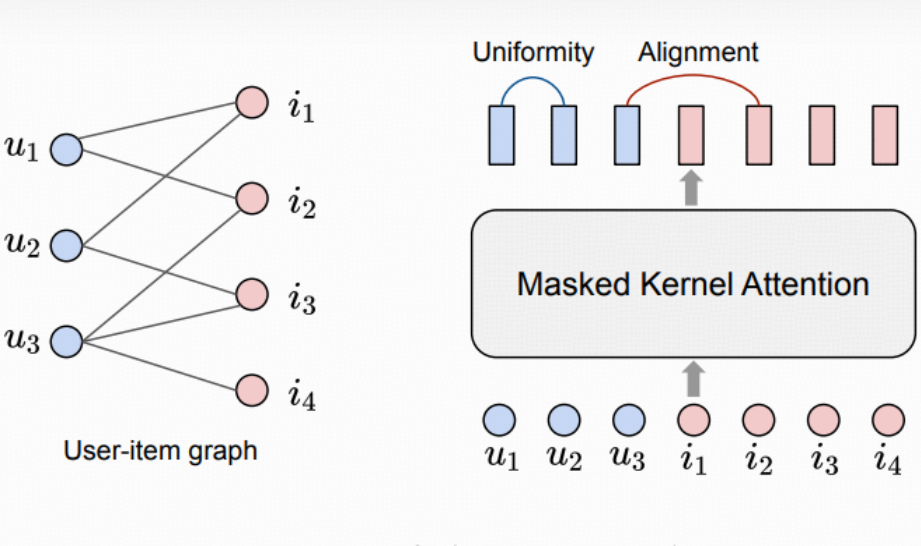


2024 Visa：图Transformer在工业级推荐系统中的应用

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

图神经网络（GNNs）在推荐系统中因强大表现而常用，但其局限性在于难以处理长序列依赖，导致表达力受限。深度GNN通过扩大视野来应对复杂模式，但可能导致过拟合和计算复杂度增加。相比之下，图Transformer（GTs）凭借自注意力机制在长序列依赖捕捉上表现出色，克服了GNN的局部限制。GT通过全局的注意力传递和创新的位置编码策略，如拉普拉斯、中心性、空间编码和随机游走结构编码，显著提升了性能。在多项测试中，GT超越了传统的GNN。尽管图Transformer在处理长序列依赖上有优势，但它面临的主要问题是空间和时间复杂度随节点数量呈二次方增长，这在工业推荐中是个突出难题。在NLP中，针对处理长序列的效率优化如稀疏注意力、局部敏感哈希和低秩近似已有所应用，但这些方法在面对百万级别的节点（或称令牌）仍需有效解决。

METHODOLOGY

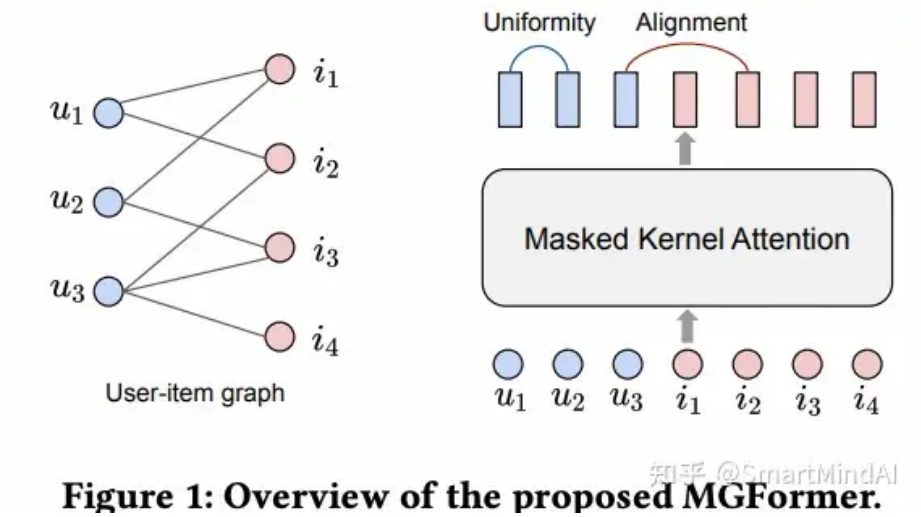


Figure 1: Overview of the proposed MGFormer.

Problem Setup

在给定用户-内容交互矩阵 $\mathbf{R} \in \mathbb{R}^{M \times N}$

其中M代表用户数N代表内容数的情况下，我们的任务是构建一个推荐系统。

一个模型，为每个用户生成一个有限数量的未观察过的内容列表，这些列表按与用户相关性由高到低进行排序。这通常涉及协同过滤✦或者基于内容的推荐算法，旨在利用已有的用户行为数据来推测潜在的兴趣和推荐。

Self-Attention Mechanism

在这个上下文中$\mathbf{h}_i$是通过注意力机制✦计算的，其中： -$\mathbf{Q}$ $\mathbf{K}$和$\mathbf{V}$是查询（query）、键（key）和值（value）向量，它们通常来自用户和内容嵌入的表示。 -$\mathbf{Attention}$是注意力函数，它接收这三个输入并进行计算。 -d是维度参数，用于缩放点积✦以避免数值不稳定。

公式中，softmax函数将查询和键的点积转换为概率分布，使得每个位置的权重反映了与该位置关联的重要性。然后，这个加权和$\mathbf{h}_i$作为内容i的相关度估计，用于决定哪些内容应出现在用户推荐列表的顶部。这种方法常用于推荐系统中，帮助系统理解用户当前的兴趣并据此进行个性化推荐。

$$\mathbf{h}_i = \frac{\sum_{j=1}^{n} \mathrm{sim}(\mathbf{q}_i, \mathbf{k}_j) \cdot \mathbf{v}_j}{\sum_{j=1}^{n} \mathrm{sim}(\mathbf{q}_i, \mathbf{k}_j)}, \quad \text{where} \quad \mathrm{sim}(\mathbf{q}_i, \mathbf{k}_j) = \exp\left(\frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d}}\right),$$

其中 $\mathbf{sim}(\mathbf{q}_i, \mathbf{k}_j)$ 表示计算的是查询向量$\mathbf{q}_i$与键向量$\mathbf{k}_j$的相似度，这是注意力机制的基础。这种相似性用来确定每个位置的注意力权重，权重由$\mathbf{q}_i$和$\mathbf{k}_j$的点积除以$\sqrt{d}$得出。

然而，这个过程的时间复杂度和内存消耗是$\mathcal{O}(n^2)$，其中n是序列长度，对于长序列处理非常耗时，限制了Transformer在大数据任务中的应用。为应对这个问题，研究者引入了优化策略，如局部注意力（Local Attention）或注意力头的分片（Sliced Attention），以减少空间复杂度✦，提高处理效率。

The Proposed MGFormer

Embedding Lookup

$$\mathbf{e}_u = \mathbf{E}_U[u] \quad \text{and} \quad \mathbf{e}_i = \mathbf{E}_I[i]$$

其中$\mathbf{E}_U$和$\mathbf{E}_I$是用户和物品的专属嵌入矩阵u和i是用户和物品的标识符$[\cdot]$表示索引。$\mathbf{e}_u$和$\mathbf{e}_i$就是用户和物品的嵌入向量✦，它们承载了用户和内容的信息，是后续处理的基础。

在MGFormer中，用户和物品的嵌入向量 $\mathbf{e}_u \in \mathbb{R}^d$ 和 $\mathbf{e}_i \in \mathbb{R}^d$

分别对应于用户u和物品i，它们具有相同维度。将所有节点的嵌入组合成一个矩阵$\mathbf{E}$，其维度为$(M+N) \times d$，其中M和N是用户和内容数量。这样做是为了将用户的特性与物品信息融合，以便在后续的图形神经网络（GNN）操作中处理和分析。这种合并有助于提供更全面的上下文信息，支持更精准的推荐或理解。

Structural Encodings

在利用图结构信息融入Transformer时，这是提升推荐系统性能的关键步骤。我们采用用户-物品交互矩阵的SVD（奇异值分解✦）来生成用户和物品的结构编码。首先，对矩阵进行SVD分解，得到低维的用户和物品向量，它们捕获了用户和物品间潜在的关联，如相似性或相关性。接着，我们将这些结构化的信息附加到用户的原始嵌入或物品的原始嵌入上，作为额外输入传递给Transformer。这样做增强了模型对社交网络中复杂关系的解析，从而显著提升了推荐的精确度。

在MGFormer中，我们使用SVD对用户-物品交互矩阵$\mathbf{R}$进行重构，得到用户结构编码 $\hat{\mathbf{U}}\sqrt{\mathbf{\Sigma}}$（维度$M \times d$）和物品结构编码 $\hat{\mathbf{V}}\sqrt{\mathbf{\Sigma}}$（维度$N \times d$）。这些结构编码保留了用户和物品间的关系信息。接着，将两者合并成一个$(M+N) \times d$的向量集$\mathbf{P}$，它包含全局图结构的特征。

随后，我们将用户的原始嵌入$\mathbf{E}$与结构嵌入$\mathbf{P}$组合，形成输入数据给图Transformer模型。这样做的目的是结合用户的静态属性和图的动态信息，显著提升推荐系统的准确性和效率。

Masked Kernel Attention

计算方式。计算公式为：

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K$$

然后，我们用这两个向量计算值向量$\mathbf{V}$：$\mathbf{V} = \mathbf{X}\mathbf{W}_V$

这些计算结果随后被用于注意力机制，通过比较查询和键，为序列中的不同位置赋予不同的权重，从而在处理序列任务时考虑了位置相关性。

受此思路启发，我们可以使用泛化的内积核 $\boldsymbol{\kappa}(\mathbf{a}, \mathbf{b})$ 替代传统相似度函数$\mathbf{sim}(\cdot, \cdot)$，用随机特征$\boldsymbol{\phi}(\cdot)$来近似，表达为 $\boldsymbol{\kappa}(\mathbf{a}, \mathbf{b}) = \boldsymbol{\phi}(\mathbf{a})^\top \boldsymbol{\phi}(\mathbf{b})$

这样一来，注意力机制的公式调整为：

$$\mathbf{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \boldsymbol{softmax}(\frac{\boldsymbol{\phi}(\mathbf{Q})\boldsymbol{\phi}(\mathbf{K})^\top}{\sqrt{m}})\boldsymbol{\phi}(\mathbf{V})$$

这种方法通过内积的随机特征表示，既能保持信息的传递，又能降低计算复杂度，特别适用于处理大规模图，因为它避免了直接点积操作，减少了空间和时间消耗。

$$\mathbf{h}_i = \frac{\sum_{j=1}^n \boldsymbol{\phi}(\mathbf{q}_i)^\top \boldsymbol{\phi}(\mathbf{k}_j) \cdot \mathbf{v}_j}{\sum_{j=1}^n \boldsymbol{\phi}(\mathbf{q}_i)^\top \boldsymbol{\phi}(\mathbf{k}_j)} = \frac{\boldsymbol{\phi}(\mathbf{q}_i)^\top \cdot \sum_{j=1}^n \boldsymbol{\phi}(\mathbf{k}_j) \mathbf{v}_j^\top}{\boldsymbol{\phi}(\mathbf{q}_i)^\top \sum_{j=1}^n \boldsymbol{\phi}(\mathbf{k}_j)}.$$

在特征映射方面，我们采用了Simplex随机特征（SimRFs），这是一种新颖的方法。SimRFs通过定义为：

$$\boldsymbol{\phi}(\mathbf{x}) = \frac{1}{\sqrt{m}}(\sin(\omega_1 x_1), \ldots, \sin(\omega_m x_m), \cos(\omega_1 x_1), \ldots, \cos(\omega_m x_m))$$

其中 $\mathbf{x} \in \mathbb{R}^m$ ω_i是来自高斯分布[+]的随机参数，满足$\sum_i \omega_i^2 = m$。SimRFs通过将输入映射到m维的正交子空间，具备线性时间复杂度，同时保留了与Softmax注意力相近的计算性质。SimRFs的选择不仅考虑了非线性，如ReLU和ELU，还可以用RFFs、ORFs或PRFs来近似Softmax注意力。这种选择旨在平衡计算效率和信息处理能力，使得在处理大规模图时，既能够高效处理，又能够有效地利用图的结构信息。

$$\boldsymbol{\phi}(\mathbf{a}) \stackrel{\text{def}}{=} \sqrt{\frac{1}{m}} \exp(\frac{-\|\mathbf{a}\|_2^2}{2})[\exp(\mathbf{w}_1^\top \mathbf{a}), \cdots, \exp(\mathbf{w}_m^\top \mathbf{a})]$$

在这个场景中，$\mathbf{W}$是一个$m \times m$的随机矩阵，由m个m维的权重向量 $\mathbf{w}_1, \ldots, \mathbf{w}_m$ 构成，每个向量通过独立来自高斯分布的正实数 $\omega_1, \omega_2, \ldots, \omega_m$ 相乘形成。SimRFs的设计目的是为了线性时间内完成特征映射，同时确保其与Softmax注意力的相似功能，这对于处理大数据集时既保证效率又能有效利用结构信息至关重要。

$\mathbf{W} = \mathbf{DSR}_o$ 这个过程通过标准化（$\sum_{j=1}^m s_{ij}^2 = 1$），确保每行的权重向量单位范数，即单位归一化，这样做的目的是在进行特征映射时，新生成的特征项具有单位长度，便于后续计算和比较，有利于特征的重要性度量。这样处理有助于保持信息的不变性并简化处理流程。

$$\boldsymbol{s}_i = \begin{cases} \sqrt{\frac{m}{m-1}}\mathbf{u}_i - \frac{\sqrt{m}+1}{(m-1)^{3/2}}(1, \ldots, 1, 0)^\top & \text{for } 1 \leq i < m, \\ \frac{1}{\sqrt{m-1}}(1, 1, \ldots, 1, 0)^\top & \text{for } i = m, \end{cases}$$

在图数据中，图的拓扑结构[+]是重要的先验信息；它不仅能作为节点位置的编码，还能强化注意力机制。比如，通过邻接矩阵[+]或最短路径矩阵，我们可以对注意力图进行加权或屏蔽处理。我们的核注意力模型，如前所述，能进一步利用这种结构，引入一个可学习的、基于拓扑的掩模$\mathbf{M}$来调节注意力得分的分布。具体公式如下：

$$\mathbf{Attention}_{\mathbf{masked}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{M}) = \mathbf{softmax}(\mathbf{Q}\mathbf{K}^\top + \mathbf{M}) \odot (\mathbf{V}\mathbf{M}^\top)$$

这里的屏蔽mask通过调整注意力权重，确保模型能更精准地聚焦于图结构信息，有效利用并提升对图数据的理解。

$$\mathbf{h}_i = \frac{\sum_{j=1}^n \mathbf{M}_{ij} \boldsymbol{\phi}(\mathbf{q}_i)^\top \boldsymbol{\phi}(\mathbf{k}_j) \cdot \mathbf{v}_j}{\sum_{j=1}^n \mathbf{M}_{ij} \boldsymbol{\phi}(\mathbf{q}_i)^\top \boldsymbol{\phi}(\mathbf{k}_j)} = \frac{\boldsymbol{\phi}(\mathbf{q}_i)^\top \cdot \sum_{j=1}^n \mathbf{M}_{ij} \boldsymbol{\phi}(\mathbf{k}_j) \mathbf{v}_j^\top}{\boldsymbol{\phi}(\mathbf{q}_i)^\top \sum_{j=1}^n \mathbf{M}_{ij} \boldsymbol{\phi}(\mathbf{k}_j)},$$

图结构相关区域，这有效地利用了图的先验知识，提升了模型对图数据的理解和处理能力。$\mathbf{P_2}$的计算方式与此类似，只是不包含值向量$\mathbf{v}_j$。这两个矩阵的生成，都是为了在保持注意力机制的同时，更加精准地聚焦于图结构信息。

$$\mathbf{P}_1 = \begin{pmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} & \cdots & \mathbf{M}_{1n} \\ \mathbf{M}_{21} & \mathbf{M}_{22} & \cdots & \mathbf{M}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{M}_{n1} & \mathbf{M}_{n2} & \cdots & \mathbf{M}_{nn} \end{pmatrix} \begin{pmatrix} \text{vec}(\phi(\mathbf{k}_1)\mathbf{v}_1^\top) \\ \text{vec}(\phi(\mathbf{k}_2)\mathbf{v}_2^\top) \\ \vdots \\ \text{vec}(\phi(\mathbf{k}_n)\mathbf{v}_n^\top) \end{pmatrix},$$

在处理中 $\text{vec}(\cdot)$ 代表向量化操作。当使用预先计算的密集型掩模$\mathbf{M}$计算$\mathbf{P_1}$和$\mathbf{P_2}$时，由于涉及到两次内积运算，时间复杂度会是线性的，这会消耗掉核注意力原本可能带来的优势。

在自然语言处理中，相对位置编码（Relative Positional Encoding）因其能体现位置间关系而表现优秀。因此，我们引入了一种基于相对余弦角度的中心性矩阵，作为掩模。这种掩模通过计算节点间相对距离的余弦值，来相对量化节点的重要性，表达节点在图结构中的关联性。公式为：

$$\textbf{Relative Degree Centrality Mask}(\mathbf{K}) = \sin\left(\frac{(i-j)\pi}{2d+2}\right)_{i,j=1}^{n,n}$$

这里的$\mathbf{K}$是特征矩阵i和j是节点索引d是节点最大相对距离。通过这种方式，我们在保持计算效率的同时，更准确地反映了节点在图中的中心性和影响力。

$$\mathbf{M}_{ij} = \sin\left(\frac{\pi}{2} \times \frac{z_{\deg(i)} + z_{\deg(j)}}{2}\right)$$

在这个表达式中，$\mathbf{M}_{ij}$定义为节点度$\deg(i)$与$\mathbf{M}'_{ij}$与相对度中心性矩阵

$$\textbf{Relative Degree Centrality Mask}(\mathbf{K})$$

的乘积经过ReLU激活的结果。ReLU函数用于增加非线性$\mathbf{M}'_{ij}$是系数。$\phi(\cdot)$是特征映射$\mathbf{q}_i$和$\mathbf{k}_j$分别是查询和键向量。 通过引入节点度的信息，这个修正的注意力计算方式使得模型能够更精准地集中在那些在图结构中有较高影响力的节点上，增强了模型对图数据的理解和利用。

$$\begin{aligned}
&\mathbf{M}_{ij}\phi(\mathbf{q}_i)^\top\phi(\mathbf{k}_j) = \sin(\frac{\pi}{2} \times \frac{z_{\deg(i)} + z_{\deg(j)}}{2})\phi(\mathbf{q}_i)^\top\phi(\mathbf{k}_j) \\
=&\left(\sin(\frac{\pi z_{\deg(i)}}{4})\cos(\frac{\pi z_{\deg(j)}}{4}) + \cos(\frac{\pi z_{\deg(i)}}{4})\sin(\frac{\pi z_{\deg(j)}}{4})\right)\phi(\mathbf{q}_i)^\top\phi(\mathbf{k}_j) \\
=&\left(\phi(\mathbf{q}_i)\sin(\frac{\pi z_{\deg(i)}}{4})\right)^\top\left(\phi(\mathbf{k}_j)\cos(\frac{\pi z_{\deg(j)}}{4})\right) + \\
&\left(\phi(\mathbf{q}_i)\cos(\frac{\pi z_{\deg(i)}}{4})\right)^\top\left(\phi(\mathbf{k}_j)\sin(\frac{\pi z_{\deg(j)}}{4})\right) \\
=&\phi^{\sin}(\mathbf{q}_i)^\top\phi^{\cos}(\mathbf{k}_j) + \phi^{\cos}(\mathbf{q}_i)^\top\phi^{\sin}(\mathbf{k}_j),
\end{aligned}$$

在当前情境中，我们新增了四个与节点度相关的特征映射：$\phi^{\sin}(\mathbf{q}_i)$ $\phi^{\cos}(\mathbf{k}_j)$ $\phi^{\cos}(\mathbf{q}_i)$和$\phi^{\sin}(\mathbf{k}_j)$，它们分别基于节点度$z_{\deg(i)}$应用正弦和余弦函数。这四个新的映射将原始的查询和键向量与节点度关联起来，以便更充分地利用节点的影响力。 原来的注意力计算公式可以更新为：

$$\begin{aligned}
&\textbf{Attention}_{\text{modified}} \\
&= \text{softmax}\left(\sum_{j=1}^{n}\left(\mathbf{M}_{ij}\cdot(\phi^{\sin}(\mathbf{q}_i)^\top\phi^{\cos}(\mathbf{k}_j) + \phi^{\cos}(\mathbf{q}_i)^\top\phi^{\sin}(\mathbf{k}_j))\right)\right) \odot (\mathbf{V}\mathbf{M}^\top)
\end{aligned}$$

这种修改后的注意力机制增强了对图结构中重要节点的关注，使得模型能更好地利用结构信息来处理图数据。

$$\begin{aligned}
\mathbf{h}_i =& \frac{\sum_{j=1}^{n}\mathbf{M}_{ij}\phi(\mathbf{q}_i)^\top\phi(\mathbf{k}_j)\cdot\mathbf{v}_j}{\sum_{j=1}^{n}\mathbf{M}_{ij}\phi(\mathbf{q}_i)^\top\phi(\mathbf{k}_j)} \\
=& \frac{\sum_{j=1}^{n}\phi^{\sin}(\mathbf{q}_i)^\top\phi^{\cos}(\mathbf{k}_j)\cdot\mathbf{v}_j + \sum_{j=1}^{n}\phi^{\cos}(\mathbf{q}_i)^\top\phi^{\sin}(\mathbf{k}_j)\cdot\mathbf{v}_j}{\sum_{j=1}^{n}\phi^{\sin}(\mathbf{q}_i)^\top\phi^{\cos}(\mathbf{k}_j) + \sum_{j=1}^{n}\phi^{\cos}(\mathbf{q}_i)^\top\phi^{\sin}(\mathbf{k}_j)} \\
=& \frac{\phi^{\sin}(\mathbf{q}_i)^\top\cdot\sum_{j=1}^{n}\phi^{\cos}(\mathbf{k}_j)\mathbf{v}_j^\top + \phi^{\cos}(\mathbf{q}_i)^\top\cdot\sum_{j=1}^{n}\phi^{\sin}(\mathbf{k}_j)\mathbf{v}_j^\top}{\phi^{\sin}(\mathbf{q}_i)^\top\sum_{j=1}^{n}\phi^{\cos}(\mathbf{k}_j) + \phi^{\cos}(\mathbf{q}_i)^\top\sum_{j=1}^{n}\phi^{\sin}(\mathbf{k}_j)}.
\end{aligned}$$

中的重要节点。这种设计优化了信息处理过程，增强了对结构信息的利用。

Optimization

为了确保对齐性和均匀性，我们在模型训练中采用了DirectAU损失。具体来说，这个损失函数用于优化模型参数，确保模型在处理数据时能够遵循这些特性。对齐性意味着模型输出与输入有明确的对应关系，而均匀性则保证了不同位置或特征的权重分布均匀，没有明显的偏重。通过优化这个损失，我们可以促使模型在学习过程中自动平衡这两点，提升其在处理复杂网络数据时的性能。

在直接AU损失的指导下λ是一个正则化参数，它控制着l_2范数，用来防止过拟合。p_{pos}是正向用户-内容配对的概率分布p_{user}和p_{item}分别代表用户和内容的真实分布，目标是让实际连接的节点（即用户-内容配对）相互靠近，而非随机用户或内容的节点均匀分布在以高维球面为基础的空间中，以体现对齐性。

特别指出，MGFormer设计的一个优点是，即使在仅使用单层单头注意力的情况下，也能高效学习到所有交互的复杂表达，相较于依赖于图形神经网络（GNN）的方法，它能更好地捕捉长序列依赖关系，展示了更强的表示能力。

EXPERIMENTS

Experimental Settings

在实验部分，我们使用三个基准数据集：**亚马逊**、**Yelp2018**和**阿里巴巴**[+]。

Table 1: Statistics of three benchmark datasets.

Dataset	#user	#item	#inter.	density
Beauty	22.4k	12.1k	198.5k	0.07%
Yelp	31.7k	38.0k	1561.4k	0.13%
Alibaba	106.0k	53.6k	907.5k	0.016%

我们衡量性能时采用标准的Top-k指标，包括Recall@k和NDCG@k（具体为$k = 20$），并采用全排名协议来评估。

Experimental Results

Method	Beauty		Yelp		Alibaba	
	recall	ndcg	recall	ndcg	recall	ndcg
BPRMF [30]	0.1153	0.0534	0.0693	0.0428	0.0439	0.0190
LightGCN [11]	0.1201	0.0581	0.0833	0.0514	0.0585	0.0275
SGL [40]	0.1228	0.0644	0.0896	0.0554	0.0602	0.0295
GOTNet [5]	0.1309	0.0650	0.0924	0.0567	0.0643	0.0301
SimGCL [49]	0.1367	0.0682	0.0937	0.0571	0.0667	0.0311
GFormer [21]	0.1362	0.0671	0.0955	0.0597	0.0653	0.0305
DirectAU [34]	0.1465	0.0710	0.0981	0.0615	0.0664	0.0308
MGFormer	**0.1531**	**0.0748**	**0.1051**	**0.0668**	**0.0702**	**0.0328**
Improv.	+4.50%	+5.35%	+7.14%	+8.62%	+5.25%	+5.47%

表中展示了各模型在Recall\@20和NDCG\@20指标下的表现。结果显示，我们的MGFormer显著优于LightGCN及其变型，性能提升幅度达到8.62%。这证实了单层核注意力模型的有效性，它在链接预测任务中表现出色。

知乎

8.7秒。尽管MGFormer在计算查询和键的权重以及随机特征变换上增加了额外成本，但总体来看，它在保持高性能的同时，展现出对复杂性管理的优势。尤其是在大规模推荐任务中，它超越了所有基线，维持了良好的效率。

原文《Masked Graph Transformer for Large-Scale Recommendation》

发布于 2024-05-29 10:49 · IP 属地北京

Visa 推荐系统 Transformer

推荐阅读

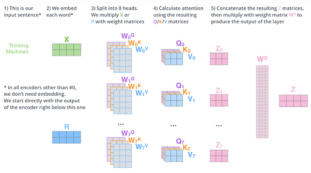
AIGC大模型八股整理（3）：Transformer中的前馈层和...

最近在学习准备大模型的相关知识，接触了快一年多了，还有很多基础感觉还是很薄弱，一边学习一边整理吧，欢迎指正 一、前馈层 FFNTransformer模型的前馈层对于每个位置的词向量独立地进行操...

IT图书馆

学习大模型的第一步：理解Transformer && 动手实现...

Alan 小分享



轻松理解Transformer中的Q,K,V,O矩阵

生生

Transformer重读 Day0

摘要最好的RNN和CNN模型 Encoder-Decoder架构。最 型也都采用了Attention mechanism. 本文提出一个不 CNN和RNN就和融合注意力 名为Transformer的简洁架构

生态