf{E}^i) + \bar{\mathcal{A}}_{*,i}^m \cdot \mathbf{E}^i,$$

使用所有单模态特征表示$\hat{\mathcal{Z}}^m$（其中$m \in \mathcal{M}$），我们通过加和形成多模态特征表示。

由于每个模态在形成的多模态特征表示中可能具有不同的影响力，我们引入可学习参数化向量κ_m来控制模态m的表示在多模态特征表示中的权重，以实现适当的平衡。

$$\mathbf{h}_u = \sum_{m \in \mathcal{M}} \kappa_m \hat{\mathbf{z}}_u^m, \quad \mathbf{h}_i = \sum_{m \in \mathcal{M}} \kappa_m \hat{\mathbf{z}}_i^m$$

借助图神经网络在$\bar{\mathcal{A}}$中传递消息，探索高阶协作信号。

$$\mathbf{H}_{l+1} = \bar{\mathcal{A}} \cdot \mathbf{H}_l, \quad \mathbf{H}_0 = \mathbf{H}_u \text{ or } \mathbf{H}_i$$

$\mathbf{H}_l$和$\mathbf{H}_{l+1}$分别是第l层和第$l+1$层的特定层嵌入。因此$\mathbf{H}$的维度是$\mathbb{R}^{I \times d}$或$\mathbb{R}^{U \times d}$。 通过求和池化操作，这些层嵌入被合并，生成最终嵌入$\bar{\mathbf{H}}$。

$$\bar{\mathbf{H}} = \sum_{l=0}^{L} \mathbf{H}_l + \omega Norm(\mathbf{H}_0)$$

ω为超参数，用于调节归一化$\mathbf{H}_0$的权重，以避免过平滑。借助最终的嵌入： $\hat{y}_{u,i} = \bar{\mathbf{h}}_u^{\mathrm{T}} \cdot \bar{\mathbf{h}}_i$

Multi-Task Model Training

因此，定义用于优化模态m的扩散模块的损失函数如下：

$\mathcal{L}_m = [\text{具体表述损失函数}]$

其目标是： $\min_{\text{参数}} \mathcal{L}_m$

确保优化过程：

针对特定模态m的扩散模块进行

$$\mathcal{L}_{dm}^m = \mathcal{L}_{elbo} + \lambda_0 \mathcal{L}_{msi}^m$$

λ_0是一个超参数，用于调节 MSI 的力度。 在推荐任务中，我们采用了对比损失$\mathcal{L}_{cl}$的贝叶斯个性化排名损失策略。具体而言，我们的贝叶斯个性化排名损失$\mathcal{L}_{bpr}$如下：

$$\mathcal{L}_{bpr} = \sum_i \sum_{j \in \mathcal{N}(i)} \log(1 + \exp(-\mathcal{L}_{cl}(p_i, p_j)))$$

其中$\mathcal{N}(i)$表示与用户i相关的其他用户集合p_i和p_j分别表示用户i和用户j对物品的偏好评分。通过优化这个损失函数，我们能够提升推荐系统的个性化能力，从而更有效地调节 MSI 的力度。

$$\mathcal{L}_{bpr} = \sum \quad \log \sigma(\hat{y}_{\quad} - \hat{y}_{\quad})$$

$\mathcal{O}$集合包含了训练数据，其中$\mathcal{O}^+$和$\mathcal{O}^-$分别表示正样本和负样本。$\mathcal{O}^-$表示未观察到的交互集合，等同于$\mathcal{U}$和$\mathcal{I}$与$\mathcal{O}^+$的笛卡尔积除以$\mathcal{O}^+$。推荐任务的综合优化损失定义为：

$$\mathcal{O}^- = \frac{\mathcal{U} \times \mathcal{I}}{\mathcal{O}^+}$$

$$\mathcal{L}_{rec} = \mathcal{L}_{bpr} + \lambda_1 \mathcal{L}_{cl} + \lambda_2 ||\Theta||_2^2$$

Θ代表可训练参数λ_1和λ_2分别调控对比学习强度和L_2正则化，用于调整模型参数。

Experimental Settings

Evaluation Dataset

实验中使用了三个公开多模态数据集：抖音短视频、亚马逊婴儿产品、亚马逊体育商品。具体信息请参阅附录。

Evaluation Protocols

通过使用K召回率（R\@K）、K精确率（P\@K）和归一化累计增益（NDCG\@K）三种指标，我们评估了最佳K推荐结果的准确性。遵循以往研究，我们采用了全排名策略进行项目评估。在测试集上，对所有用户报告的平均分数作为评估指标。

Compared Baselines.

在评估中，我们对比了-BPR、广泛使用的基于GNN的CF模型、LightGCN、DiffRec以及SGL、NCL、HCCF等自监督推荐方法。这些自监督推荐方法为近期的研究成果。基准详情见附录。

Performance Comparison (RQ1)

Table 1: Performance comparison on TikTok, Amazon datasets in terms of *Recall@20*, *Precision@20*, and *NDCG@20*.

Dataset	Metric	MF-BPR	NGCF	LightGCN	SGL	NCL	HCCF	VBPR	LGCN-*M*	MMGCN	GRCN	LATTICE	CLCRec	MMGCL	SLMRec	BM3	DiffMM
TikTok	Recall@20	0.0346	0.0604	0.0653	0.0603	0.0658	0.0662	0.0380	0.0679	0.0730	0.0804	0.0843	0.0621	0.0799	0.0845	0.0957	**0.1129**
	Precision@20	0.0017	0.0030	0.0033	0.0030	0.0034	0.0029	0.0018	0.0034	0.0036	0.0036	0.0042	0.0032	0.0037	0.0042	0.0048	**0.0056**
	NDCG@20	0.0130	0.0238	0.0282	0.0238	0.0269	0.0267	0.0134	0.0273	0.0307	0.0350	0.0367	0.0264	0.0326	0.0353	0.0404	**0.0456**
Amazon-Baby	Recall@20	0.0440	0.0591	0.0698	0.0678	0.0703	0.0705	0.0486	0.0726	0.0640	0.0754	0.0829	0.0610	0.0758	0.0765	0.0839	**0.0975**
	Precision@20	0.0024	0.0032	0.0037	0.0036	0.0038	0.0037	0.0026	0.0038	0.0032	0.0040	0.0044	0.0032	0.0041	0.0043	0.0044	**0.0051**
	NDCG@20	0.0200	0.0261	0.0319	0.0296	0.0311	0.0308	0.0213	0.0329	0.0284	0.0336	0.0368	0.0284	0.0331	0.0325	0.0361	**0.0411**
Amazon-Sports	Recall@20	0.0430	0.0695	0.0782	0.0779	0.0765	0.0779	0.0582	0.0705	0.0638	0.0833	0.0915	0.0665	0.0775	0.0838	0.0977	**0.1017**
	Precision@20	0.0023	0.0037	0.0042	0.0041	0.0040	0.0041	0.0031	0.0035	0.0034	0.0044	0.0048	0.0035	0.0046	0.0043	0.0051	**0.0054**
	NDCG@20	0.0202	0.0318	0.0369	0.0361	0.0349	0.0361	0.0265	0.0324	0.0279	0.0377	0.0424	0.0301	0.0409	0.0376	0.0442	**0.0458**

表展示了性能比较的评估结果。

- 我们的模型在各类数据集上始终表现出色，超越所有基准。这种优势得益于跨模态对比学习的增强，有效整合了多模态信息，以及多模态图聚合组件的应用。多模态推荐系统强调整合多模态上下文的重要性，已经超越了基于图的协同过滤+模型。这些系统在推荐系统中整合多模态上下文，展现出显著的效果。
- 以往方法，如SGL、NCL和HCCF，在对比NGCF和LightGCN时，仅实现了轻微性能提升。我们假设，这是因为它们在生成自我监督信号时忽视了多模态上下文信息。相比之下，我们提出的方法利用多模态信息，包括从多模态图扩散模型生成的用户-项目图中提取的模态感知对比视角和对比增强。这允许我们捕获模态感知的自我监督信号，这些信号能有效补充多模态推荐系统中的监督任务。我们的多模态图扩散模型能生成模态感知的用户-项目图，进而产生模态感知对比视角和对比增强。通过这些，我们能获取模态感知的自我监督信号，显著增强多模态推荐系统的性能。
- Multi-Modal Graph Diffusion 的有效性。尽管MMGCL和SLMRec等多模态方法通过利用模态信息提升对比学习效果并执行数据增强，但它们存在局限性。例如，直接屏蔽模态特征可能导致关键信息的损失，而SLMRec基于预定义层次相关性生成增强视图，可能削弱自我监督信号在多模态推荐数据集中的有效性。相比之下，DiffMM-项目图，并利用跨模态对比学习实现有效的多模态增强，具有独特优势。

原文《DiffMM: Multi-Modal Diffusion Model for Recommendation》

编辑于 2024-08-01 14:25 · IP 属地北京

推荐系统

理性发言，友善互动

1 条评论 默认 最新

徐尘往

把vae-cf直接套壳到扩散模型上，感觉多模态这部分像是为了过审强行加的没看出来多模这部分和扩散模型有啥关系啊

08-02 · 北京 回复 喜欢

推荐阅读

听课笔记：腾讯多模态预训练技术的探索和实践

什么是预训练？在大规模无监督或者半监督任务上让模型学习尽可能多的通用先验知识，之后在下游任务中进行微调，实现知识迁移。 进行预训练的原因：可以加快模型收敛，降低下游任务对数据量...

残血的三井... 发表于小魏的论文...

5种网络IO模型（有图，很清楚）

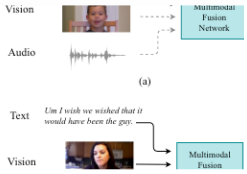
复制粘贴自 5种网络IO模型（有图，很清楚） - findumars - 博客园 同步（synchronous） IO和异步（asynchronous） IO，阻塞（blocking） IO和非阻塞（non-blocking）IO分别是什么，到底...

二毛儿

多模态融合技术升级！新阶段2大融合模式取得最优性能

传统的多模态融合方法面临着模态表示不一致、灵活性不足等问题，难以适应日益复杂的实际需求。 而随着大模型等新技术的发展，研究者将这些新技术与传统的多模态融合相结合，提出了新阶段的...

鱼子酱 发表于学姐带你读...



动态多模态融合

多模态机器... 发表于多模