

陆圣尧. 一种轻量化的图注意力网络推荐模型[J]. 智能计算机与应用, 2025, 15(9): 146–152. DOI: 10. 20169/j. issn. 2095–2163. 250923

一种轻量化的图注意力网络推荐模型

陆圣尧

(上海理工大学 光电信息与计算机工程学院, 上海 200093)

摘要: 目前, 图神经网络在推荐系统中展现出了强大的能力。然而, 许多模型都存在模块多余、邻域聚合效率低等问题。为此, 本文提出了一种轻量化的图注意力网络推荐模型 LGAT。与现有的图注意力网络不同, LGAT 去除了特征转换和非线性激活操作, 只保留了邻域加权聚合这个核心组件, 简化了模型结构。本文还提出了一种自适应的邻域采样策略, 可以根据图结构来动态地选择最合适的邻居节点, 从而提高邻域聚合的质量和计算效率。在 3 个公开数据集上进行的实验结果表明, LGAT 在推荐准确率上更具有优势, 证明了对图注意力网络推荐系统轻量化的有效性。

关键词: 推荐系统; 图注意力网络; 自适应邻域采样

中图分类号: TP391.3

文献标志码: A

文章编号: 2095–2163(2025)09–0146–07

A lightweight graph attention network model for recommendation

LU Shengyao

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Graph neural networks have shown powerful capabilities in recommendation systems. However, many models face questions such as redundant modules, low efficiency of neighborhood aggregation, etc. To address these problems, a lightweight graph attention network model for recommendation LGAT is proposed. Unlike graph attention networks, LGAT removes the feature transformation and nonlinear activation operations, and only retains the most core component of neighborhood weighted aggregation. This simplifies the model structure and improves the model performance. This paper also proposes an adaptive neighborhood sampling strategy, which can dynamically select the most suitable neighbor nodes according to the graph structure, thereby improving the quality of neighborhood aggregation and enhancing the computational efficiency. Experiments are conducted on three public datasets, and it is indicated that LGAT has more advantages in recommendation accuracy, which well demonstrates the effectiveness of lightweighting graph attention network for recommendation.

Key words: recommendation system; graph attention network; adaptive neighborhood sampling

0 引言

推荐系统是一种利用用户和物品的信息与两者间的交互来预测用户对物品的偏好或评分的技术, 在电子商务、广告、营销等领域有着广泛的应用。随着互联网数据的爆炸性增长, 推荐系统面临着数据稀疏性、冷启动问题和多样性等挑战^[1-2]。为了解决这些问题, 基于图神经网络的推荐系统受到了越来越多的关注, 并展开了一系列研究^[3-5]。

图神经网络(Graph Neural Networks, GNN)通过构建一个用户-物品交互图(User-Item Interaction Graph, UIG), 利用图中节点之间的连接关系来学习

节点的表示^[6], 进而进行推荐预测。GNN 背后的核心思想是学习如何使用神经网络迭代聚合来自邻域的特征信息^[7-8]。具体来说, GNN 每次聚合来自节点的单跳邻居的特征信息, 并通过堆叠多个这样的聚合操作, 信息可以在图的远处传播^[9]。GNN 可以有效地利用 UIG 中的高阶连接信息, 从而缓解数据稀疏性和冷启动问题。早期的 GNN 从矩阵分解(Matrix Factorization, MF)演变过来, 其基于频域方法, 使用图的拉普拉斯矩阵进行计算^[10-12]。这种方式会使得模型的计算量很大, 且还依赖于图的全局结构, 因此无法用于动态图。后来的一些模型使用了基于空域的方法, 只选取固定数量的邻居节点来

计算当前节点的嵌入,因此模型的计算量要小得多。基于图的归纳学习方法(GraphSAGE)^[13]就是经典的基于空域的推荐模型,目前有很多模型都使用了GraphSAGE的思想。上述的这些模型都属于图卷积网络(Graph Convolutional Neural Network, GCN),其缺点是没有考虑各个邻居节点的重要性。图注意力网络(Graph Attention Networks, GAT)^[14]将注意力机制引入到GNN中,赋予每个邻居节点不同的权重,增加了模型的表达能力。并且GAT也可以使用基于空域的方法进行归纳学习。但是,目前基于GAT的模型都会存在2个问题。

首先,很多模型没有做充分的消融研究,而某些组件在推荐系统中是多余的。轻量化的GCN推荐算法(LightGCN)^[15]简化了图卷积网络的设计,认为图卷积网络中的特征转换和非线性激活操作对推荐结果作用不大,只保留了邻域聚合这一最核心的组件,并通过实验证明了对GCN轻量化的有效性。但是LightGCN具有一般图卷积网络的一些缺点,即依赖图的全局结构,且没有考虑邻居节点的重要性。受到LightGCN的启发,本文对GAT的各个模块进行有效性分析。研究认为目前一些基于GAT的推荐模型设计比较繁重,GAT最初是用于节点分类任务的,每个节点具有初始输入特征,而在UIG中,每个节点只通过一个one-hot向量表示,没有具体语义。在这种情况下,GAT中的特征转换与非线性激活操作基本是没有作用的,反而会增加训练难度。

其次,由于UIG中的节点数量和边数量都很大,如果直接使用全图来进行图注意力网络的计算,会导致内存和计算开销过大。因此,就需要使用一种邻域采样策略来减少图的规模和复杂度。一种常用的方法是使用固定大小的邻域采样策略,即为每个节点随机地选择一个固定大小的邻域子集作为其邻域节点,如GraphSAGE^[13]。这样做也可以解决图神经网络无法用于动态图的问题,但是却会带来一些新的问题。由于使用随机采样,可能会丢失重要的信息,也可能会引入噪声。

在本文中,提出了一种基于GAT的轻量化推荐模型LGAT,只包括邻域加权聚合这一最核心的组件,并证明了去除多余的组件可以提升模型的性能。LGAT模型使用了一种自适应的邻域采样策略,可以提高邻域聚合的质量,提升计算效率。具体来说,论文做了以下贡献:

(1)提出了LGAT,去掉了传统GAT中的特征

转换和非线性激活操作,只保留了邻域加权聚合这一最核心的组件,即在用户-物品交互图中线性传播用户与物品的嵌入。

(2)提出了一种自适应邻域采样策略,可以根据图结构动态地选择最相关的邻域节点。具体来说,提出了一个采样重要性系数,每次采样过程中会根据当前该节点的注意力系数更新采样重要性系数,然后每次聚合时选择采样重要性系数最大的一部分邻居。

(3)在3个公开数据集上进行了实验,结果表明,LGAT模型在推荐性能上显著优于其他模型。同时还对LGAT模型进行了深入的分析,证明了特征转换和非线性激活操作对推荐效果没有帮助,反而会增加模型的参数数量和计算开销。并且还探讨了不同参数设置、层组合策略对模型性能的影响。

1 相关工作

在推荐系统中,GNN是一种新兴的技术,它可以有效地利用图结构的信息来学习用户和物品的表示^[6]。最早将GNN应用于推荐系统的工作之一是图卷积矩阵补全(Graph Convolutional Matrix Completion, GC-MC)^[10],将MF模型转化为GCN,使用一阶谱卷积运算来传播用户和物品的嵌入。后续的工作对GC-MC进行了改进和扩展,例如堆积重构图卷积网络(STAR-GCN)^[12]直接学习低维的用户和物品嵌入作为网络的输入,减少了模型的空间复杂度。解耦图卷积神经网络(Disentangled Graph Convolutional Networks, DGCF)^[16]分离了用户意图来获得细粒度的用户和物品的嵌入,提高了模型的可解释性。这些模型都是基于频域的图卷积网络的变体,依赖图的全局结构。神经图协同过滤(Neural Graph Collaborative Filtering, NGCF)^[17]使用多层图卷积网络来捕提高阶连接,以显式方式将协作信号有效地注入到嵌入过程中,通过聚合器为每个节点采样一个固定大小的邻域,这是GraphSage的变体,不再依赖图的全局结构。但是上述模型都没有考虑到每个节点各个邻居的重要性。

GAT^[14]是一种基于自注意力机制的图神经网络,可以学习节点的邻域信息,并为不同的邻居节点赋予不同的权重。多分量图卷积协同过滤(Multi-Component Graph Convolutional Collaborative Filtering, MCCF)^[18]是一种基于图注意力网络的推荐模型。MCCF模型首先将UIG中的边分解为多个

分量,每个分量代表一种潜在的购买动机。然后利用 GAT 来学习每个分量下的用户和物品的表示,最后将不同分量下的表示进行聚合来实现推荐预测。

He 等学者^[15]认为 GCN 在推荐系统中过于复杂,其中一些组件没有作用,这也启发了 LGAT 设计工作的开展。LightGCN 是基于 GCN 的,通过使用拉普拉斯矩阵进行计算,依赖图的全局结构,且未考虑邻居节点的重要性。相比之下, LGAT 采纳了 LightGCN 轻量化的特点,且使用注意力系数增加了模型的表达能力,并通过自适应邻域采样将模型用于推导式学习,使得 LGAT 更轻巧有效。

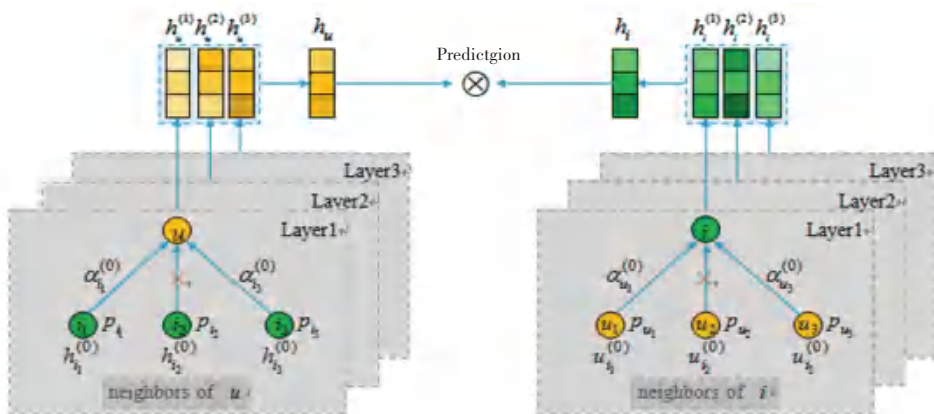


图1 LGAT 模型结构图

Fig. 1 Model structure of LGAT

2.1 GAT 的轻量化

GAT 可以为每个邻居节点分配一个权重向量,用于表示该邻域节点对中心节点的重要性。GAT 的定义公式如下:

$$h_i^{(l+1)} = \sigma \left(\sum_{j \in N_i} \alpha_{ij}^{(l)} \mathbf{W}^{(l)} h_j^{(l)} \right) \quad (1)$$

其中, $h_i^{(l)}$ 表示第 l 层中节点 i 的嵌入向量; N_i 表示节点 i 的邻域节点集合; $\alpha_{ij}^{(l)}$ 表示第 l 层中节点 i 对其邻域节点 j 的注意力权重; $\mathbf{W}^{(l)}$ 表示第 l 层中的特征转换矩阵; σ 表示非线性激活函数。

注意力权重 α_{ij} 是通过一个可学习的向量 \mathbf{a} 来计算的,可以捕捉 2 个节点之间的相似性或相关性。GAT 中定义了 \mathbf{a} 是一个单层前馈神经网络,由一个权重向量参数化,并且使用了非线性激活函数 LeakyReLU,对此可以表示为:

$$e_{ij}^{(l)} = \text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}^{(l)} h_i^{(l)} \parallel \mathbf{W}^{(l)} h_j^{(l)}]) \quad (2)$$

其中,注意力系数 e_{ij} 表示节点 j 对节点 i 的重要性,“ \parallel ”表示向量拼接操作。然后,为了使得每个节点的注意力权重之和为 1,需要对其注意力系数

2 方法

在本节中,介绍本文的模型 LGAT,模型结构如图 1 所示。LGAT 包括 3 个部分:首先对已初始化的用户和物品嵌入进行加权聚合,其中去掉了 GAT 中的特征转换与非线性激活操作,并且根据每个邻居节点的采样重要性系数有选择性地聚合。其次,通过多层堆叠得到每个节点的高阶嵌入,并组合每一层的嵌入作为最终嵌入。最后,使用内积操作来计算用户和物品之间的预测评分。

进行归一化处理,即:

$$\alpha_{ij}^{(l)} = \frac{\exp(e_{ij}^{(l)})}{\sum_{k \in N_i} \exp(e_{ik}^{(l)})} \quad (3)$$

可以看到,在学习注意力系数 e_{ij} 和聚合邻居节点时,都使用了一个共享的权重矩阵 \mathbf{W} 。但是这对于推荐系统来说没有什么作用。在节点分类中,每个节点都有丰富的语义特征作为输入^[19]。因此,进行非线性变换有利于特征学习。然而,在推荐系统中,UIG 的每个节点只有一个 ID 作为输入,没有具体的语义。在这种情况下,执行非线性变换将不会有助于学习更好的特征,反而会增加训练的难度。

将式(1)中的权重矩阵 \mathbf{W} 与激活函数 σ 去掉后,每一层的更新公式变为:

$$h_i^{(l+1)} = \sum_{j \in N_i} \alpha_{ij}^{(l)} h_j^{(l)} \quad (4)$$

其中,注意力权重 α_{ij} 仍然使用单层感知器来计算,并进行归一化处理。将式(2)中的特征转换矩阵 \mathbf{W} 去掉后,注意力系数 e_{ij} 的计算公式变为:

$$e_{ij}^{(l)} = \text{LeakyReLU}(\mathbf{a}^T [\mathbf{h}_i^{(l)} \parallel \mathbf{h}_j^{(l)}]) \quad (5)$$

但是,在式(2)计算注意力系数时,非线性激活函数 LeakyReLU 不能去掉。如果将其去掉,那么:

$$\exp(e_{ij}^{(l)}) = \exp(\mathbf{a}^T [\mathbf{W}^{(l)} \mathbf{h}_i^{(l)} \parallel \mathbf{W}^{(l)} \mathbf{h}_j^{(l)}]) \quad (6)$$

将向量 \mathbf{a} 拆分为 2 个向量,即:

$$\mathbf{a} = \mathbf{a}_1 \parallel \mathbf{a}_2 \quad (7)$$

由此推得:

$$\begin{aligned} \exp(e_{ij}^{(l)}) &= \exp([\mathbf{a}_1 \parallel \mathbf{a}_2]^T [\mathbf{W}^{(l)} \mathbf{h}_i^{(l)} \parallel \mathbf{W}^{(l)} \mathbf{h}_j^{(l)}]) = \\ &= \exp(\mathbf{a}_1^T \mathbf{W}^{(l)} \mathbf{h}_i^{(l)} + \mathbf{a}_2^T \mathbf{W}^{(l)} \mathbf{h}_j^{(l)}) = \\ &= \exp(\mathbf{a}_1^T \mathbf{W}^{(l)} \mathbf{h}_i^{(l)}) \exp(\mathbf{a}_2^T \mathbf{W}^{(l)} \mathbf{h}_j^{(l)}) \end{aligned} \quad (8)$$

如果没有 LeakyReLU,则式(3)就变为:

$$\alpha_{ij}^{(l)} = \frac{\exp(\mathbf{a}_1^T \mathbf{W}^{(l)} \mathbf{h}_i^{(l)}) \exp(\mathbf{a}_2^T \mathbf{W}^{(l)} \mathbf{h}_j^{(l)})}{\sum_{k \in N_i} \exp(\mathbf{a}_1^T \mathbf{W}^{(l)} \mathbf{h}_i^{(l)}) \exp(\mathbf{a}_2^T \mathbf{W}^{(l)} \mathbf{h}_k^{(l)})} = \frac{\exp(\mathbf{a}_2^T \mathbf{W}^{(l)} \mathbf{h}_j^{(l)})}{\sum_{k \in N_i} \exp(\mathbf{a}_2^T \mathbf{W}^{(l)} \mathbf{h}_k^{(l)})} \quad (9)$$

也就是说,分子分母同时约去了节点 i 的信息, α_{ij} 实际变成了 $\alpha_{j\circ}$ 。这时,节点对之间的注意力权重并没有考虑节点 i 的表示。如果加上激活函数 LeakyReLU,那么后面的归一化就不会约去节点 i 的信息。因此在计算注意力系数时保留激活函数。

2.2 自适应邻域采样

LGAT 模型为每个节点维护一个采样重要性系数 p_j , 用于表示节点 j 的重要性,重要性越高的节点被采样的概率也越高。初始时,每个节点的采样重要性系数为:

$$p_j^{(0)} = \frac{1}{|N_j|} \quad (10)$$

其中, $|N_j|$ 表示节点 j 的邻居数量。这样设置的目的是考虑到了推荐的公平性,因为当某个节点的邻居数量较少,则表示购买此物品的用户比较少或此用户购买的物品数量较少。对于冷门节点,初始应该赋予其更大的采样概率。

在每次训练迭代中, LGAT 模型根据当前的注意力权重 α_{ij} 来更新采样重要性系数,使得注意力权重较高的邻域节点被采样的概率也较高。具体的更新公式为:

$$p_j^{(l+1)} = p_j^{(l)} + \eta \alpha_{ij} \quad (11)$$

其中, η 是一个超参数,用于控制采样重要性系数的变化速度。通过这种方式, LGAT 模型可以自适应地选择合适的邻域进行采样,从而提高模型的泛化能力。

对于节点 i , 从该节点的邻居节点集合 N_i 中选

取采样重要性系数最大的 K 个节点,对此可以表示为:

$$N_i^K = \text{top-rank}\{p_j \text{ for } j \in N_i, K\} \quad (12)$$

其中, top-rank 表示返回采样重要性系数最大的 K 个邻居节点的函数, N_i^K 表示从节点 i 的邻居节点中选取采样重要性系数最大的 K 个节点组成的集合。

最终, LGAT 每一层的更新公式为:

$$\mathbf{h}_i^{(l+1)} = \sum_{j \in N_i^K} \alpha_{ij}^{(l)} \mathbf{h}_j^{(l)} \quad (13)$$

由于 UIG 是一个动态变化的图,用户和物品之间会不断产生新的交互记录。因此,在每次训练迭代中,需要更新 UIG 的结构和权重,并重新计算注意力权重和采样重要性系数。这样可以使得模型能够及时地捕捉用户和物品之间的最新动态,并提高推荐效果。

2.3 层组合与模型预测

在 LGAT 中,对每个节点的嵌入初始化后,通过式(13)可以计算出更高层的嵌入。在经过 L 层迭代聚合之后,将用户和物品节点在每层计算得到的嵌入进行组合,以形成最终表示:

$$\mathbf{h}_i = \sum_{l=0}^L \beta_l \mathbf{h}_i^{(l)} \quad (14)$$

其中, $\beta_l \geq 0$ 表示第 l 层嵌入在节点最终嵌入中的重要性,这是一个需要手动调整的超参数。通过研究发现将 β_l 按指数递减的方式分配(即高阶邻居对中心节点的影响按指数减少)与均匀分配相比,模型的性能差距不大。因此,在本次研究中无需优化 β_l , 而是简单地将 β_l 全部设置为 $\frac{1}{L+1}$, 这样也可以保持 LGAT 轻量化的特点。

采用层组合策略来获得每个节点最终的表示更具有优势。首先,随着层数的增加,邻近的用户与物品的嵌入将变得越来越相似,即过度平滑^[20]。因此,只是简单地使用最后一层是有问题的。其次,不同层的嵌入捕获的语义是不同的。例如,第一层的用户节点捕获其购买过的物品的特征,物品节点捕获购买它的用户的特征,第二层的用户节点捕获与其购买过相似物品的用户的特征,物品节点捕获被相似用户购买过的物品的特征,更高层捕获更高阶的邻居特征^[17]。因此,将前述特征结合起来可使得每个节点的嵌入表达更加全面。

研究中将使用用户和物品最终嵌入的内积作为预测评分:

$$\hat{r}_{ui} = \mathbf{h}_u^T \mathbf{h}_i \tag{15}$$

2.4 模型训练

在 LGAT 中,可训练的模型参数是第 0 层的嵌入与注意力系数。由于任务是评分预测,因此模型主要目标是最大限度地减少预测评分与真实值之间的差异,可由如下公式计算求得:

$$\mathcal{L}_r = \frac{1}{2|O|} \sum_{(u,i) \in O} (\hat{r}_{ui} - r_{ui})^2 \tag{16}$$

其中, O 表示已有评分的集合, r_{ui} 表示用户 u 对物品 i 的真实评分。

研究使用 L2 正则化来防止过拟合,并使用 Adam 优化器^[21],最终的损失函数如下:

$$\min_{\Theta} \mathcal{L} = \mathcal{L}_r + \lambda \|\Theta\|^2 \tag{17}$$

其中, Θ 表示模型参数集,即第 0 层的嵌入 $\Theta = \{\mathbf{H}^{(0)}\}$, λ 用于控制正则化强度。

3 实验

在本节中,将介绍论文的实验设置和结果分析。首先,介绍实验数据集、评价指标和对比方法;然后,展示实验结果,分析 LGAT 模型中各个组件的有效性;接着,进行消融研究,分析不同层组合策略对推荐结果的影响;最后,进行超参数研究,探讨不同采样重要性系数对模型性能的影响。

3.1 实验设置

论文中使用 3 个公开的真实数据集来进行实验,分别是 MovieLens、Amazon 和 Yelp。这些数据集涵盖了不同领域、不同规模 and 不同密度的用户-物品交互记录,其中 Amazon 是一个广泛使用的商品推荐数据集。表 1 给出了这些数据集的统计信息。

表 1 数据集统计信息
Table 1 Dataset statistics

Dataset	User	Item	Interaction	Density
MovieLens	943	1 682	100 000	0.063 0
Amazon	1 000	1 000	65 170	0.065 2
Yelp	1 286	2 614	30 838	0.009 2

研究中按照 8 : 2 的比例将每个数据集划分为训练集和测试集,并去除孤立节点。

接下来,将 LGAT 模型与其他几种方法进行对比,包括基于图卷积网络的协同过滤模型 STAR-GCN、LightGCN,使用聚合函数的图卷积网络模型 NGCF,基于图注意力网络的模型 MCCF。此外,还对 LGAT 进行了消融研究,采用了 LGAT 的 3 种变体来分析没有去掉特征转换或非线性激活所带来的

影响。

论文在 {8, 16, 32, 64, 128} 范围内寻找合适的节点嵌入维度,目前设置的最优值为 64。文中使用 Xavier 方法初始化节点^[22],然后使用 Adam 作为优化器。批量大小和学习率分别在 {64, 128, 256, 512} 和 {0.000 5, 0.001 0, 0.002 0, 0.002 5} 中寻找。 L_2 正则化系数 λ 为 $1e^{-4}$ 。并且将所有基线的嵌入维度都设置为 64,使得比较结果更具有说服力。实验中采用了 2 种广泛使用的评估指标:均方根误差(RMSE)和平均绝对误差(MAE)。RMSE 和 MAE 都可以衡量模型预测评分与真实评分之间的误差大小,其值越小,表示模型的推荐效果越好。

3.2 实验结果和分析

表 2 显示了 LGAT 与基线方法的性能比较,其中展示了每个模型的最佳结果。从表 2 中可以看出, LGAT 模型在所有数据集上都优于其他方法,其简单合理的设计证明了 LGAT 模型在推荐性能上的优势。在基线中,LightGCN 表现出最强的性能,体现出对图神经网络在推荐系统中适当的轻量化可以提升模型的性能。

研究中对 GAT 的各个模块进行了有效性分析,以探索非线性激活和特征转换的影响。主要实现了 LGAT 的 3 种变体:

- (1) LGAT-f。保留了 GAT 中的特征转换矩阵 W 。
- (2) LGAT-n。保留了 GAT 中的非线性激活函数 σ 。
- (3) LGAT-fn。保留了 GAT 中的特征转换矩阵和非线性激活函数。

对于这 3 种变体,保持所有超参数(例如,学习率、正则化系数、嵌入维度等)与 LGAT 的最佳设置相同。可以看出,在所有 3 个数据集上,与 LGAT-fn 相比,去除特征转换(即 LGAT-n)会提升模型的性能。而只去除非线性激活不会对精度产生太大影响。如果在去除特征转换的基础上去除非线性激活(即 LGAT),性能会得到明显提高。根据这些观察,得出的结论是:

- (1) 添加特征转换会对 LGAT 产生负面影响,因为在 LGAT 和 LGAT-n 中去掉特征变换都会明显提高模型性能。
- (2) 当包含特征转换时,只去除非线性激活会产生轻微的负面影响。
- (3) 总体而言,特征转换和非线性激活对推荐效果产生了比较大的负面影响,因为同时将其去掉,

LGAT 的最终结果比 LGAT-fn 有了比较大的改进。

将 LGAT-f 的特征转换矩阵设置为单位矩阵后蜕变为 LGAT,可以看出 LGAT-f 比 LGAT 具有更高

的表达能力。但是从实验结果中可以得出结论,复杂的模型会提升训练难度,反而会影响模型的推荐效果。

表 2 评分预测的性能比较

Table 2 Performance comparison of rating prediction

Model	MovieLens-100k		Amazon		Yelp	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
STAR-GCN	0.910 1	0.716 2	0.892 4	0.658 7	0.387 4	0.110 6
NGCF	0.908 9	0.712 6	0.886 3	0.649 4	0.382 6	0.105 2
LightGCN	0.902 7	0.701 0	0.879 3	0.640 4	0.381 4	0.101 8
MCCF	0.907 4	0.705 3	0.889 2	0.644 1	0.381 9	0.104 3
LGAT-f	0.912 5	0.709 0	0.896 7	0.650 2	0.384 6	0.103 9
LGAT-n	0.904 3	0.705 1	0.881 4	0.642 6	0.378 4	0.101 8
LGAT-fn	0.911 3	0.706 8	0.891 7	0.647 5	0.381 4	0.102 6
LGAT	0.901 7	0.697 4	0.873 5	0.640 2	0.372 6	0.100 5

3.3 消融研究

研究中对 LAGT 使用不同层组合方式的影响进行研究。图 2 显示了 LGAT 及其只使用每个节点最后一层嵌入作为最终特征的变体 LGAT-single 在不同层数时的表现。本文在 Amazon 数据集上进行实验。对于 LGAT-single,可以发现模型层数从 1 到 4 时,其性能先升高后降低。将模型层数设置为 2 时,其性能基本是最高的。这表明,当使用高阶邻居时,会出现过平滑问题。而对于 LGAT,分析发现其性能随着层数的增加而逐渐提高。即使模型设置为 4 层,其性能也不会突然降低。因此可以得出结论,不能简单地使用最后一层嵌入作为最终的节点特征,将每层的权重全部设置为 $\frac{1}{L+1}$ 可以解决过平滑的问题。

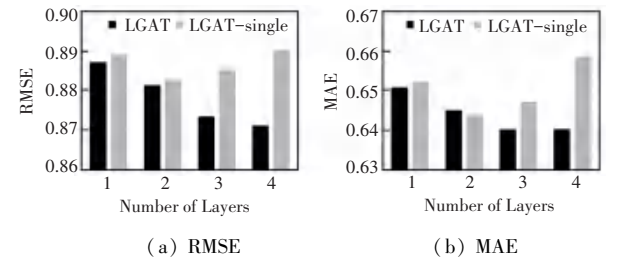


图 2 LGAT 与 LGAT-single 堆叠不同层数时的结果比较

Fig. 2 Comparison of results when LGAT and LGAT-single are stacked with different number of layers

3.4 超参数研究

在 LGAT 训练过程中,控制采样重要性系数变化速度的超参数 η 对模型的性能也会产生重要的影响。因此,对 η 进行研究。图 3 显示了在 Amazon 数

据集上设置不同的 η 对模型性能的影响。可以看到, η 从 1 至 $1e^{-3}$, 最终的误差先降低后升高。这表明,当 η 较大时,采样重要性系数增速较快,这样对于邻域的选择会聚焦于一些热门节点,导致每个节点失去了个性化的特征。而 η 设置得过小也会产生一些问题,如当 η 为 $1e^{-3}$ 时,其值接近于 0,这样式 (11) 可以改写为:

$$p_j^{(t+1)} = p_j^{(t)} \tag{18}$$

如此则表示采样重要性系数将不会有什么变化,所以对于邻域的选择会聚焦于初始值较大的节点,即冷门节点,这样的话各个节点之间将会孤立,联系较小,严重损害模型的表达能力。因此,研究中将 η 设置为 $1e^{-1}$, 其结果最佳。

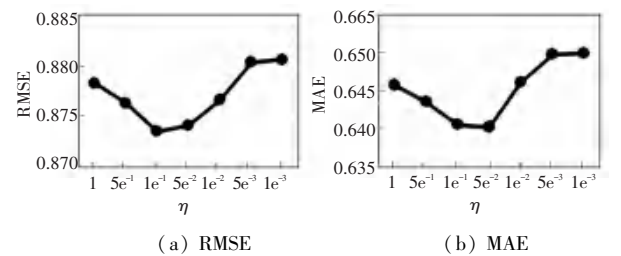


图 3 不同的 η 对模型性能的影响

Fig. 3 Effect of different η on model performance

4 结束语

本文研究了用于推荐系统的图注意力网络不必要的复杂设计,并进行了实证研究来证明这一论点。论文提出了 LGAT,其中去除了特征转换和非线性激活这 2 种图神经网络中的标准操作,否则就会增加训练难度。并且采用了一种自适应的邻域采样策

略,可以动态地为每个节点选择最合适的邻域节点,提高聚合质量,增加计算效率。在层组合中,研究将每个节点在所有层上嵌入的加权和作为最终嵌入,这样可以缓解过平滑的问题。通过实验证明了LGAT的优势:更容易训练,更好的泛化能力,更简单有效。今后的工作中,将致力于探索更优的自适应邻域采样策略,进一步提高聚合的质量。

参考文献

- [1] CHEN Hao, WANG Zefan, HUANG Feiran, et al. Generative adversarial framework for cold-start item recommendation [C]//Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2022: 2565–2571.
- [2] ZHAO Jun, ZHOU Zhou, GUAN Ziyu, et al. IntentGC: A scalable graph convolution framework fusing heterogeneous information for recommendation [C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2019: 2347–2357.
- [3] CHEN Hao, HUANG Zhong, XU Yue, et al. Neighbor enhanced graph convolutional networks for node classification and recommendation [J]. Knowledge-Based Systems, 2022, 246: 108594.
- [4] CHEN Hao, DENG Zengde, XU Yue, et al. Non-recursive graph convolutional networks [C]//Proceedings of 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2021). Piscataway, NJ: IEEE, 2021: 3205–3209.
- [5] CAI Xuheng, HUANG Chao, XIA Lianghao, et al. LightGCL: Simple yet effective graph contrastive learning for recommendation [J]. arXiv preprint arXiv, 2302.08191, 2023.
- [6] HAMILTON W L, YING R, LESKOVEC J. Representation learning on graphs: Methods and applications [J]. arXiv preprint arXiv, 1709.05584, 2017.
- [7] CHEN Jie, MA Tengfei, XIAO Cao. FastGCNn: Fast learning with graph convolutional networks via importance sampling [J]. arXiv preprint arXiv, 1801.10247, 2018.
- [8] YING R, HE Ruining, CHEN Kaifeng, et al. Graph convolutional neural networks for web-scale recommender systems [C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2018: 974–983.
- [9] XU Yue, CHEN Hao, DENG Zengde, et al. Flattened graph convolutional networks for recommendation [J]. arXiv preprint arXiv, 2210.07769, 2022.
- [10] BERG R, KIPF T N, WELLING M. Graph convolutional matrix completion [J]. arXiv preprint arXiv, 1706.02263, 2017.
- [11] ZHENG Lei, LU Chunta, JIANG Fei, et al. Spectral collaborative filtering [C]//Proceedings of the 12th ACM Conference on Recommender Systems. New York: ACM, 2018: 311–319.
- [12] ZHANG Jiani, SHI Xingjian, ZHAO Shenglin, et al. Star-GCN: Stacked and reconstructed graph convolutional networks for recommender systems [J]. arXiv preprint arXiv, 1905.13129, 2019.
- [13] HAMILTON W, YING Z, LESKOVEC J. Inductive representation learning on large graphs [C]//Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Long Beach, USA: NIPS Foundation, 2017: 1025–1035.
- [14] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks [J]. arXiv preprint arXiv, 1710.10903, 2017.
- [15] HE Xiangnan, DENG Kuan, WANG Xiang, et al. LightGCN: Simplifying and powering graph convolution network for recommendation [C]//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2020: 639–648.
- [16] WANG Xiang, JIN Hongye, ZHANG An, et al. Disentangled graph collaborative filtering [C]//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2020: 1001–1010.
- [17] WANG Xiang, HE Xiangnan, WANG Meng, et al. Neural graph collaborative filtering [C]//Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2019: 165–174.
- [18] WANG Xiao, WANG Ruijia, SHI Chuan, et al. Multi-component graph convolutional collaborative filtering [J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(4): 6267–6274.
- [19] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks [J]. arXiv preprint arXiv, 1609.02907, 2016.
- [20] LI Qimai, HAN Zhichao, WU Xiaoming. Deeper insights into graph convolutional networks for semi-supervised learning [J]. arXiv preprint arXiv, 1801.07606, 2018.
- [21] KINGMA D P, BA J. Adam: A method for stochastic optimization [J]. arXiv preprint arXiv, 1412.6980, 2014.
- [22] GLOROT X, BENGIO Y. Understanding the difficulty of training deep feedforward neural networks [C]//Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Sardinia, Italy: JMLR Workshop and Conference Proceedings, 2010: 249–256.