



分类号： TP391

学校代码： 11066

学 号： 202000358049

烟台大学

专 业 学 位 硕 士 论 文

基于图神经网络的特征交互推荐算法研究

**Research of Feature Interactive Recommendation
Algorithm Based on Graph Neural Network**

研 究 生 姓 名： 马昕欣

指 导 教 师： 崔振东 教授

专业学位类型： 电子信息

专业学位领域： 计算机技术

论文提交日期： 2023 年 6 月 10 日

摘要

随着信息技术的快速发展和大数据的爆发式增长，如何发现有效的目标数据具有重要意义，推荐系统能够为用户推荐个性偏好的数据信息，从而得到了越来越广泛的应用。基于特征交互的推荐算法是推荐系统的研究热点之一，近年来提出的特征交互推荐算法取得了不错的成果，但仍存在一些不足。主要表现为：一方面，传统的机器学习方法能通过暴力枚举实现低阶交互建模，但无法对高阶的特征组合进行有效捕捉；另一方面，基于深度学习的方法使用深度神经网络学习高阶特征交互是隐式的，对特征交互过程缺乏一定的解释性。针对这些问题，本研究基于图神经网络进行特征交互建模研究。主要内容如下：

（1）提出了一种注意力感知的图匹配特征交互推荐算法。该方法将推荐问题转变为图匹配问题，把传统点到点的建模范式扩展到了图到图。首先，为了捕捉不同级别的特征交互过程，该方法将用户和物品信息建模为两个特征图，并从内部交互和外部交互两个角度来建模特征交互。内部交互捕捉单个特征图内的特征交互，外部交互捕捉两个特征图之间的节点匹配。其次，在特征节点交互阶段和节点表示融合阶段都引入注意力机制方法捕捉交互的重要性。最后，在两个公开数据集上开展了实验研究，实验结果表明所提出的算法相较于对比算法具有良好的预测效果。

（2）提出了一种分层双级别图融合特征交互推荐算法。该方法针对注意力感知的图匹配特征交互没有引入上下文信息的不足，把推荐问题转变为图分类问题。首先，将每个特征域视为特征图中的节点，节点之间相连接的边视为特征之间的交互作用，通过分层图结构学习更新节点之间的连接情况，得到有意义的特征交互。其次，设计了一个双级别节点和图的表示生成模块，该模块包含两个级别的特征交互和融合过程，局部级别交互使用边权重来更新节点的表示，全局级别交互通过压缩激励动态捕捉各个特征域的重要程度。接下来，设计了双线性融合方法，从多个角度对两个级别交互的节点信息进行融合。最后，通过与多个基线模型进行实验对比，结果证明分层双级别图融合特征交互推荐算法具有更高的预测准确率。

关键词：推荐系统；图神经网络；特征交互；图匹配；图结构学习

Abstract

With the rapid development of information technology and the explosive growth of big data, how to discover effective target data is of great significance, and the recommendation system can recommend data information for users to match their personality preferences has been more and more widely used. The recommendation algorithm based on feature interaction is one of the research hotspots of recommendation systems, and the feature interaction recommendation algorithm proposed in recent years has achieved good results, but there are still some shortcomings. On the one hand, traditional machine learning methods can achieve low-order interaction modeling by violent enumeration, but cannot effectively capture higher-order feature combinations; on the other hand, deep learning-based methods using deep neural networks to learn higher-order feature interactions are implicit, and lack some explanation of the feature interaction process. To address these problems, this study is based on graph neural networks for feature interaction modeling, which are mainly as follows:

(1) An attention-aware graph matching feature interaction recommendation algorithm is proposed. The method transforms the recommendation problem into a graph matching problem by extending the traditional point-to-point modeling paradigm to a graph-to-graph. First, to capture different levels of feature interaction processes, the method models user and item information as two feature graphs and models feature interactions from both internal and external interactions. The internal interaction captures the feature interactions within a single feature graph, and the external interaction captures the node matching between two feature graphs. Secondly, attention mechanism approach is introduced to capture the importance of interactions in both feature node interaction phase and node representation fusion phase. Finally, an experimental study is conducted on two publicly available datasets, and the experimental results show that the proposed algorithm has good prediction results compared to the comparison algorithm.

(2) A hierarchical two-level graph fusion feature interaction recommendation

algorithm is proposed. The method addresses the deficiency that attention-aware graph matching feature interactions do not introduce contextual information, and transforms the recommendation problem into a graph classification problem. First, each feature field is regarded as a node in the feature graph, and the connected edges between nodes are regarded as the interactions between features, and meaningful feature interactions are obtained by learning to update the connections between nodes through a hierarchical graph structure. Secondly, a bi-level node and graph representation generation module is designed, which contains two levels of feature interaction and fusion process, local level interaction uses edge weights to update the representation of nodes, and global level interaction dynamically captures the importance of each feature field by compression stimulus. Further, a bilinear fusion method is designed to fuse the node information from multiple perspectives for both levels of interaction. Finally, the results are experimentally compared with several baseline models to demonstrate that the hierarchical bi-level graph fusion feature interaction recommendation algorithm has higher prediction accuracy.

Key words: Recommender Systems; Graph Neural Networks; Feature Interaction; Graph Matching; Graph Structure Learning

目 录

1 绪论.....	1
1.1 研究背景与意义.....	1
1.2 国内外研究现状.....	2
1.2.1 基于传统的浅层特征交互推荐模型.....	2
1.2.2 基于深度学习的深层特征交互推荐模型	3
1.3 本文研究内容.....	5
1.4 本文结构安排.....	6
2 相关理论和技术	7
2.1 推荐系统概述.....	7
2.2 基于特征交互的推荐算法	8
2.2.1 FM 模型	8
2.2.2 DeepFM 模型	9
2.2.3 Fi-GNN 模型	10
2.3 图神经网络相关理论.....	13
2.3.1 图神经网络基本概念.....	13
2.3.2 常见图神经网络.....	14
2.3.3 图相似性学习.....	17
2.3.4 图结构学习.....	19
2.4 本章小结.....	21
3 注意力感知的图匹配特征交互推荐算法	23
3.1 引言.....	23
3.2 问题描述.....	23
3.2.1 符号描述.....	23
3.2.2 问题定义.....	23
3.3 模型介绍.....	24
3.3.1 模型概览.....	24
3.3.2 特征图构建模块.....	25
3.3.3 节点匹配图神经网络模块.....	25
3.3.4 节点和图表示生成模块.....	27
3.3.5 预测和优化模块.....	27
3.4 算法流程.....	28
3.5 实验结果与分析.....	29
3.5.1 实验数据集.....	29
3.5.2 评估指标.....	30

3.5.3 基线方法	30
3.5.4 实验设置	31
3.5.5 实验结果	31
3.5.6 参数对比分析	32
3.5.7 消融实验	34
3.6 本章小结	34
4 分层双级别图融合特征交互推荐算法	37
4.1 引言	37
4.2 问题描述	37
4.2.1 相关概念与表示	37
4.2.2 问题定义	38
4.3 模型介绍	38
4.3.1 模型概览	38
4.3.2 特征图构建模块	39
4.3.3 双级别节点和图表示生成模块	41
4.3.4 预测和训练模块	44
4.4 算法流程	44
4.5 实验结果与分析	46
4.5.1 实验数据集	46
4.5.2 评估指标	47
4.5.3 基线方法	47
4.5.4 实验设置	47
4.5.5 实验结果	48
4.5.6 参数对比分析	49
4.5.7 消融实验	51
4.6 本章小结	52
5 总结与展望	53
参考文献	55

1 绪论

1.1 研究背景与意义

随着互联网的快速发展,海量的信息丰富了人们的物质和精神世界,人们可以通过各种各样的方式访问互联网,获取自己所需要的信息。但是,全球信息数据量的爆炸式增长也为用户带来了选择难题,用户很难从复杂丰富的信息中及时准确地找到符合自己需求的信息^[1]。同时,内容服务商也很难选出满足用户需求的信息^[2],进而影响了他们为用户所提供服务质量。当信息的获取不再是问题,海量的数据反而干扰了用户的正常决策,信息过载问题的解决成了当务之急^[3-4]。推荐系统作为一种主要解决信息过载的方式,近年来引起了各方参与者的重视并被广泛应用。推荐系统根据用户和物品的基本属性信息以及两者之间的历史交互行为数据,发现用户的普遍喜好和潜在需求规律,获取有价值的信息,向用户推送其感兴趣的商品列表。推荐系统的实用性和便捷性在生活中已经被广泛证实,例如,淘宝、天猫、亚马逊等电商平台均搭载推荐系统,根据用户的历史购买记录、商品详细介绍、用户评论等信息,给目标用户提供推荐列表,其中不仅包括推荐的内容和用户交互的反馈方式,甚至还会给出推荐理由^[5]。一个好的推荐系统不仅可以提升用户的满意度,提高用户对平台的忠实度,而且可以给商家带来丰厚的利润,最终实现用户和商家的双赢。

目前推荐系统中的推荐算法主要分为三类:基于内容的推荐算法^[6-7]、基于协同过滤的推荐算法^[8-9]和混合推荐的推荐算法^[10-11]。基于内容的推荐通过对用户和商品的属性直接分析和计算做出推荐,基于协同过滤的推荐通过用户和商品的交互信息给用户推荐感兴趣的物品。上述两种方法都有各自的缺点:基于内容的推荐只能推荐内容相似的商品,从而缺少推荐的多样性,不能给用户带来一定的新鲜感;基于协同过滤的推荐无法避免数据的稀疏性和冷启动问题。混合推荐不只使用单纯的一种推荐机制,而是将多个方法融合在一起,从而达到更好的推荐效果,一定程度缓解了上述问题。

在现有的研究中,利用额外的辅助信息进行建模同样也可以缓解数据稀疏和冷启动问题。辅助信息建模中最重要的是如何建模特征之间的交互,从而形成具有高信

息增益的特征组合。在特征交互中,通常分为一阶特征组合、二阶特征组合和高阶特征组合。一阶特征可以看作是全局建模,二阶以及高阶特征组合是特征间个性化建模。低阶组合特征和高阶组合特征对最终的个性化推荐都起到了十分重要的作用,广泛地应用于目前的特征交互推荐算法中,成为了当前的研究热点之一。

1.2 国内外研究现状

对推荐系统来说,仅使用原始的信息进行预测取得的效果不佳。为了给用户做出更精准的个性化推荐,需要考虑信息之间的交互情况。国内外学者对特征交互推荐进行了广泛研究,并提出了许多推荐模型。从特征交互的角度看,这些推荐模型主要可以分为两种类型,即基于传统的特征交互推荐模型和基于深度学习的特征交互推荐模型。本节将从这两个方面简要介绍当前的研究现状。

1.2.1 基于传统的浅层特征交互推荐模型

当前,大多基于传统的低阶特征交互推荐模型发展比较成熟,这为后续构建基于深度学习的模型打下基础。相对于只利用用户和物品相互行为信息进行推荐的协同过滤模型来说,逻辑回归(Logistic Regression, LR)模型^[12]综合利用了用户信息、物品信息、上下文信息等多种不同特征,具有可解释性强、开销小和利于简单部署等特点。但 LR 模型表达能力不强,无法捕捉不同特征之间的交互情况,只能对单一特征进行线性组合。POLY2 模型^[13]延续了 LR 模型的思想,对不同特征之间进行两两交叉建模,为模型加入了二阶非线性交互特征,从一定程度上增加了组合特征的可能性,但其根本上还是线性模型,使用暴力组合特征让原本稀疏的特征向量变得更为稀疏,导致模型的效率不高。为了解决 POLY2 模型的缺陷,因子分解机(Factorization Machine, FM)模型^[14]在 2010 年被 Rendle 提出。FM 为了降低训练模型时的开销,在 POLY2 的基础上引入了矩阵分解的思想,为每个特征学习一个隐向量,在进行特征之间的交互时,将两个特征的隐向量内积作为交互权重,赋予重要的特征交互更大的权重。同时,FM 有效地解决了数据稀疏问题,并且可以学习历史上从未出现过的特征组合。在 FM 模型提出之后,基于 FM 的一系列变体被提出。域感知的因子分解机(Field-aware Factorization Machine, FFM)模型^[15]引入了特征域 field 的概念,将

相同类别的特征都关联到一个特征域上。对于不同的域，每个特征都能学习到不同的隐向量表示。FFM 模型在多项比赛中斩获冠军，但因其参数量过大，在处理实际业务时应用性较差。域权重因子分解机（Field-weighted Factorization Machine, FwFM）模型^[16]使用一个标量来表示不同特征域之间的交互强度，在提高模型表达能力的同时减少了模型的参数量。域矩阵因子分解机（Field-matrixed Factorization Machine, FmFM）模型^[17]用一个二维的域矩阵替换了 FwFM 的标量，提高了模型的性能。注意力因子分解机（Attentional Factorization Machines, AFM）^[18]区分了不同特征交互的重要性，使用注意力模块来提取不同特征交互对最终结果的重要性，对预测贡献程度高的特征交互赋予更大的权重值。Facebook 公司提出了梯度提升树（Gradient Boosting Decision Tree, GBDT）^[19]与 LR 相融合的组合算法。该方法首先使用 GBDT 得到某个叶子节点对应的一组特征组合，再将特征以独热向量的形式传入 LR 模型进行学习。相较于单一的 LR 模型，该组合算法有着更优的性能，但非端到端两段式训练过程，增加了计算的复杂度。

1.2.2 基于深度学习的深层特征交互推荐模型

深度学习在图像处理^[20]、自然语言处理^[21]和生物医学^[22]等众多领域取得优异的成果，近几年学者们也将深度学习引入推荐系统中，逐渐发展成为了新一代推荐模型的主流技术。以下将从高阶特征的提取、低阶特征与高阶特征的结合、显式建模特征交互、图结构建模特征交互四个方面介绍相关研究现状。

（1）高阶特征的提取

基于因子分解机的神经网络（Factorization Machine Supported Neural Network, FNN）^[23]灵感来自于图像领域通过相邻层连接扩大感受野的做法，使用 DNN 对 FM 建模的二阶特征再次进行交叉，从而产生更加高阶的特征组合，达到增强模型学习能力的目的。Deep Crossing 模型^[24]将稀疏特征嵌入为稠密的向量，把不同类型的特征向量拼接后送入带有残差的多层感知机（Multilayer Perceptron, MLP）中，实现完成特征各个维度充分交叉融合的目的。基于乘积的神经网络（Product-Based Neural Network, PNN）^[25]使用内积和外积操作对不同特征之间的交互进行捕捉，提高了模型的表达能力。神经因子分解网络（Neural Factorization Machine, NFM）^[26]针对 FM

不能进行高阶交互的不足，用一个表达能力强的函数代替 FM 中二阶隐向量内积部分，并且引入特征交叉池化层，加强了交互能力。

（2）低阶与高阶特征的结合

低阶特征和高阶特征对于最终的推荐预测都是有帮助的，一味地追求高阶特征提取有可能会淡化低阶交互的影响，因此需要对低阶与高阶交互进行结合。Wide&Deep 模型^[27]通过并行结合 LR 和 DNN 对低阶与高阶交互同时建模，实现了 Wide 部分的“记忆能力”和 Deep 部分的“泛化能力”，在当时成为业界的主流模型。Deep&Cross 模型^[28]借鉴了 Wide&Deep 模型的思路，使用多层交叉层代替了 Wide 部分，增加特征之间的交互力度。基于神经网络的因子分解机（Factorization Machine Based Neural Network, DeepFM）模型^[29]针对 Wide&Deep 模型 Wide 部分自动进行特征组合能力不足的问题，用 FM 代替了 Wide 部分，加强了浅层网络的特征交互能力。但是该模型的高阶交互使用了隐式的建模方式，不符合特征交互的直观理解，缺乏良好的解释性。

（3）显式建模特征交互

学者们通过设计独特的模块来显式地建模特征交互，用以提高算法的可解释性。极深因子分解机（eXtreme Deep Factorization Machine, xDeepFM）模型^[30]设计了压缩交互式网络（Compressed Interaction Network, CIN）代替 Deep&Cross 中的多层交叉层，从而进行显式地特征交互。双线性特征交互网络（Feature Importance and Bilinear Feature Interaction NETwork, FiBiNET）^[31]借鉴图像领域的压缩激励机制^[32]动态学习特征重要性的策略，并使用双线性特征交互网络建模细粒度的特征交互，显式地构建交互特征。自动特征交互（Automatic Feature Interaction Learning, AutoInt）模型^[33]使用自注意力机制显式地学习高阶特征交互，有效提升了预测的准确率。解耦自注意网络（Disentangle Self-attention Network, DESTINE）^[34]将一元交互特征重要性计算从二阶特征交互中解耦出来，分别学习一元特征重要度和特征对交互，相对于 AutoInt 模型增强了单个特征域对最终结果的影响。分层注意力可解释预测模型（Interpretable CTR Prediction Model with Hierarchical Attention, InterHAt）^[35]通过新的层次注意力机制量化了任意阶数的特征交互影响，根据学习到的特征显著性解释了推荐的结果。自适应因子网络（Adaptive Factorization Network, AFN）模型^[36]设计了带有对数转换

层的结构，将特征组合中每个特征的幂转换为要学习的参数，提取了 MLP 忽略的特征，提升了模型性能的同时增加模型的可解释性。

（4）图结构建模特征交互

随着图神经网络在各个领域取得成功的同时，基于图结构进行特征交互建模也取得了不错的效果。特征交互的图神经网络 (Feature Interaction Graph Neural Network, Fi-GNN) 模型^[37]使用自注意力网络和图神经网络学习特征节点的交互，为后续的图结构建模开创了理论先河。 L_0 正则化统计交互图神经网络 (L_0 activation regularization Statistical Interaction Graph Neural Network, L_0 -SIGN)^[38]使用图神经网络自动发掘对推荐系统有利的二阶特征交互，将特征节点和检测到的有益特征交互边作为输入特征图，并通过有效地建模和聚合特征节点输出最终的推荐结果。分层意向嵌入网络 (Hierarchical Intention Embedding Network, HIEN)^[39]自上而下地考虑属性图中属性之间的依赖性，同时提出用户意图和项目意图的双重意图，以准确地学习用户和项目表示。图因子分解机 (Graph Factorization Machine, GraphFM) 模型^[40]使用图神经网络可以通过迭代多层以建模任意阶数交互的优势，克服了 FM 只能建模二阶交互的不足。神经图匹配协同过滤 (Neural Graph Matching based Collaborative Filtering, GMCF) 模型^[41]考虑了不同级别的特征交互对最终推荐预测的影响，将特征交互分为内部交互和交叉交互两个部分，用以共同决定最终的预测结果。

1.3 本文研究内容

本研究重点关注推荐系统中特征交互的预测任务，对分类变量进行建模，借助图结构完成特征交互。利用图机器学习中两大相关技术，即图匹配技术和图结构学习技术，并融合图神经网络主要框架，对现有的图神经网络特征交互推荐算法进行改进。本研究提出了注意力感知的图匹配特征交互推荐算法和分层双级别图融合特征交互推荐算法，并通过实验验证了模型的有效性、合理性和准确性。

主要研究内容如下：

首先，提出了一种注意力感知的图匹配特征交互推荐 (Attention-aware Graph Matching Recommendation, AGMRec) 算法。AGMRec 算法的基本思想是将用户属性和物品属性建模为用户和物品两个特征图，把特征交互点到点的建模范式扩展到

图到图，从内部和外部两个角度来进行特征节点之间的交互，并通过注意力机制对交互过程进行合理的权重分配。基于两个公开数据集开展实验验证，算法的整体性能、模块消融实验、模型超参实验等方面都证实了 AGMRec 算法的合理性和有效性。

其次，鉴于推荐任务中用户和物品发生交互时的上下文环境信息仍能对推荐结果产生影响，进一步引入上下文信息特征从而提高推荐的准确性，提出了分层双级别图融合推荐（Hierarchical Dual-level Graph Feature Interaction, HDGFI）算法。HDGFI 将特征域信息建模为图结构中的节点，运用分层邻接矩阵 Top-k 选择机制筛选重要的特征交互，并从局部和全局两个角度进行特征节点之间的交互。最后，通过实验验证了 HDGFI 算法的有效性和准确性。

1.4 本文结构安排

论文章节结构如下：

第一章：绪论。首先，对推荐系统的研究背景与意义进行了相关介绍；其次，概述了特征交互推荐算法的国内外研究现状；最后，阐述本论文的研究内容和章节结构。

第二章：相关理论和技术。对涉及到的相关技术进行介绍。首先，概述了推荐系统以及推荐系统中的主要算法；其次，介绍了图神经网络的基本定义和常见的三种图神经网络；最后，介绍了图相似性学习方法和图结构学习方法。

第三章：注意力感知的图匹配特征交互推荐算法。首先，介绍了目前特征交互推荐算法存在的问题；其次，针对目前算法的不足，提出了注意力感知的图匹配特征交互推荐算法；接下来，对实验所用到的数据集、评价指标和基线模型进行描述；最后，对实验结果进行了展示和分析，证明该算法能够有效提升推荐预测效果。

第四章：分层双级别图融合特征交互推荐算法。首先，介绍了相关算法的局限性；然后，提出了分层双级别图融合特征交互推荐算法，并对整体框架、具体的组件进行了详细介绍；最后，在三个公开数据集上与多个基线方法进行分析比较，证明模型的有效性和准确性。

第五章：总结与展望。对本文研究工作进行了总结，指出存在的不足，并对未来工作进行了展望。

2 相关理论和技术

本章是论文相关理论和技术介绍。首先，介绍了推荐系统的基本概念和基本理论，对推荐系统进行全面了解；然后，介绍具有代表性的特征交互推荐算法原理；其次，介绍图神经网络相关理论；最后，介绍图相似性学习和图结构学习相关技术。

2.1 推荐系统概述

推荐系统作为一种信息过滤工具，通过对用户的偏好进行学习和预测，并向用户推荐用户可能喜爱的物品。推荐系统包含三大组成部分：用户、物品和推荐方法。其中，用户是推荐系统的使用者，物品是推荐系统要推荐的商品，推荐方法决定着推荐结果的好坏，也是推荐系统的核心研究内容。目前，推荐系统在电商平台、金融市场和社交网络等众多领域中得到了广泛应用；它在帮助用户快速找到自己所需物品的同时，也为企业和制造商带来巨大效益。一般来说，推荐系统主要分为三个阶段：分别是召回、排序和调整。推荐系统的三个阶段如图 2.1 所示。

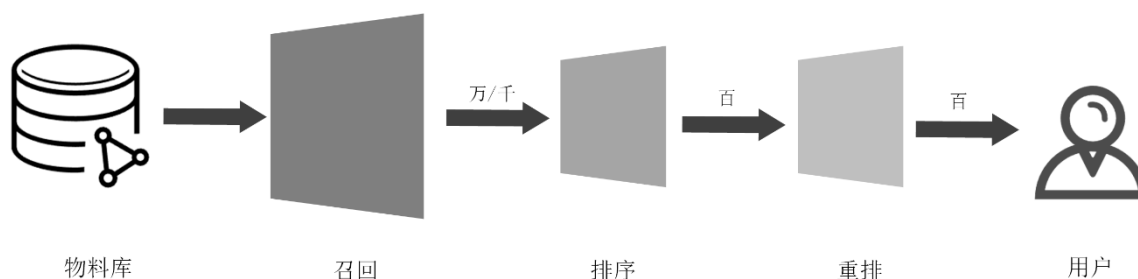


图 2.1 推荐系统的三个阶段

（1）召回阶段

召回阶段主要根据用户的部分特征，从海量的物品库中，快速召回小部分用户潜在感兴趣的物品^[42]。由于召回面对的候选物品数量非常庞大，所以召回阶段的算法通常具有轻量、快速和低延迟等特性。可使用的召回策略主要有协同过滤召回、内容相似召回、用户画像召回和热门召回等。由于每个召回策略的出发点和侧重点不同，一般会采用多种召回策略一起进行召回，将尽可能把与用户相关的物品全部找出来。

（2）排序阶段

经过召回阶段，候选物品数量已经大大减少了，排序阶段使用复杂的点击率预测

模型，从用户与物品的交互行为中挖掘更为丰富的信息，将召回的结果进行排序。排序阶段最常用的是 Pointwise 排序方法，它考虑用户和物品的特征，并关注交互时的上下文环境信息^[43]。Pointwise 排序方法有基于传统机器学习的 LR、FM 等算法，以及基于深度学习的 DeepFM、Wide&Deep 等算法。若用户和物品初始数据量较少，可以跳过召回阶段，直接进行排序预测。

（3）调整阶段

在经过召回和排序阶段，由于要考虑购买物品的时效性和真实性，对于重复推荐、已经或正在购买、已经下线的物品进一步过滤。当前两阶段推荐结果不够时，选取热门的物品对候选集进行补充，最后将补充的候选集和经过筛选后的候选集进行合并，把含有完整数据的商品推荐列表反馈给用户。

2.2 基于特征交互的推荐算法

本节按照特征交互推荐算法的发展历史介绍一些主要的代表性算法模型。主要包含基于传统机器学习特征交互的 FM 模型和基于深度学习的 DeepFM 和 Fi-GNN 模型。

2.2.1 FM 模型

传统的逻辑回归模型学习能力有限，因此 Steffen Rendl 等人提出了 FM 模型。FM 模型将逻辑回归模型扩展为二阶特征交互，它具有两大优势：其一是进行特征组合，以两两组合的方式，引入交叉项特征，使模型能够学习到特征与特征之间的关系；其二是缓解了高维度稀疏矩阵带来的维度爆炸问题，通过引入隐向量，对原始为 0 的特征也可以很好地学习。假设一个输入样本的特征向量为 $X \in \mathbb{R}^n$ ，初始的模型表达如公式(2-1)所示。

$$y(X) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n w_{ij} (x_i * x_j) \quad (2-1)$$

其中， n 代表每条样本数据的特征数量， x_i 是第 i 个特征的值， w_0 、 w_i 和 w_{ij} 分别代表全局偏差、第 i 个特征的权重以及第 i 个特征和第 j 个特征的交互权重。FM 模型在保

留特征线性关系的同时，通过增加任意两个特征之间的交互结果来提高预测的效果。从公式(2-1)可以看出二阶组合特征的参数量共 $n(n-1)/2$ ，在实际应用场景中，样本的特征向量是及其稀疏的，表现为 x_i 和 x_j 的值大都为 0。若直接学习二阶特征交互的参数，而实际问题中的二阶交互特征有时不存在。为了解决这个问题，引入了矩阵分解的思想，为每一个特征 x_i 学习一个隐向量 v_i ，将二阶交互权重参数 w_{ij} 用 $v_i^T v_j$ 去替代，改进后如公式(2-2)所示。

$$y(X) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j \quad (2-2)$$

其中， $\langle v_i, v_j \rangle$ 代表两个隐向量之间的点积操作， v_i 隐向量可以设置为远小于特征向量维度的长度。根据矩阵变换原理，将公式(2-2)进行优化为公式(2-3)，使模型的时间复杂度从原来的 $O(kn^2)$ 降低到 $O(kn)$ 。公式(2-3)的表示如下：

$$\begin{aligned} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j &= \frac{1}{2} \sum_{i=1}^n \sum_j \langle v_i, v_j \rangle x_i x_j - \frac{1}{2} \sum_{i=1}^n \langle v_i, v_i \rangle x_i x_i \\ &= \frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n \sum_{f=1}^k v_{i,f} v_{j,f} x_i x_j - \sum_{i=1}^n \sum_{f=1}^k v_{i,f} v_{i,f} x_i x_i \right) \\ &= \frac{1}{2} \sum_{f=1}^k \left[\left(\sum_{i=1}^n v_{i,f} x_i \right) \left(\sum_{j=1}^n v_{j,f} x_j \right) - \sum_{i=1}^n v_{i,f}^2 x_i^2 \right] \\ &= \frac{1}{2} \sum_{f=1}^k \left[\left(\sum_{i=1}^n v_{i,f} x_i \right)^2 - \sum_{i=1}^n v_{i,f}^2 x_i^2 \right] \end{aligned} \quad (2-3)$$

其中， $v_{i,f}$ 是隐向量 v_i 的第 f 个元素。FM 能够对二阶特征交互进行建模，但考虑到效率问题，一般不会直接使用 FM 来进行高阶特征交互的学习。

2.2.2 DeepFM 模型

FM 虽然可以对稀疏的数据进行特征交互建模，但只能用线性方式对线性多阶交互进行建模。实际应用场景中存在着数据高度非线性关系，仅用线性交互模型难以建模。深度学习网络具有建模复杂特征交互的能力，DNN 建模高阶特征交互逐渐成为

主流。DeepFM 将 FM 模块与 DNN 模块并行，两个模块共享输入特征向量。其中，FM 模块用于捕捉一阶和二阶交互特征，DNN 模块用于捕捉高阶交互，整个模型是一个端到端的结构。DeepFM 模型架构如图 2.2 所示。

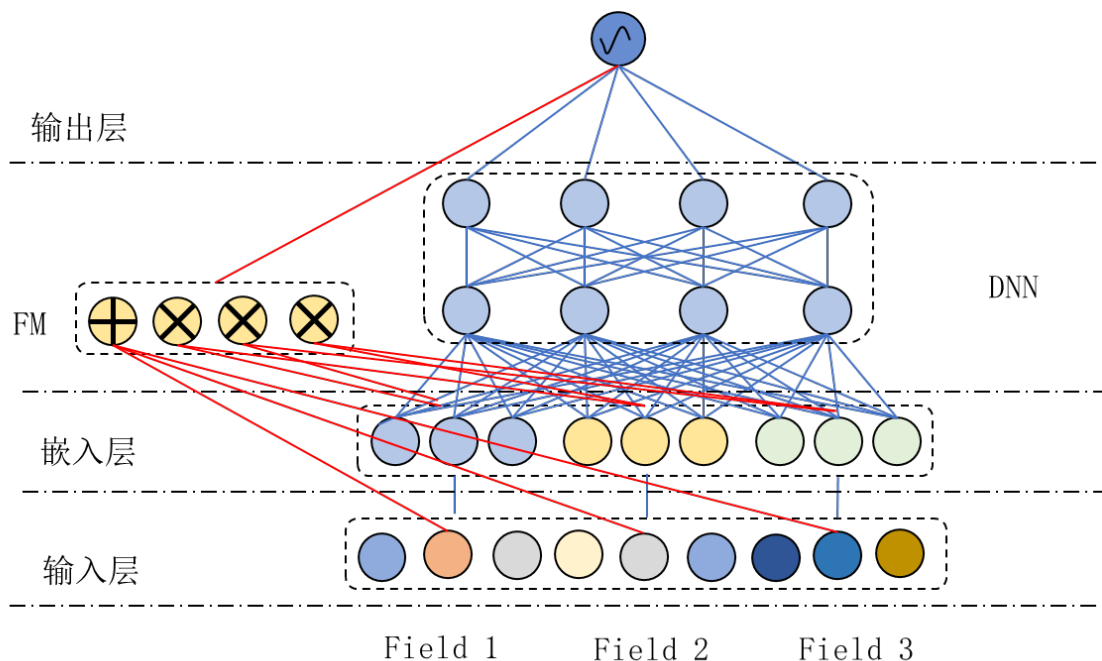


图 2.2 DeepFM 模型框架图

FM 部分的公式表达在 2.2.1 节已经详细介绍过，本节将介绍 DNN 模块和两部分融合方法。DNN 模块的输入是所有特征向量的拼接向量，其表示如公式(2-4)所示。

$$y_{DNN} = Relu(W_l * h_{l-1} + b_l) \quad (2-4)$$

其中， $Relu$ 是非线性激活函数， W_l 和 b_l 是第 l 层的训练参数。最终的模型如公式(2-5)所示。

$$\hat{y} = Sigmoid(y_{FM} + y_{DNN}) \quad (2-5)$$

其中， $Sigmoid$ 激活函数将预测值标准化在 0 到 1 之间。DeepFM 模型使用并行捕捉低阶和高阶特征交互的方式为后续推荐算法设计提供了新的路线，众多的研究在此基础上进行了更新与完善。

2.2.3 Fi-GNN 模型

当前基于 DNN 高阶特征交互模型的输入是把特征向量进行简单拼接,然后将结果输入到多层感知机中。这些做法隐式地捕捉了高阶交互,但缺乏很好的解释性。图神经网络通过迭代进行消息传递和节点表示更新,可以显式地捕捉节点之间的交互情况。Li 等人在 2019 年提出 Fi-GNN,这是第一个用图神经网络建模高阶交互的推荐算法。在 Fi-GNN 中,用图结构表示各类的特征,每一个节点代表一个特征域,通过获取邻居节点的状态信息并结合自身当前的隐藏状态来更新自己的当前状态。每多堆叠一个 GNN 层,相当于目标节点与邻居节点交互更深了一层,因此,时间步的数量等于特征交互的阶数。Fi-GNN 模型框架如图 2.3 所示。

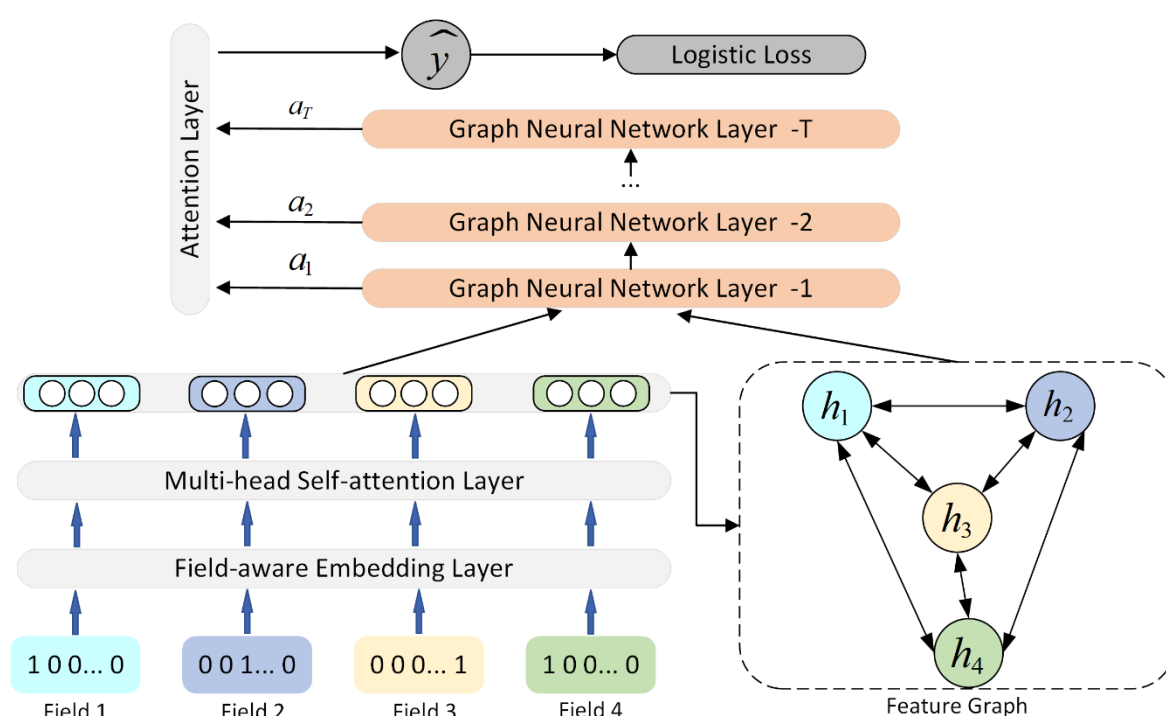


图 2.3 Fi-GNN 模型框架图

Fi-GNN 模型中,首先通过一个特征嵌入层将特征转化为低维稠密向量,然后通过多头注意力机制捕捉成对特征的复杂依赖,接下来构建加权的全连接图,最后通过堆叠图神经网络来聚合邻居信息。特征交互图神经网络是 Fi-GNN 的核心部分,本节主要介绍特征交互图神经网络部分。

在特征交互图神经网络中,主要分为三个步骤,即初始化参数、状态聚合和状态更新。初始化参数将经过多头注意力机制层得到的向量作为特征图节点初始的隐藏

向量。在交互时间步 t 状态聚合时, 每个节点聚合邻居的状态信息, 具体的思路是节点 n_i 的聚合信息是邻居的转换信息的总和: $a_i^t = \sum_{n_j \rightarrow n_i \in \mathcal{E}} A[n_j, n_i] W_p h_j^{t-1}$ 。其中, A 是带有权重的邻接矩阵, W_p 是变换函数。显然, 变换函数和邻接矩阵决定了节点的交互, 由于每条边上的交互应该不同, 因此, 需要每条边都有一个独特的权重和变换函数。在计算邻接矩阵权重时, 通过注意力机制学习边权重, 节点 n_i 到节点 n_j 的边的权重计算如公式(2-6)所示。

$$w(n_i, n_j) = \frac{\exp(\text{Leakyrelu}(W_w[e_i \parallel e_j]))}{\sum_k \exp(\text{Leakyrelu}(W_w[e_i \parallel e_k]))} \quad (2-6)$$

$$A[n_i, n_j] = \begin{cases} w(n_i, n_j), & \text{if } i \neq j \\ 0, & \text{else} \end{cases}$$

考虑到所有边都使用固定的变换函数无法模拟灵活的交互, 因此每条边的唯一变换是必不可少的。由于特征交互图是具有大量边的完全连通图, 如果为每条边分配一个变换权, 会造成参数空间的冗余和较高的时间复杂度。因此, Fi-GNN 为每条边分配了一个输入矩阵 W_{in} 和输出矩阵 W_{out} , 利用输入矩阵和输出矩阵构造变换函数。具体的变换函数如公式(2-7)所示。

$$W_p^{n_i \rightarrow n_j} = W_{out}^i W_{in}^j \quad (2-7)$$

在状态更新阶段, 利用循环神经网络单元 GRU 对最近一次节点隐藏状态和聚合状态信息进行操作, 对应的表达式为 $h_i^t = \text{GRU}(h_i^{t-1}, a_i^t)$ 。具体的计算如公式(2-8)所示。

$$\begin{aligned} z_i^t &= \sigma(W_z a_i^t + U_z h_i^{t-1} + b_z) \\ r_i^t &= \sigma(W_r a_i^t + U_r h_i^{t-1} + b_r) \\ \tilde{h}_i^t &= \tanh(W_h a_i^t + U_h (r_i^t \odot h_i^{t-1}) + b_h) \\ h_i^t &= \tilde{h}_i^t \odot z_i^t + h_i^{t-1} \odot (1 - z_i^t) \end{aligned} \quad (2-8)$$

其中 W_* 、 U_* 和 b_* 是可训练参数。Fi-GNN 同时引入了残差连接, 同时考虑高阶和低阶交互, 促进低阶特征的重用和训练时梯度的反向传播, 计算过程如公式(2-9)所示。

$$h_i^t = \text{GRU}(h_i^{t-1}, a_i^t) + h_i^l \quad (2-9)$$

2.3 图神经网络相关理论

得益于计算机硬件计算性能和存储能力的大幅提升,近年来深度学习得到了快速发展,并在各个领域获得巨大成功。然而,现有的神经网络只能对欧式结构数据进行处理,不能直接将传统的深度学习模型迁移应用到非欧式结构数据中^[44]。图神经网络将实体和实体之间的关系建模为图结构,直接对节点信息和边信息进行集成与学习,在处理实际问题时可以将各种非欧式数据问题抽象为图结构。图神经网络是一个学习方法的总称,核心的思想是基于消息传递机制不断更新节点的表示,去更好地完成下游任务。学者们针对不同类型的图、不同的消息传递机制设计出了各种图神经网络的变体。本章将从图神经网络的基本概念、常见的图神经网络和图神经网络的应用三个方面详细介绍图神经网络。

2.3.1 图神经网络基本概念

图结构在生活中无处不在,是一种典型的数据结构。图表示为 $G = \{V, E\}$, 其中 V 是节点的集合, E 是图结构中边的集合。 $v_i \in V$ 表示图结构中的一个节点, $e_{ij} = (v_i, v_j)$ 表示节点 v_i 和 v_j 之间有一条边连接。 $N(v_i) = \{v_j | (v_i, v_j) \in E\}$ 表示节点 v_i 的邻居集合。一般来说,图可以从下面三个方面来分类。

(1) 有向图和无向图

按照边是否有方向,图可以分为有向图和无向图。一个图结构中,若明确用箭头指明了边的方向,这种图称为有向图,否则称为无向图。在无向图中,一条边用圆括号表示,且 (v_i, v_j) 与 (v_j, v_i) 表示意义相同。在有向图中,一条边用尖括号表示,且 $\langle v_i, v_j \rangle$ 与 $\langle v_j, v_i \rangle$ 表示意义不同。 $\langle v_i, v_j \rangle$ 表示从顶点 v_i 发向顶点 v_j 的边, v_i 为始点, v_j 为终点。

(2) 同构图和异构图

按照节点和边类型,图可以分为同构图和异构图。若节点的类型和边的类型只有一种,这种图称为同构图,否则称为异构图。图中节点类型映射函数 $\phi: V \rightarrow A$, 边类

型映射函数 $\varphi: E \rightarrow R$ ，其中 A 和 R 分别代表了节点类型集合和边类型集合。当 $|A| + |R| \geq 2$ 时代表异构图，当 $|A| = 1, |R| = 1$ 时代表同构图。在同构图中，数据只存在一种节点类型和边连接关系，在构建网络时所有图结构中节点共享同样的模型参数。在异构图中，不同类型的节点拥有不同维度的特征和属性。

(3) 普通图和超图

按照边连接节点的个数，图可以分为普通图和超图，若一条边可以连接任意数量的节点，这种图称为超图。若一条边只能连接两个节点，则这种图称为普通图。超图可以表示为 $G = \{V, E, W\}$ ，其中 V 为顶点集， E 为超边集， W 为超边权重。

图神经网络在不同图上进行学习的核心思想是迭代更新节点的信息，在迭代的过程中结合自身节点信息和邻居节点信息共同来更新目标节点^[45]。图神经网络在计算过程中主要包括两步，即消息构建和信息聚合。消息构建指的是图中的节点产生一条消息并传递给它的邻居节点，如公式(2-10)所示。

$$m_u^{(l)} = \text{Message}^{(l)}(h_u^{(l-1)}) \quad (2-10)$$

其中， $h_u^{(l-1)}$ 表示节点 u 在第 $(l-1)$ 层的向量表征； $\text{Message}^{(l)}$ 表示第 l 层自定义的消息函数，此函数一般为线性变换函数； $m_u^{(l)}$ 为节点 u 产生的消息信息。在获得每个节点的消息信息后，下一步就是聚合邻域的信息，计算如公式(2-11)所示。

$$h_v^{(l)} = \text{Aggregation}^{(l)}(\{m_u^{(l)}, u \in N(v)\}) \quad (2-11)$$

其中， $N(v)$ 表示节点 v 的邻居集合； $\text{Aggregation}^{(l)}$ 表示第 l 层的自定义聚合函数，主要聚合方式有平均池化、加和池化以及基于注意力的池化等； $h_v^{(l)}$ 表示聚合了邻居节点信息的节点 v 向量。为了考虑节点自身在消息构建和信息聚合时的影响，对上面两个公式进行修正，修正方法如公式(2-12)所示。

$$\begin{cases} m_u^{(l)} = \text{Message}^{(l)}(h_u^{(l-1)}), u \in \{N(v) \cup v\} \\ h_v^{(l)} = \text{Aggregation}^{(l)}(\{m_u^{(l)}, u \in N(v)\}, m_v^{(l)}) \end{cases} \quad (2-12)$$

2.3.2 常见图神经网络

图神经网络根据不同的消息构建和信息传递方式产生了许多变体，接下来介绍几种常见的变体形式。

（1）图卷积网络（Graph Convolutional Network, GCN）

图卷积网络^[46]类似于卷积神经网络，实际是针对图结构的一个特征提取器。图卷积网络是谱图卷积一阶局部近似，每一个卷积层仅处理一阶邻域信息，通过叠加若干个卷积层实现多阶邻域的信息传递。图卷积网络结构如图 2.4 所示。

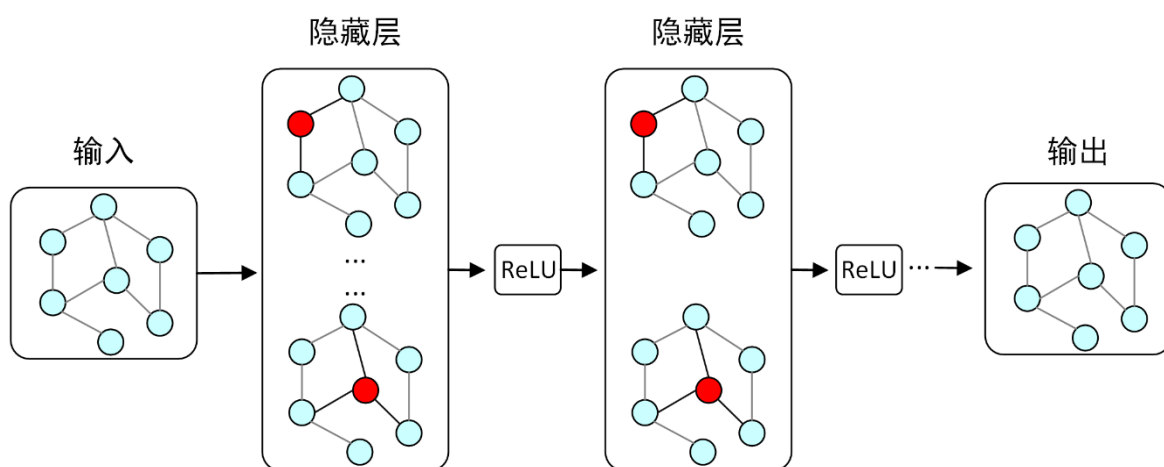


图 2.4 图卷积网络结构

图卷积网络输入的是节点信息向量矩阵 H 和邻接矩阵 A ，每一个卷积的计算过程如公式(2-13)所示。

$$H^{(l+1)} = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)}) \quad (2-13)$$

其中， σ 是非线性激活函数； $W^{(l)}$ 是第 l 层的线性转换矩阵； \tilde{A} 是添加了自连接的邻接矩阵； \tilde{D} 邻接矩阵的度矩阵。图卷积网络利用整个图的邻接矩阵来更新节点的表示，因此一般用于直推式任务而不能处理归纳式任务。

（2）图采样和聚集网络（Graph Saple and Aggregation, GraphSAGE）

为了解决图卷积网络不能处理归纳式任务的问题，图采样和聚集网络^[47]被提出。图采样和聚集网络结构如图 2.5 所示。

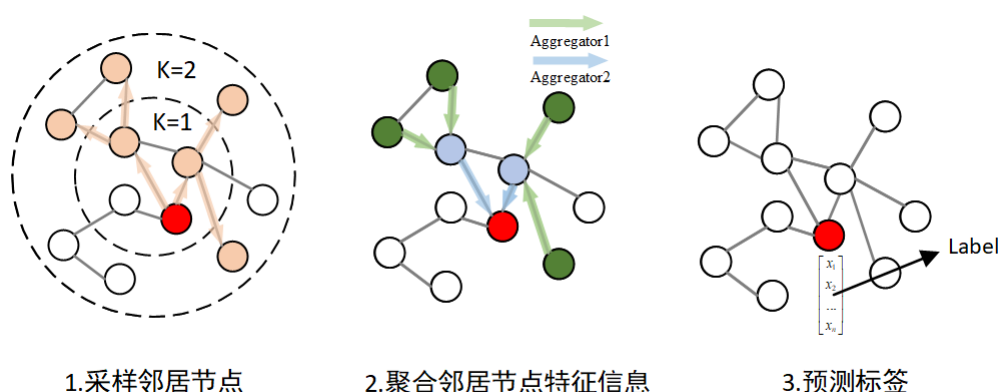


图 2.5 图采样和聚集网络结构

图采样和聚集网络分为三个部分，首先对邻居节点进行采样，然后通过多层聚合函数不断地将相邻节点融合，最后用融合后的节点信息针对下游任务进行学习。在采样阶段，定义一个采样邻居节点的个数 k ，当目标节点邻居节点个数大于 k 时，通过采样策略获取 k 个邻居节点，纳入目标节点的邻居节点集合。若目标节点邻居节点个数小于 k 时，有放回地重采样直到采样出 k 个节点。具体计算如公式(2-14)所示。

$$h_v^{(l)} = \sigma(W^{(l)} \cdot \text{CONCAT}(h_v^{(l-1)}, \text{Aggregation}(\{m_u^{(l)}, u \in N(v)\}))) \quad (2-14)$$

其中， $h_v^{(l)}$ 是节点 v 在第 l 层的表示， σ 是非线性激活函数。首先计算节点 v 的所有邻居节点消息信息 $m_u^{(l)}$ ，然后进行消息的聚合。为了避免节点 v 自身信息在传递过程中丢失，通过 CONCAT 操作将节点 v 上一层的输出 $h_v^{(l-1)}$ 和邻居聚合后的结果进行拼接，再通过一个线性映射和激活函数得到当前层的节点输出向量。与图卷积网络相比，图采样和聚集网络引入了三种消息聚合函数，分别为平均聚合，最大池化聚合和 LSTM 聚合。

(3) 图注意力网络 (Graph Attention Network, GAT)

图卷积网络中假设每个节点同样重要，从而忽略了节点之间关系的重要程度和每个邻居节点对目标节点的重要程度。注意力机制已经成功应用于图像处理、机器翻译等领域。研究人员将注意力机制与图卷积网络相结合，提出了图注意力网络^[48]。图注意力层的结构如图 2.6 所示。

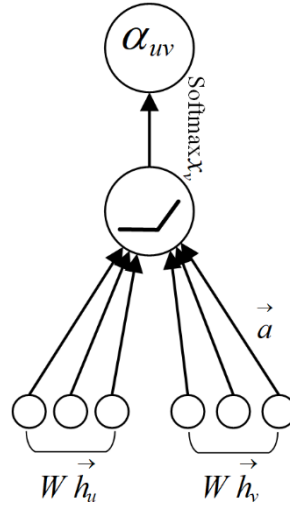


图 2.6 图注意力层结构

图注意力网络通过堆叠图注意力层，实现节点表示的更新。具体的计算过程如公式(2-15)和(2-16)所示。

$$\alpha_{uv} = \frac{\exp(\text{Att}(h_v^{(l-1)}, h_u^{(l-1)}))}{\sum_{k \in N(v)} \exp(\text{Att}(h_v^{(l-1)}, h_k^{(l-1)}))} \quad (2-15)$$

$$h_v^{(l)} = \delta(\sum_{u \in N(v)} \alpha_{uv} W^{(l)} h_u^{(l-1)}) \quad (2-16)$$

其中， α_{uv} 表示节点 u 到节点 v 的注意力系数， δ 是激活函数， $N(v)$ 表示节点 v 的邻居节点集合。 $\text{Att}(\cdot)$ 为注意力计算方法，常见方式是 $\text{Leakyrelu}(a^T [W^{(l)} h_v^{(l-1)} \parallel W^{(l)} h_u^{(l-1)}])$ ，其中， $W^{(l)}$ 为节点向量线性变换权重矩阵， a^T 是可学习的参数， \parallel 表示节点向量的拼接操作。通过公式(2-16)可以得到最终更新的节点 v 在第 l 层的表示 $h_v^{(l)}$ 。

2.3.3 图相似性学习

图神经网络通过迭代聚合局部信息更新目标节点表示，可以有效地学习结构化数据的表示和解决监督预测问题。当前大多数图神经网络的研究都是围绕单个图结构进行，而对于两个图或者多个图结构之间的相似性学习在计算机视觉、自然语言处理和推荐系统领域有更为广泛的应用需求。图相似学习有两种主要学习思路：一种是图嵌入，得到每个图的向量表示，不同图之间的相似度通过图向量的距离衡量。这种

方法的好处是如果需要对数据库中的大量图结构进行对比时，只需利用已有的一些距离计算方法对数据库中的图向量进行比较即可。另一种是对图进行结对比较，重点比较两个图之间各个节点的相似性，这种方式精确度更高，但随之而来的是高的计算复杂度。两种方法的流程如图 2.7 所示。

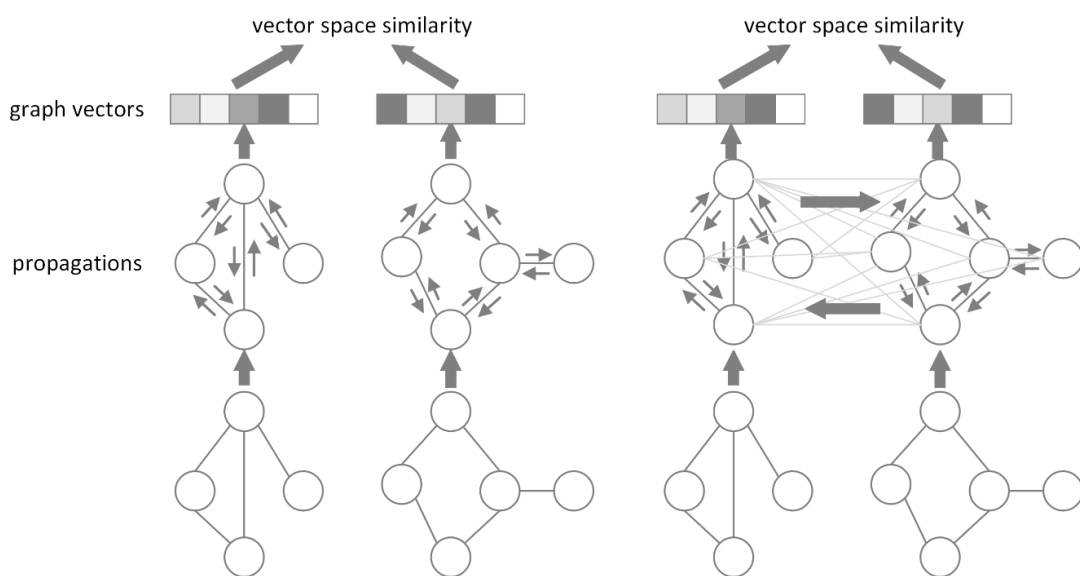


图 2.7 图相似性计算流程

具体来说，首先给定两个图结构 $G_1 = (V_1, E_1)$ 和 $G_2 = (V_2, E_2)$ ，每个图中的 V 和 E 代表节点集合和边集合，每个节点 v 都具有节点特征向量 h_v 。图相似性学习的任务是设计一个模型 f 来生成两个图的相似性得分，即 $score = f(G_1, G_2)$ 。图嵌入学习图向量的方式可以由 2.3.2 节所介绍的几种方法得出。

对图比较方法也称为图匹配模型。Li 等人 2019 年提出了图匹配网络 (Graph Matching Network, GMN)^[49] 改变了每个传播层中节点的更新方式，不仅捕捉了每个图在边上聚合的信息，而且还考虑了一个跨图匹配向量。跨图匹配向量描述了一个图中的节点与另一个图中的节点或多个节点的匹配程度。图匹配网络主要分为三个步骤，即编码层、传播层和聚合层，计算过程如下：

$$m_{j \rightarrow i} = f_{message}(h_i^{(l)}, h_j^{(l)}, e_{ij}), \forall (i, j) \in E_1 \cup E_2 \quad (2-17)$$

$$\mu_{j \rightarrow i} = f_{match}(h_i^{(l)}, h_j^{(l)}), \forall i \in V_1, j \in V_2 \text{ or } i \in V_2, j \in V_1 \quad (2-18)$$

$$h_i^{(l+1)} = f_{node}(h_i^{(l)}, \sum_j m_{j \rightarrow i}, \sum_j \mu_{j \rightarrow i}) \quad (2-19)$$

其中, $f_{message}$ 和 f_{node} 分别代表消息的构建和消息聚合, 实现方法和普通图神经网络一致; f_{match} 是一个传递跨图信息的函数, 使用一个基于注意力的模块来实现。 f_{match} 的计算方法如公式(2-20)和公式(2-21)所示。

$$a_{j \rightarrow i} = \frac{\exp(s_h(h_i^{(l)}, h_j^{(l)}))}{\sum_j \exp(s_h(h_i^{(l)}, h_j^{(l)}))} \quad (2-20)$$

$$\mu_{j \rightarrow i} = a_{j \rightarrow i}(h_i^{(l)} - h_j^{(l)}) \quad (2-21)$$

其中, $a_{j \rightarrow i}$ 是注意力权重; s_h 是向量相似度计算方法。相似度计算方法常见的有余弦相似度、欧几里得距离等。获得两个图的节点表示后, 通过全局池化操作得到两个图的整体嵌入表示, 最后使用向量空间中相似度量来计算两个图的相似性。

2.3.4 图结构学习

图神经网络对图的拓扑结构进行解析和建模过程中, 对设定的图结构敏感性很强^[50]。一方面, 图神经网络的迭代机制具有级联效应, 会将小噪声通过层层消息传播到邻域, 降低其它节点的表示效果。另一方面, 将现实对象交互建模为图结构时需要建模重要的子结构, 有限的先验知识和预定义的图结构只携带了部分信息, 甚至有些现实对象之间并不存在天然的图结构, 因此建模出的图结构不能有效地对目标任务进行分析和处理。为了给下游任务提供一个最佳的图结构, 许多学者针对图结构学习方法开展了广泛的研究。图结构学习的学习流程如图 2.8 所示。

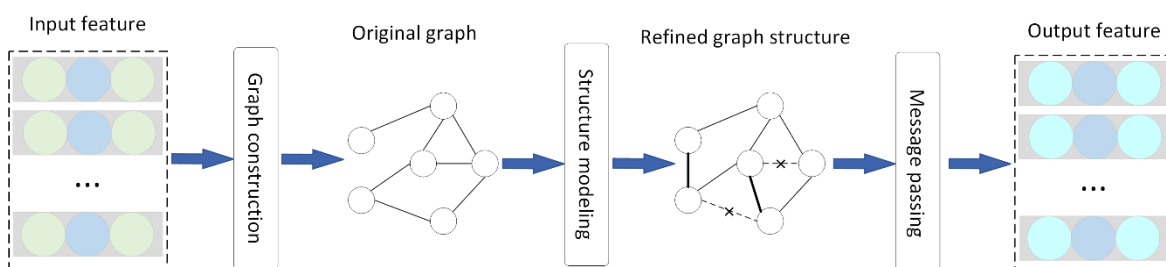


图 2.8 图结构学习流程

图结构建模是图学习的一个核心步骤。图结构建模完成对边连接的建模, 选择重

要的边进行保留,并过滤噪音边连接。目前的图结构建模可以分为三种类型,即基于度量的方法、基于神经网络的方法和直接方法。以下分别介绍这三种类型的图结构建模方法。

(1) 基于度量的方法

基于度量的方法使用核函数计算节点的相似度作为边权重,根据边倾向于连接相似节点的假设,通过优化边连接来生成更为准确的图结构。自适应图卷积神经网络(Adaptive graph convolutional neural networks, AGCN)^[51]提出了一个通用灵活的图卷积网络,利用广义马氏距离计算每对节点之间的相似度,使用高斯核来细化拓扑结构。迭代深度图学习(Iterative deep graph learning, IDGL)方法^[52]基于更好的节点嵌入学习更好的图结构和基于更好图结构学习更好的节点嵌入的关键原理,使用余弦相似度核的方式迭代学习图结构和节点嵌入,使图结构足够接近下游的预测任务。

(2) 基于神经网络的方法

与基于度量的方法相比,基于神经网络方法图结构建模是利用深度神经网络来建模给定节点特征和表示边的权重。图学习卷积网络(Graph learning-convolutional networks, GLCN)^[53]通过单层神经网络获得节点对之间的边连接关系。参数化拓扑去噪网络(Parameterized topological denoising network, PTDNet)^[54]使用多层感知机来学习邻接矩阵并去除与任务无关的边连接,提高了图神经网络的鲁棒性和泛化能力。

(3) 直接方法

直接方法将邻接矩阵视为一个可训练的参数,通过反向传播进行优化。由于直接方法训练过程中不依赖于节点的表示,所以具有更强大的灵活性,但缺点是相对于其它两种方法学习邻接矩阵参数较为困难。图学习神经网络(Graph learning neural networks, GLNN)^[55]将邻接矩阵的稀疏性和特征平滑性加入损失函数中,使图拓扑结构适应输入的数据。离散结构学习图神经网络(Learning Discrete Structures for Graph Neural Networks, LDS-GNN)^[56]将图结构学习问题视为双层规划问题,从具有可学习参数的伯努利分布中采样,对每对节点之间的边缘进行建模,在学习数据点之间的离散和稀疏依赖关系的同时训练图卷积网络参数。

2.4 本章小结

本章首先介绍了推荐系统实现过程，包括召回、排序和调整三个阶段；然后针对三个具有代表性的特征交互推荐算法 FM、DeepFM 和 Fi-GNN 分别进行了介绍；最后介绍了图神经网络的相关理论，包括图神经网络的基本概念、常见的图神经网络、图相似性学习和图结构学习等。

3 注意力感知的图匹配特征交互推荐算法

3.1 引言

推荐算法根据用户的历史行为对用户兴趣进行建模，为用户推荐满足需求的物品或信息。基于协同过滤的推荐是一种经典的推荐方法，它根据兴趣相似的用户对相似的物品有共同偏好的假设进行预测，从而筛选出目标用户可能感兴趣的物品。传统的协同过滤算法大多根据用户和项目的 ID 信息进行过滤，忽略了用户和物品的属性信息，但这些信息的共现性往往对推荐能否成功起着重要作用。学习复杂的属性之间特征交互对于推荐系统来说是非常必要。例如，在电影推荐场景下，男性用户更偏爱动作片和科幻片，女性用户则对爱情片更感兴趣。使用用户性别和电影类型两个属性联合起来进行推荐策略，往往会比单纯考虑单个属性信息更有效。当前特征交互模型主要存在两方面问题：一方面，使用相同的方法来建模用户与物品属性之间的交互，但不同类型的属性交互（用户属性-用户属性，用户属性-物品属性）对最终推荐的影响不同，因此无法明确显示用户和物品属性之间的交互过程；另一方面，现有研究方法主要通过用户与物品属性之间交互的隐向量直接连接或平均进行最终推荐预测，但没有考虑不同属性之间的交互权重。

针对上述问题，本章提出了一种注意力感知的图匹配特征交互推荐 AGMRec（Attention-aware Graph Matching Recommendation）算法。该算法将属性特征建模为图中的节点，特征之间的交互建模为边，把推荐过程转变为图匹配问题。

3.2 问题描述

基于特征交互的推荐算法旨在根据用户和项目属性特征学习各自的静态表示，从而给出点击率预测结果。下面给出本章所研究问题的符号定义。

3.2.1 符号描述

属性特征与属性集合。点击记录即一个样本包括用户和项目的属性特征。令 \mathbf{A}^U 和 \mathbf{A}^I 分别表示用户和项目属性， $C_i^U = \{f_1, f_2, \dots, f_p\} (f_p \in \mathbf{A}^U)$ 表示第 i 个点击记录中用户的属性集合， $C_i^I = \{f_1, f_2, \dots, f_q\} (f_q \in \mathbf{A}^I)$ 表示第 i 个点击记录中项目的属性集合。

3.2.2 问题定义

根据 3.2.1 节中定义的用户和项目属性集合, 可将所有的点击记录表示为 $D^{all} = \{X_1, X_2, \dots, X_N\}$, 其中 $X_i = \{(C_i^U, C_i^I)\}$ 代表第 i 条点击记录, N 为数据集中的总体样本数量。预测任务为: 学习一个预测函数 $F(D)$, 输入已知用户的点击记录集合 D , 预估其点击概率 $Y^{all} = \{y_1, y_2, \dots, y_N\}$, 其中 $y_i \in [0, 1]$, $i \in [1, N]$, 如公式(3-1)所示。

$$F(D^{all}) = Y^{all} \quad (3-1)$$

3.3 模型介绍

3.3.1 模型概览

在考虑不同属性之间的交互权重的基础上, 本研究提出了一个注意力感知图匹配特征交互推荐算法 AGMRec, 其结构如图 3.1 所示。

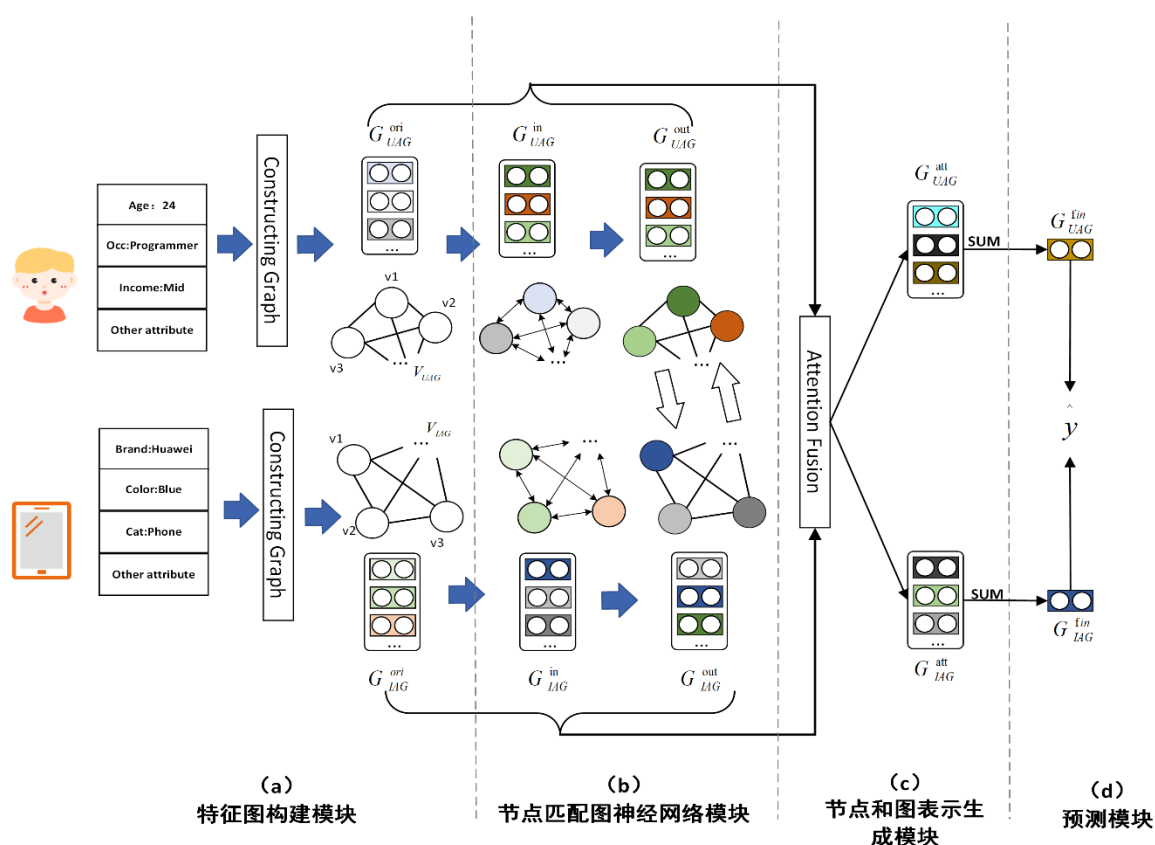


图 3.1 AGMRec 模型结构图

AGMRec 模型的创新主要包括两点:

(1) 本模型将推荐系统中的特征交互问题转换为了图匹配问题, 把用户和物品属性信息建模为两个特征图。特征图中的节点代表特征, 特征之间的交互建模为图结

构中的边。AGMRec 模型设计了一个节点匹配图神经网络模块,从内部和外部两个角度来建模特征交互,显式地建模特征之间的交互过程。

(2) 在内部节点交互部分、外部节点匹配部分和节点表示融合部分都运用了注意力机制。该机制有选择性地增强关联特征之间的交互过程,并自适应地捕捉对最终预测相对重要的阶数表示。

接下来,按照模型的层次依次介绍具体细节。

3.3.2 特征图构建模块

为了更好地表明属性之间的交互,本节将用户和物品属性分别表示为一个图结构。特征图中的一个节点表示用户或者物品的一个属性信息,节点之间的边代表一对属性的交互。首先,考虑到属性之间都需要彼此进行交互,将特征图建模为完全连通图。具体来说,用户特征图表示为 $G_{UAG} = (V_{UAG}, E_{UAG})$, 其中每个节点 $v_i \in V_{UAG}$ 对应用户的一个属性信息, E_{UAG} 是所有用户特征图中边的集合。同时,对物品属性信息进行相同的操作,获得物品特征图 $G_{IAG} = (V_{IAG}, E_{IAG})$ 。然后,创建一个用户特征图的节点属性嵌入矩阵 $W_{emb}^u \in R^{|A^U| \times d}$, 其中 d 是特征嵌入维度, $|A^U|$ 是用户属性的数量。

同样的,创建项目特征图的节点属性嵌入矩阵 $W_{emb}^i \in R^{|A^I| \times d}$ 。最终,得到用户和物品特征图的初始化节点表示 $G_{UAG}^{ori} = \{h_1^{ori}, h_2^{ori}, \dots, h_{|UAG|}^{ori}\}$, $G_{IAG}^{ori} = \{h_1^{ori}, h_2^{ori}, \dots, h_{|IAG|}^{ori}\}$ 。

3.3.3 节点匹配图神经网络模块

3.3.3.1 基于图神经网络的内部交互

AGMRec 模型通过在单个特征图的图节点之间进行信息传递完成内部交互。特征图邻接矩阵是一个二元邻接矩阵,它只能反映属性节点之间是否有关系,并不能反映特征之间的交互强度。为了更好地表达不同特征节点之间交互的差异,本节利用注意力机制来学习不同节点之间交互的权重。注意力计算如公式(3-2)所示:

$$e_{ij} = \text{Leakyrelu}(a^T [Wh_i \odot Wh_j]) \quad (3-2)$$

其中, Leakyrelu 是激活函数, \odot 是元素乘积操作, $a \in R^d$ 和 $W \in R^{d \times d}$ 为可训练参数, e_{ij} 表示节点 v_j 对于节点 v_i 的重要性。在获得不同邻居节点对中心节点的重要性后,通过 softmax 函数将重要性系数归一化处理。

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j \in N(i) \cup i} \exp(e_{ij})} \quad (3-3)$$

其中, $N(i)$ 是节点 v_i 的邻居节点集合, α_{ij} 是聚合邻居节点时的注意力权重。

$$h_i' = \sum_{j \in N(i) \cup i} \alpha_{ij} W h_j \quad (3-4)$$

$$h_i^{in} = \frac{1}{k} \sum_{j \in N(i) \cup i} \alpha_{ij}^k W^k h_j \quad (3-5)$$

在获得目标节点对于邻居节点的注意力权重后, 通过公式(3-4)计算邻居节点的初始嵌入的加权和来更新目标节点, h_i' 是经过内部特征交互后的节点表示。考虑到不同视角下节点之间的重要性不同, 公式(3-5)使用了多头注意力来计算多语义空间中的复杂依赖关系。完成上述操作后, 节点嵌入包含内部信息传递后的二阶交互特征, 表示为: $G_{UAG}^{in} = \{h_1^{in}, h_2^{in}, \dots, h_{|UAG|}^{in}\}$, $G_{LAG}^{in} = \{h_1^{in}, h_2^{in}, \dots, h_{|LAG|}^{in}\}$ 。

3.3.3.2 基于图匹配的外部交互

AGMRec 通过对用户特征图和物品特征图之间的节点匹配进行外部交互。跨级别的节点匹配更能反映节点之间交互的可能性, 例如, 男性用户更喜欢运动配件, 那么在外交互的过程中, 节点匹配的相似度应该更高并且彼此传递更多的消息。特征节点的外部交互主要分为三个主要步骤: (1) 计算每个节点的跨级别节点重要性; (2) 计算每个节点的跨级别特征图嵌入; (3) 将一个特征图的节点嵌入与另一个特征图的相关图级嵌入向量进行匹配, 生成相似性特征向量。AGMRec 使用余弦相似度函数来计算跨级节点的重要性, 计算如公式(3-6)所示。

$$\beta_{mn} = \cos(h_m^{in}, h_n^{in}), m \in V_{UAG}, n \in V_{LAG} \quad (3-6)$$

其中, β_{mn} 表示 G_{UAG} 中节点 m 与 G_{LAG} 中节点 n 的相似性得分。

接下来, 从一个特征图节点的角度来学习另一个特征图的图嵌入。具体来说, 对于一个用户特征图节点 $m \in V_{UAG}$, 相对应的注意力图嵌入向量 z_m 通过对所有 G_{LAG} 节点嵌入加权得到。同样的, 物品特征图节点 $n \in V_{LAG}$ 的注意力图嵌入 z_n 也可以由此得到。计算过程如公式(3-7)和公式(3-8)所示。

$$z_m = \sum_{n \in V_{LAG}} \beta_{mn} h_n^{in}, m \in V_{UAG} \quad (3-7)$$

$$z_n = \sum_{m \in V_{UAG}} \beta_{mn} h_m^{in}, n \in V_{LAG} \quad (3-8)$$

受多头注意力机制的启发, 引入了一个可学习的度量矩阵 $M \in R^{d' \times d}$, 从 d' 个角度考虑一个节点和其对应的图级嵌入的相似性, 从而得到外部交互匹配向量。具体计算过程如公式(3-9)和公式(3-10)所示。

$$h_m^{out} = [\cos(h_m^{in} \odot M_t, z_m \odot M_t)], t \in \{1, 2, \dots, d'\}, m \in V_{UAG} \quad (3-9)$$

$$h_n^{out} = [\cos(h_n^{in} \odot M_t, z_n \odot M_t)], t \in \{1, 2, \dots, d'\}, n \in V_{LAG} \quad (3-10)$$

在上述用户和物品特征图的节点匹配过程后, 特征图中的每个节点捕捉到与另一个图节点的高阶特征交互。所有节点更新后外部交互嵌入表示为:

$$G_{UAG}^{out} = \{h_1^{out}, h_2^{out}, \dots, h_{|UAG|}^{out}\}, G_{LAG}^{out} = \{h_1^{out}, h_2^{out}, \dots, h_{|LAG|}^{out}\}。$$

3.3.4 节点和图表示生成模块

在分别学习了用户和物品不同方面的潜在表示后, 将它们整合在一起, 以获得多方位的融合嵌入。一种策略是平等地将所有因素相加或平均, 但现实问题中并不是所有方面的因素对最终的嵌入都有相同的影响。因此, 首先构建一个两层网络自适应地学习三个方面的权重系数, 其计算方法为:

$$\gamma^{ori} = \frac{\exp(W_2^T (W_1^T \cdot h^{ori}))}{\sum_{aspect \in \{ori, in, out\}} \exp(W_2^T (W_1^T \cdot h^{aspect}))} \quad (3-11)$$

$$h^{att} = \sum_{aspect \in \{ori, in, out\}} \gamma^{aspect} h^{aspect} \quad (3-12)$$

其中, γ^{ori} 是节点初始嵌入的权重系数, W_* 是可训练参数。通过融合三个方面的特征, 得到融合后的嵌入表示为: $G_{UAG}^{att} = \{h_1^{att}, h_2^{att}, \dots, h_{|UAG|}^{att}\}, G_{LAG}^{att} = \{h_1^{att}, h_2^{att}, \dots, h_{|LAG|}^{att}\}。$

然后, 使用加和池化将图内的融合节点表示聚合成特征图的表示。最终, 特征图向量的表示如公式(3-13)所示。

$$G_{UAG}^{fin} = \sum_{i \in V_{UAG}} h_i^{att}, G_{LAG}^{fin} = \sum_{i \in V_{LAG}} h_i^{att} \quad (3-13)$$

3.3.5 预测和优化模块

在获得用户和物品的最终表示后, 利用这两个表示的相似度衡量用户对物品的偏好程度。相似度越高, 用户就越喜欢这个物品。使用一个简单的内积运算来预测用户对目标物品的偏好程度, 其计算如公式(3-14)所示:

$$\hat{y} = \sigma(G_{UAG}^{fin} \cdot G_{LAG}^{fin}) \quad (3-14)$$

其中, σ 是 Sigmoid 激活函数。

在训练阶段, 使用 Logloss 作为损失函数, 定义如公式(3-15)所示:

$$\mathcal{L} = -\frac{1}{N} \sum_{n=1}^N \left(y_n \log(\hat{y}_n) + (1 - y_n) \log(1 - \hat{y}_n) \right) \quad (3-15)$$

其中, y_n 和 \hat{y}_n 分别为用户真实点击率和模型预估的点击率, n 为训练样本, N 为训练样本总数。

3.4 算法流程

AGMRec 模型基于训练集中的用户点击记录, 学习率等超参数, 输出训练完成的模型。首先, 构建训练集, 为每个用户点击记录生成用户和物品特征图; 然后, 训练模型, 计算每个对图的点击概率和 Logloss, 模型优化器选择 Adam; 最终, 直至模型收敛或到达指定轮数时停止训练, 获得已经完成训练的推荐系统模型。详细步骤如表 3.1 所示。

时间复杂度分析: 本章节设计的算法时间消耗主要来自四个方面: 特征节点的嵌入、节点内部交互过程、节点外部交互过程、节点表示注意力融合阶段。特征节点嵌入的时间复杂度表现为 $O(Cd)$, 其中 C 为节点类别 one-hot 向量的总体长度, d 为节点嵌入维度大小。节点内部交互过程的时间复杂度表现为 $O(m^2d)$, 其中 m 为特征域的数量。节点外部交互过程的时间复杂度表现为 $O(md')$, 其中 d' 为可学习度量向量的个数。节点表示注意力融合阶段的时间复杂度为 $O(m)$ 。最终, 总体时间复杂度为 $O(Cd + m^2d + md' + m)$, 可以简化为 $O(C + m(md + d'))$ 。

表 3.1 AGMRec 模型的训练

算法 3.1 注意力感知的图匹配特征交互推荐(AGMRec)模型

输入: 所有的训练集点击记录 $D^{all} = \{X_1, X_2, \dots, X_N\}$, 学习率 l , 批次大小 $batchsize$

输出: 训练完成得到 AGMRec 模型

01: //构造训练集

02: 开始构造训练集前, 初始训练集 D 为空

03: for i in $[1, N]: \{$

04: //每条用户-物品点击记录的表示

```

05:   $X_i = \{(C_i^U, C_i^I)\}$ , 其中  $C_i^U = \{f_1, f_2, \dots, f_p\} (f_p \in \mathbf{A}^U)$ ,  $C_i^I = \{f_1, f_2, \dots, f_q\} (f_q \in \mathbf{A}^I)$ 
06:  //构建用户和物品特征图
07:   $G_{UAG} = (V_{UAG}, E_{UAG})$ ,  $G_{IAG} = (V_{IAG}, E_{IAG})$ 
08:   $V_{UAG} = C_i^U = \{f_1, f_2, \dots, f_p\}$ ,  $V_{IAG} = C_i^I = \{f_1, f_2, \dots, f_q\}$ 
09:   $E_{UAG} = \{(f_i, f_j) | i \in [1, p], j \in [1, p], i \neq j\}$   $E_{IAG} = \{(f_i, f_j) | i \in [1, q], j \in [1, q], i \neq j\}$ 
10:  构建训练集实例  $(y_i, G_{UAG}, G_{IAG})$ , 存入  $D$ 
11: }
12: //训练模型
13: for i in [1, Epoch]:{
14:   for j in [1, N/batchsize]:{
15:     从训练集  $D$  中随机取出  $batchsize$  个样本构建训练实例  $D_{batch}$ 
16:     计算  $D_{batch}$  的 Logloss
17:     利用反向传播更新模型参数}
18: }
```

在测试阶段, 构造的测试集将输入到已训练好的模型中, 即可得到预估的点击率。

3.5 实验结果与分析

本节通过与多个基线模型进行对比, 来验证所提出的 AGMRec 模型的有效性。实验所用硬件的 CPU 是 I5-10400F、24G 内存、GeForce RTX 3060(12G)显卡, 运行在 Windows10 操作系统上。以下将详细介绍实验使用的数据集、模型评价指标、基线方法和实验结果。

3.5.1 实验数据集

实验基于 Book-crossing 数据集^[57]和 MovieLens 1M 数据集^[58], 对 AGMRec 模型进行评估。实验数据集中, 训练集、验证集、测试集的比例划分为 6: 2: 2。数据集的描述如表 3.2 所示。

表 3.2 数据集描述

数据集	用户数	项目数	用户属性数	项目属性数	样本个数
Book-crossing	4873	53168	87	43157	1050834
MovieLens1M	5950	3533	30	11587	1149238

(1) Book-crossing 数据集。Book-crossing 数据集包含 4873 名用户对 53168 本图书的评分, 每个用户和图书都有若干个相关的属性信息。为了适应数据的稀疏性和保证数据的质量, 实验中将所有用户对图书的明确评分视为正例样本, 并只保留拥有超过 20 个正例的用户, 为每个用户随机选取与正例样本数量相同的负例样本。

(2) MovieLens 1M 数据集。MovieLens 1M 数据集包含用户对电影的评分, 每个数据样本都包含一个用户和一部电影的相关属性信息。实验中将大于 3 的评分视为正例, 保留超过 10 个正例的用户, 为每个正例随机采样一个负例样本。

3.5.2 评估指标

对于点击率预测算法, 通常使用 AUC (Area Under ROC Curve) 和 Logloss (Log-likelihood Loss) 两种评估指标。

(1) AUC 表示 ROC 曲线和坐标之间围成的面积, 是二元分类任务的常用指标。其范围介于[0.5, 1]之间。AUC 值越接近 1, 表示模型的分类能力越好; 若小于或等于 0.5 时, 代表模型不具有分类能力。AUC 指标的形式化表达如公式(3-16)所示。

$$AUC = \frac{\sum_i \sum_j \text{pred}(x_i^+) > \text{pred}(x_j^-)}{N_{pos} \times N_{neg}} \quad (3-16)$$

其中, x_i^+ 和 x_j^- 分别代表正样本和负样本, N_{pos} 和 N_{neg} 分别代表正样本和负样本的数量。AUC 表示在随机抽取的一对正负样本中, 模型对正例样本的预测值高于对负例样本的预测值的概率。这个概率值越大, 证明模型的分类效果越好。

(2) Logloss 表示模型预测值和真实值之间的距离, 其值越小代表模型的性能越好。损失函数的计算由公式(3-15)求得, 作为反映样本平均偏差的一个直接度量。

3.5.3 基线方法

实验中, 将 AGMRec 模型与以下几种基线方法的推荐结果进行了对比:

(1) LR^[12]: 一种线性组合原始特征的机器学习方法, 但无法捕捉特征之间的二阶交互信息。

(2) FM^[14]: 在 LR 的基础上, 通过对两两特征交互进行建模, 具有捕捉二阶交互的功能。

(3) AFM^[18]: 在 FM 的基础上, 运用注意力机制区分二阶交互的重要性。

(4) NFM^[26]: 使用特征交互池化的方式代替拼接操作, 并将深度学习网络与二阶特征交互相结合, 把特征交互的阶数扩展到了高阶。

(5) DeepFM^[29]: 将神经网络与 FM 相结合, 模型可以同时捕捉高阶交互特征和二阶交互特征。

(6) InterHAt^[35]: 在特征阶数上使用层次注意力机制, 高阶特征基于低阶特征生成, 精确量化任意阶数的特征交互影响。

(7) Fi-GNN^[37]: 首个通过图神经网络建模特征交互的模型, 将每个样本构成一个特征图, 每个节点为一个特征域, 利用边建模特征之间的交互过程。

(8) GMCf^[41]: 将用户和物品特征分别建模为两个特征图, 通过节点匹配进行特征交互, 进而学习用户和物品的向量表示。该模型是提出 AGMRec 模型的基础。

3.5.4 实验设置

本章实验均是通过 Pytorch 框架实现, 使用 Adam 优化器进行求解。为了公平起见, 实验中所有模型都是以 1024 批尺寸的方式进行学习, 特征嵌入向量维数为 64, 隐藏层单元数量为 256, 学习率为 1.e-3, L2 正则化权重为 5.e-5。

3.5.5 实验结果

实验结果如表 3.3 所示。

表 3.3 AGMRec 和基线模型的实验结果

模型类型	模型	Book-crossing		MovieLens1M	
		AUC	Logloss	AUC	Logloss
传统	LR	0.7226	0.6096	0.8704	0.4507
基于 FM	FM	0.7981	0.5449	0.8829	0.4317
	AFM	0.7838	0.5604	0.8747	0.4404
基于 DNN	NFM	0.8055	0.5426	0.9051	0.3896
	DeepFM	0.8114	0.5310	0.9044	0.3919
	InterHAt	0.8187	0.5211	0.9086	0.3836
基于 GNN	Fi-GNN	0.8217	0.5246	0.9070	0.3902
	GMCF	0.8278	0.5107	0.9083	0.3836
	AGMRec	0.8395	0.4981	0.9148	0.3728

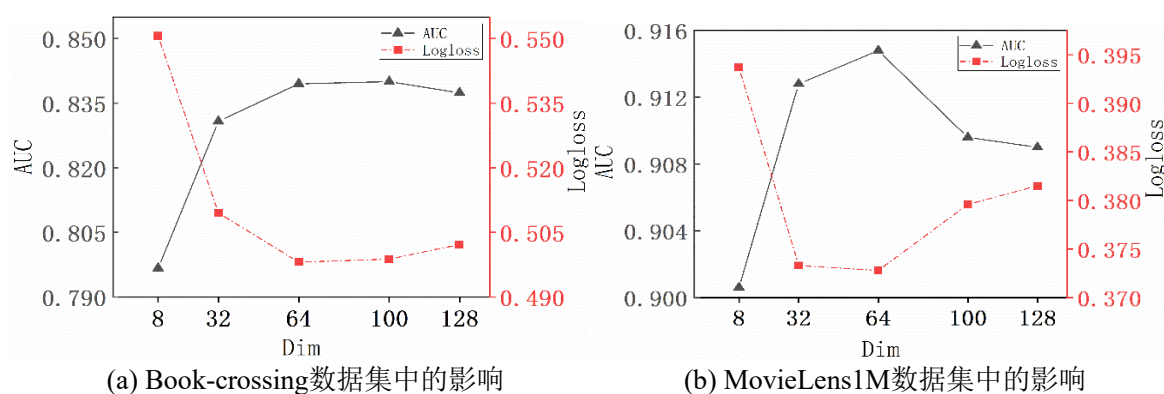
基于表 3.3 可以得到以下结论：

AGMRec 模型在 Book-crossing 数据集和 MovieLens1M 数据集上实验的评价指标都优于其它基线模型。针对 Book-crossing 数据集的实验结果中，AGMRec 模型结果的 AUC 较次优模型提高了 1.41%，Logloss 较次优模型提高了 2.5%；针对 MovieLens1M 数据集的实验结果中，AGMRec 模型结果的 AUC 较次优模型提高了 0.72%，Logloss 较之前次优模型提高了 2.82%。总体来说，AGMRec 模型相较于对比模型具有更高的性能。实验结果还显示，LR 模型的实验结果最差，说明 LR 用于单独建模特征重要性无法高效地完成点击率预测任务。FM 和 AFM 性能优于 LR，模型的成功得益于捕捉了二阶交互特征。深度特征交互模型的总体性能优于传统和二阶特征交互模型，推测其原因是神经网络的出现可以更好地挖掘特征之间的隐式高阶交互。基于 GNN 的特征交互模型以一种新的方式解释了特征交互过程，将属性特征建模为图结构来建模复杂的交互，提高了模型的性能。

3.5.6 参数对比分析

为了进一步研究超参数对 AGMRec 模型的影响，基于控制变量的方法对两个主要超参数进行实验研究。

(1) 节点嵌入维度对推荐结果的影响。一般来说，模型的嵌入维度越高，模型的代表能力就越强。然而，嵌入维度过多不仅会导致推荐成本过高，而且也可能导致模型的过拟合。Book-crossing 和 MovieLens1M 数据集上的节点嵌入结果如图 3.2 所示。



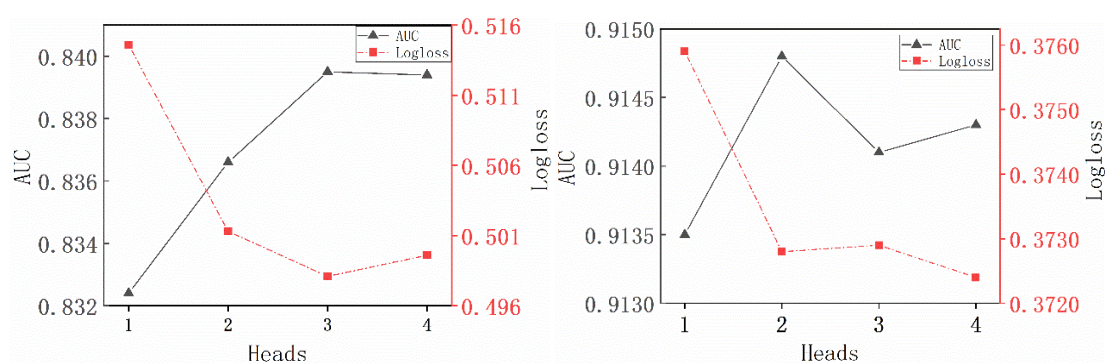
(a) Book-crossing数据集中的影响

(b) MovieLens1M数据集中的影响

图 3.2 不同嵌入维度对AUC和Logloss的影响

从图 3.2 可以看出, 在 Book-crossing 数据集和 MovieLens1M 数据集上, AUC 指标随着节点嵌入维度的增加首先上升, 当维度为 64 时达到峰值然后开始下降; Logloss 指标随着节点嵌入维度的增加首先下降, 当维度为 64 时达到最优随后开始上升。实验结果说明, 64 的节点嵌入维度能够代表足够多的信息, 从而获得良好的推荐效果。

(2) 注意力头数对 AUC 和 Logloss 的影响。多头注意机制可以捕捉到不同语义空间的节点之间的关系, 并在不同层次上聚合有益的交互特征。增加注意力头数往往能提高模型的性能, 但是过多的注意力头数可能导致过拟合。基于 AGMRec 模型, 针对不同的注意力头数开展了实验研究。当注意力头数设置为 1 时, 表示多头注意力机制被有效地去除。注意力头数对 AUC 和 Logloss 影响的实验结果如图 3.3 所示。



(a) Book-crossing数据集中的影响

(b) MovieLens1M数据集中的影响

图 3.3 不同注意力头数对AUC和Logloss的影响

从图 3.3 可以看出, MovieLens1M 数据集使用两个注意力头数时表现最好, 而 Book-crossing 数据集是三个注意力头数的结果最优。

3.5.7 消融实验

AGMRec 模型主要用了三个方面的注意力来捕捉特征之间的交互，基于此，创建了三个变体模型来进行消融实验。三个变体模型分别为：

AGMRec_in：除去内部注意力节点消息传递的 AGMRec；

AGMRec_out：除去外部注意力节点匹配的 AGMRec；

AGMRec_att：除去注意力融合节点表示的 AGMRec。

表 3.4 为 AGMRec 模型与三个变体模型的性能比较结果。

表 3.4 AGMRec 与三个变体的性能比较				
数据集	Book-crossing		MovieLens1M	
	AUC	Logloss	AUC	Logloss
模型				
AGMRec_in	0.8317	0.5063	0.9110	0.3772
AGMRec_out	0.8369	0.5061	0.9127	0.3753
AGMRec_att	0.8313	0.5137	0.9099	0.3795
AGMRec	0.8395	0.4981	0.9148	0.3728

从表 3.4 中可以看出，去除任何一个模块后，AGMRec 模型的性能都会下降。AGMRec_in 与 AGMRec 相比，虽然利用简单的元素乘积可以捕捉特征节点之间的交互特征，但是特征之间的相互作用并不具有同等的重要性；在传递信息的过程中考虑边的权重可以避免噪声对目标节点的影响，能够获得更好的效果。AGMRec_out 与 AGMRec 相比，引入的跨级注意力图匹配方法从一个全新角度建模特征交互过程，可以对有益的高阶特征交互进行有针对性地建模，提高模型的准确性。本章的目标是通过不同级别的节点特征生成最终的节点表示，进而完成推荐预测。通过 AGMRec_att 与 AGMRec 实验结果对比，可以发现一阶、二阶和高阶的特征节点嵌入表示的直接相加或拼接，并没有考虑到每个方面的特征对最终节点表征的重要性。通过注意力融合机制学习不同方面的重要性权重可以得到更为准确的节点表示，并提高算法的预测性能。

3.6本章小结

本章分析了目前特征交互算法所存在的问题，将传统点到点的建模范式扩展到了图到图，提出了一种基于注意力感知的图匹配特征交互推荐算法。首先，特征图构建模块将用户和物品属性信息，建模为用户特征图和物品特征图；其次，运用节点匹

配图神经网络模块内部和外部两个级别的融合过程，获得交互过后的节点表示；接下来，通过注意力机制进行三方面的节点特征融合；最后，预测模块通过用户和物品最终向量预估点击率。实验结果表明 AGMRec 模型是有效的，在两个公开数据集上推荐效果优于基线模型。未来工作中，考虑加入更多的辅助信息（物品图片和评论）进行多模态个性化推荐，以及使用不同的特征融合机制来更为精准地预估点击率结果。

4 分层双级别图融合特征交互推荐算法

4.1 引言

个性化推荐系统由于可以解决在线信息过载问题,已经大量应用在了广告、社交网络和电子商务等领域之中。其核心思想是根据用户的历史购买或点击记录来预估用户对项目的偏好。它能够有效利用众多的用户画像、项目属性和上下文信息进行特征交互,特征交互建模已经成为推荐系统的一个重要范式。

大数据中越来越多的属性信息可以被利用作为决策信息。不仅是第三章提到的用户和物品属性信息,上下文信息对于推荐系统来说也是极为重要。如何建模上下文信息、用户、物品属性信息的交互成为现阶段研究的重心。同时,现存模型也存在着两方面主要问题。一方面,并不是所有的特征组合都会产生积极的交互增益,一些不必要的特征交互在训练过程中往往带来噪音,影响最终的预测结果;另一方面,现存模型大多是不可解释和隐式建模交互的。

基于上述的问题,本章延伸了第三章的 AGMRec 模型,提出了分层双级别图融合特征交互推荐模型 HDGFI (Hierarchical Dual-level Graph Feature Interaction)。该模型将推荐中点击率预估问题转化为图结构的学习问题,在不同交互层中自适应地选择节点连接,并基于每层的图表示向量显式地生成最终的预估结果。具体来说,每个特征域视为特征图中的节点,并通过边来建模特征之间的交互。本章设计了一个全新的交互层,从局部和全局的角度来形成有意义的高阶特征表示。在每个交互层中,每个特征节点可以通过显式选择最相关的邻居节点进行交互,同时利用压缩激励网络进行动态的特征重要性选择。最终,通过堆叠一定数目的交互层,形成高阶的特征交互表示。

4.2 问题描述

4.2.1 相关概念与表示

定义 4-1 用户、物品集合和点击记录。数据集中包含 M 个用户和 N 个物品。用

户集合表示为 $U = \{u_1, u_2, \dots, u_M\}$ ，物品集合表示为 $V = \{v_1, v_2, \dots, v_N\}$ 。用户-物品点击记录可以表示为 $Y_{M \times N}$ ，其中 $y_{uv} = 1$ 表示用户 u 和物品 v 产生过点击记录，否则 $y_{uv} = 0$ 。

定义 4-2 属性特征域。数据集中包含 J 个用户特征域， K 个物品特征域， F 个上下文特征域，分别表示为 $A = \{A^1, A^2, \dots, A^J\}$ ， $B = \{B^1, B^2, \dots, B^K\}$ ， $C = \{C^1, C^2, \dots, C^F\}$ 。每个用户和物品都与一个属性列表相关联，表示为 $A_u \in A$ ， $B_v \in B$ 。用户对物品进行点击时还关联着一个上下文信息列表，表示为 $C_{uv} \in C$ 。

4.2.2 问题定义

根据 4.2.1 节给出的定义，可将一个点击记录表示为 $x = [u, v, A_u, B_v, C_{uv}]$ ，其中 u 和 v 代表用户和物品的 ID， A_u 和 B_v 代表当前用户和物品的属性列表， C_{uv} 为当前用户点击物品时的上下文信息列表。推荐任务是设计一个预测模型，给定一个输入样本 x ，输出目标用户对候选物品的点击预测概率 \hat{y} 。

4.3 模型介绍

4.3.1 模型概览

本研究提出了一个分层双级别图融合特征交互推荐模型 HDGFI，模型结构如图 4.1 所示。

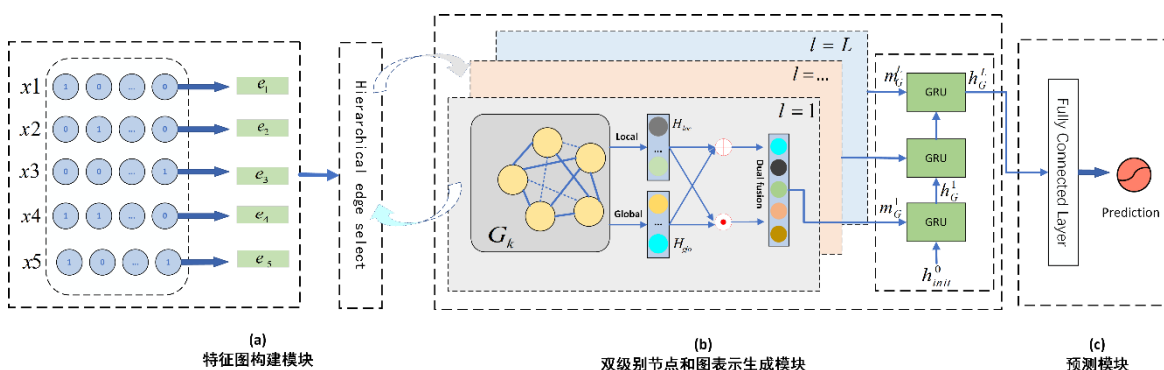


图 4.1 HDGFI 模型结构图

HDGFI 模型的创新主要分为两点：

(1) 将推荐系统中的特征交互问题转换为了图分类问题, 把特征域建模为节点, 特征之间的交互建模为图结构中的边。同时, 模型显式表达特征之间的交互情况, 通过堆叠 GNN 层捕捉高阶的特征交互, 实现了特征交互的可解释性。

(2) 通过两个级别过滤和筛选识别出交互增益强的特征交互和全局重要特征。从局部和全局两个角度动态捕捉特征和特征交互的重要性, 并使用一个融合模块合并两个方面的节点表示, 更细粒度地捕捉特征之间的重要性。

以下将逐层介绍 HDGFI 模型的实现细节。

4.3.2 特征图构建模块

4.3.2.1 特征图节点嵌入

在特征交互推荐任务中, 输入的原始特征是多字段的稀疏特征。例如, 在电影推荐中有若干特征域, 比如{语言: 英语, 导演: 克里斯托弗·诺兰, 电影类型: 动作电影、科幻电影...}。这些多域特征不能直接输入到基于 DNN 的模型中。传统的方法是将特征字段编码为二进制向量, 表示为:

$$x = [x_1, x_2, \dots, x_m] = [\underbrace{1, 0, \dots, 0}_{field_1}, \dots, \underbrace{1, \dots, 1, 0}_{field_m}] \quad (4-1)$$

其中, x 是一个含有 m 个特征域的输入实例, x_m 是第 m 个特征域的二进制编码表示, m 是特征域的个数。由于特征的总数很大, 使得二进制编码向量高维且稀疏, 这导致计算效率不高。通过特征嵌入层, 可以将二进制编码向量转换为密集的低维向量。利用公式(4-2)实现这种转换。

$$e_i = W_i^{emb} x_i \quad (4-2)$$

其中, W_i^{emb} 是仅含一个值的特征域的嵌入矩阵, x_i 是一个 one-hot 向量。对于电影类型等多值型特征域, 进一步扩展公式(4-2)为公式(4-3), 使用相应特征嵌入向量的平均值来表示多值特征字段。

$$e_j = \frac{1}{q} W_j^{emb} x_j \quad (4-3)$$

其中, W_j^{emb} 是多值特征域 j 的嵌入矩阵, q 是域内特征的数量, x_j 是 mulit-hot 向量。

从而得到输入实例中所有特征域的嵌入向量。

$$E=[e_1, e_2, \dots, e_m] \quad (4-4)$$

其中, $e_m \in R^d$ 是第 m 个特征域的嵌入表示, d 是嵌入层的维度。将特征域的嵌入视为特征图中节点的表示, 同时, 这其中还包含着关于特征的一阶交互信息。

4.3.2.2 分层邻接矩阵选择层

当前主要通过将特征转换为节点以及特征之间的交互转换为边的方法, 来构建一个完全连通的特征图。然而, 并不是所有的交互都有助于提升最终的预测效果, 有些交互则会引发交互噪声。为了捕获具有强交互增益的交互对, 并避免不必要的特征交互产生的噪声, 本研究设计了学习每个交互层的图结构来预测特征节点之间连接的策略。传统的邻接矩阵是离散的, 只会出现连接或者不连接两种情况, 这使得梯度难以反向传播。为了解决这个问题, 本设计基于 Sparse Transformer 的思想为每个层生成 Top-k 有益邻接矩阵 A^l , 其中 A^l_{ij} 表示节点 i 和节点 j 在 l 层的连接概率。基于 Top-k 选择规则, 在分层边缘选择的架构中, 只有贡献较大的元素被赋予概率。分层边选择框架如图 4.2 所示。

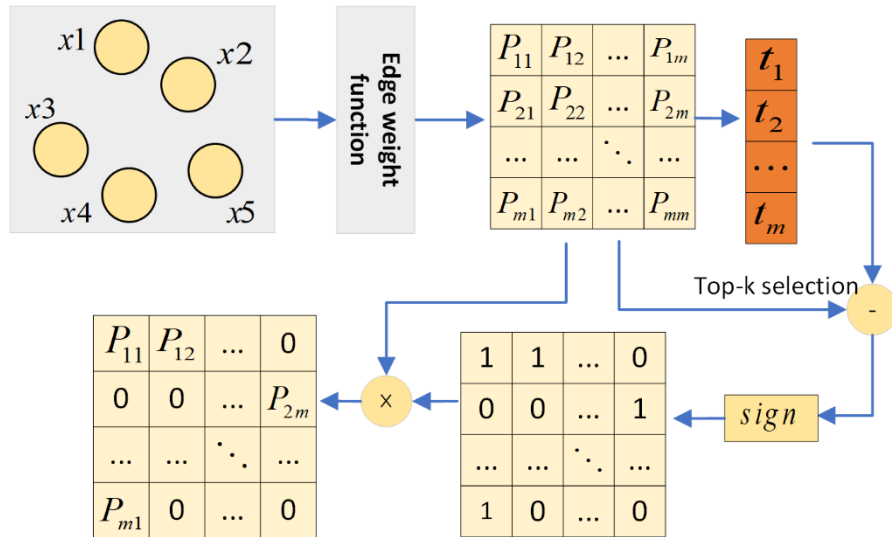


图 4.2 分层边选择框架图

首先, 通过两层全连接网络将节点对之间的相似性表示成一个注意力分数矩阵 P 。

$$P^l_{ij} = \sigma_2(W_2^s \sigma_1(W_1^s (e_i^l \odot e_j^l) + b_1) + b_2) \quad (4-5)$$

其中 W_* 和 b_* 是可训练参数, \odot 是元素乘积操作, σ_1 和 σ_2 分别是 Leakyrelu 和 Sigmoid 激活函数。

接下来, 通过掩盖操作 $M(\cdot, \cdot)$ 对注意力得分矩阵进行 Top-k 选择。通过这种方法, 最重要的 k 个特征交互被保留下来, 而其它相对不重要的信息被舍弃。通过公式 (4-6) 可以求解层次结构的有益邻接矩阵。

$$A_{ij}^l = M(P^l, k)_{ij} = \begin{cases} P_{ij}^l & P_{ij}^l \geq t_i \\ 0 & P_{ij}^l < t_i \end{cases} \quad (4-6)$$

其中, t_i 是注意力得分矩阵 P 中第 i 行中第 t 大的值, k 是控制每层邻居采样数量的超参数。模型保留得分矩阵中前 k 大的值, 将其余值设为 0, 也就意味着当前层中不考虑这两个特征之间的交互作用。

4.3.3 双级别节点和图表示生成模块

在完成特征图构建的基础上, 进一步实现节点间的特征交互。双级节点表征生成的架构如图 4.3 所示。

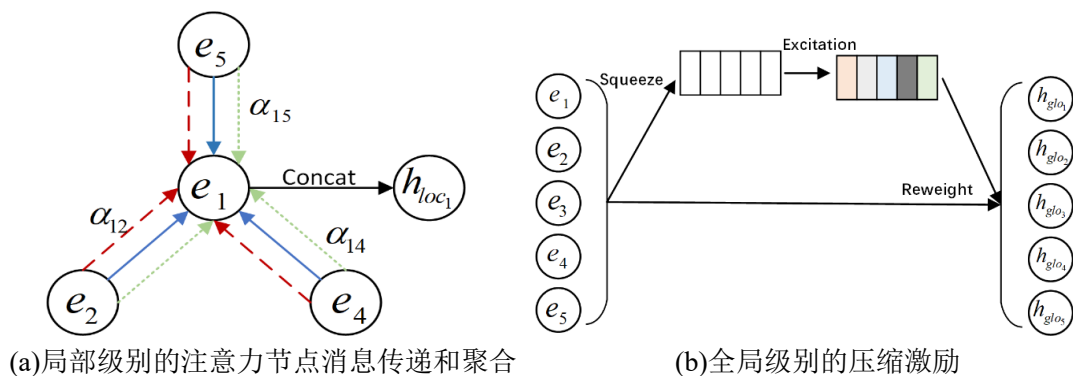


图 4.3 双级节点表征生成的架构

4.3.3.1 局部级别消息传递和聚合

两个有益特征节点之间的交互由一条边来表示, 通过一个图注意力网络逐层捕捉特征之间的高阶交互, 实现它们之间的消息传递, 其过程如图 4.3(a)所示。与传统的全连接网络有所不同, 局部级别的注意力节点消息传递过程中, 只有当前层目标节点的采样邻居才会进行消息传递。注意力系数由公式(4-7)求解。

$$c_{ij}^{(l)} = \text{Leakyrelu}\left(a^T \left(e_i^l \odot e_j^l\right)\right) \quad (4-7)$$

其中, $c_{ij}^{(l)}$ 表示在 l 层节点 j 对节点 i 的重要程度, Leakyrelu 是激活函数, $a \in R^d$ 是一个权重向量。在获得不同邻居节点对中心节点的重要性后, 通过 softmax 函数将重要性系数归一化处理。归一化处理基于公式(4-8)实现。

$$\alpha_{ij} = \frac{\exp(c_{ij}^l)}{\sum_{j \in \text{Nei}\{i\}} \exp(c_{ij}^l)} \quad (4-8)$$

其中, $\text{Nei}(i)$ 表示节点 i 的有益交互邻居集合, α_{ij} 是目标节点 i 聚合邻居节点 j 时的注意力权重。在获得中心节点对邻近节点的关注权重后, 使用多头注意力融合机制来更新目标节点的表示。计算方法如公式(4-9)所示。

$$h_{loc_i}^l = \parallel_{u=1}^U \sigma\left(\sum_{j \in N_i} A_{ij}^l \alpha_{ij}^u W^u \left(e_i^l \odot e_j^l\right)\right) \quad (4-9)$$

其中, \parallel 表示向量拼接操作, u 是注意力头数, α_{ij}^u 是第 u 个注意力头上的注意力得分, W^* 是线性变换矩阵。经过上述操作, 可以得到节点的局部级别的表示:

$$H_{loc}^l = \left[h_{loc_1}^l, h_{loc_2}^l, \dots, h_{loc_m}^l\right]。$$

4.3.3.2 全局级别压缩激励

局部级别的节点表示学习模块主要关注节点对之间的相互作用, 但没有关注到节点在全图中的重要性。不同特征对目标任务有着不同的重要性, 例如, 在电影推荐中, 性别、年龄和电影体裁往往比用户的职业和地区更重要。本节希望能动态地增加重要特征的权重, 同时减少不必要特征的权重。压缩激励网络在图像分类任务中取得了巨大的成功, 它可以捕捉通道之间的相互依赖关系, 以提高网络的表示能力。为了保留先前学习的组合特征并动态捕捉特征之间的相对重要性, 本节在全局层面设计了一个用于残差连接的压缩激励模块, 如图 4.3(b)所示。

第一步为压缩 (Squeeze) 操作。运用池化方法将各个节点特征 $[e_1^l, e_2^l, \dots, e_m^l]$ 压缩为一个统计向量 $S = [s_1, s_2, \dots, s_m]$, 池化可以选择最大化、平均和加和等操作。本文选择了全局平均池化, 具体计算如公式(4-10)所示。

$$s_i = \frac{1}{d'} \sum_{t=1}^{d'} e_i^t \quad (4-10)$$

其中 s_i 表示第 i 个特征的全局信息表示。

第二步为激励 (Excitation) 操作。这一步使用了两个全连接层来计算统计向量每一维度的权重, 通过一个缩减率超参数来先压缩维度再扩张维度。具体的计算如公式 (4-11) 所示。

$$att_{glo} = \sigma_3 \left(W_2^g \sigma_3 \left(W_1^g S \right) \right) \quad (4-11)$$

其中, att_{glo} 是全局级别的注意力权重, $W_1^g \in \mathbb{R}^{\frac{m \times m}{r}}$, $W_2^g \in \mathbb{R}^{\frac{m}{r} \times m}$ 为训练参数, σ_3 是 Tanh 激活函数。

第三步为重加权 (Re-weight)。将原始的特征表示 $[e_1^l, e_2^l, \dots, e_m^l]$ 和相应权重 att_{glo} 相乘得到新的节点向量 H_{glo}^l , 计算公式如公式 (4-12) 所示。

$$H_{glo}^l = att_{glo} \odot E^l = [att_{glo_1} \cdot e_1^l, att_{glo_2} \cdot e_2^l, \dots, att_{glo_m} \cdot e_m^l] \quad (4-12)$$

4.3.3.3 双级别节点嵌入融合

一旦获得了局部和全局级别的节点表征, 便可使用双线性交叉聚合函数来更新节点的表示, 如公式 (4-13) 所示。

$$e_i^l = W_\phi^l [h_{loc_i} \oplus h_{glo_i}, h_{loc_i} \odot h_{glo_i}] \quad (4-13)$$

其中, \oplus 和 \odot 分别代表元素加和和元素乘积操作, W_ϕ^l 是可训练参数。元素相乘的操作可以从相似的特征中传递更多的信息, 而相加的操作可以突出具有较大加和值的特征。

4.3.3.4 特征图表示生成

经过上述操作后, 节点的表示方法已经得到了更新。换句话说, 每个特征节点都是邻域感知的。进一步, 我们设计了一个图表示读出操作, 动态地捕捉交互后每一层的特征图嵌入。该读出操作由所有时间步骤共享, 对特征图中所有节点的向量进行平均池化来得到全局图的向量, 计算公式如下。

$$m_G^l = \frac{1}{m} \sum_{i=1}^m e_i^l \quad (4-14)$$

然后, 将当前层的全局表示向量 m_G^l 和上一层读出的特征图向量 h_G^{l-1} 共同输入到 GRU 中, 来更新获得当前层特征图向量 h_G^l 。

$$h_G^l = GRU(m_G^l, h_G^{l-1}) \quad (4-15)$$

初始的特征图向量 h_G^0 是初始节点嵌入的加和, 通过读出操作, 图的特征在每个时间步骤中被更新。

4.3.4 预测和训练模块

在双级别节点和图表示生成模块之后, 可以得到最后一层的特征图向量。本节采用以 θ 为参数的全连接层和 Sigmoid 激活函数来获得最终的预测, 具体如下。

$$\hat{y} = Sigmoid(\theta^T \cdot h_G^L) \quad (4-16)$$

模型使用 Logloss 作为损失函数, 具体计算如式(4-17)所示。

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \left(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right) \quad (4-17)$$

其中, N 是训练样本的数量, y_i 和 \hat{y}_i 分别为索引为 i 的样本的真实标签和预测标签。模型的最优参数通过 Adam 最小化损失函数求解。

4.4 算法流程

基于上述对 HDGFI 模型的描述, 接下来介绍 HDGFI 模型的训练过程。该训练输入所有训练集中的用户点击记录、学习率等超参数, 输出训练完成的 HDGFI 模型。首先, 为训练数据中每个点击实例构建一个特征图, 此特征图中只有特征节点集合, 没有边集合, 边的连接情况在训练阶段学习; 然后, 分层计算特征图的邻接矩阵并进行局部和全局的特征节点的交互, 得到点击预测概率; 最后, 计算出模型整体损失 Logloss 值, 使用 Adam 优化器进行参数更新, 直至模型收敛或到达指定轮数时停止训练。训练详细步骤如表 4.1 所示。在测试阶段, 构造的测试集将输入到已训练好的

模型中，即可得到预估的点击率。

时间复杂度分析：本章节设计的算法时间消耗主要来自四个方面：特征节点的嵌入、分层邻接矩阵构造、两个级别的节点表示融合、图表示生成。特征节点嵌入的时间复杂度表现为 $O(Cd)$ ，其中 C 为节点类别 one-hot 向量的总体长度， d 为节点嵌入维度大小。分层邻接矩阵构造的时间复杂度表现为 $O(L(m^2d))$ ，其中 L 为图神经网络迭代次数， m 为特征域的数量。两个级别的节点表示融合时间复杂度表现为 $O(Lm(md + m^3/r))$ ，其中 L 为迭代层数， r 为全局衰减比例。图表示生成的时间复杂度为 $O(L(md^2))$ 。最终，总体时间复杂度为 $O(Cd + L(m^2d) + Lm(md + m^3/r) + L(md^2))$ ，可以简化为 $O(C + m^2(d + m^2/r))$ 。

表 4.1 HDGFI模型的训练

算法 4.1 分层双级别图融合特征交互推荐(HDGFI)模型

输入：数据集中所有点击实例 $D^{all} = \{X_1, X_2, \dots, X_N\}$ ，学习率 l ，批次大小 $batchsize$

输出：训练完成的 HDGFI 模型

```

01: //构造训练集
02: 开始构造训练集前，初始训练集  $D$  为空
03: for  $i$  in  $[1, N]: \{$ 
04:   //每条点击实例的符号化表示
05:    $x_i = [u, v, A_u, B_v, C_{uv}]$ ，其中  $A_u \in A$ ， $B_v \in B$ ， $C_{uv} \in C$ 
06:   //构建特征图
07:    $G_i = (V_i, E_i)$ 
08:    $V_i = \{u, v, A_u, B_v, C_{uv}\}$ 
09:    $E_i = \emptyset$  //在训练中学习
10:   构建训练集实例  $(y_i, G_i)$ ，存入  $D$ 
11: }
12: //训练模型
13: for  $i$  in  $[1, \text{Epoch}]: \{$ 
14:   for  $j$  in  $[1, N/\text{batchsize}]: \{$ 

```

-
-
- ```

15: 从训练集 D 中随机取出 $batchsize$ 个样本构建训练实例 D_{batch}
16: 计算 D_{batch} 的 Logloss
17: 利用反向传播更新模型参数}
18: }
```
- 

## 4.5 实验结果与分析

本节通过与多个基线模型进行对比,来验证提出的个性化推荐模型 HDGFI 的有效性和合理性。HDGFI 模型实现由 python 语言编写,实验使用硬性条件是:CPU 型号是 I5-12400F、32G 内存, GeForce RTX 3060(12G)显卡,运行在 Ubuntu 20.04 操作系统上。下面将详细介绍实验使用的数据集、模型评价指标、基线方法和实验结果等。

### 4.5.1 实验数据集

实验选择了 KKBox<sup>[59]</sup>、Frappe<sup>[60]</sup>、MovieLens 1M<sup>[58]</sup>三个公开数据集。实验过程中,三个数据集中,训练集、验证集、测试集的划分比例为 6: 2: 2。三个数据集的详细信息如表 4.2 所示。

表 4.2 数据集描述

| 数据集          | 特征域个数 | 特征总数  | 样本个数    |
|--------------|-------|-------|---------|
| KKBox        | 13    | 92247 | 7377418 |
| Frappe       | 10    | 5382  | 288609  |
| MovieLens 1M | 10    | 22100 | 1149238 |

(1) KKBox 数据集。KKBox 数据集是一款专业的数字音乐信息服务软件,每个用户可以在短时间内找到他们喜欢的音乐。数据集包含 13 个特征域信息,例如,各歌曲的 ID、词曲作者姓名、用户的年龄等。

(2) Frappe 数据集。Frappe 数据集是一个上下文感知应用程序发现工具。Frappe 数据集中每个记录包含用户的 ID 和 APP 的 ID,以及天气、城市等 8 个上下文信息。

(3) MovieLens 1M 数据集。MovieLens 1M 数据集包含用户对电影的评分,每个数据样本都包含一个用户和一部电影的相关属性信息。实验中将大于 3 的评分视为正例,保留超过 10 个正例的用户,为每个正例随机采样一个负例样本。

### 4.5.2 评估指标

对于点击率预测算法，通常使用 AUC (Area Under ROC Curve) 和 Logloss (Log-likelihood Loss) 两种推荐指标。

(1) AUC 表示 ROC 曲线和坐标之间围成的面积，是二元分类任务的常用指标。其范围介于[0.5, 1]之间，AUC 值越接近 1 代表模型的分类能力越好，若小于或等于 0.5 时代表模型不具有分类能力。具体公式如下所示。

$$AUC = \frac{\sum_i \sum_j pred(x_i^+) > pred(x_j^-)}{N_{pos} \times N_{neg}} \quad (4-18)$$

公式(4-18)是 AUC 指标的形式化表达，其中  $x_i^+$  和  $x_j^-$  分别代表正样本和负样本， $N_{pos}$  和  $N_{neg}$  分别代表正样本和负样本的数量。AUC 表示在随机抽取的一对正负样本中，模型对正例样本的预测值高于对负例样本的预测值的概率。这个概率值越大，说明模型的分类效果越好。

(2) Logloss 表示预测分数与真实标签之间的距离，其值越小代表模型的性能越好。Logloss 的计算公式和上节预测模块的损失函数相同，在本实验中作为反映样本平均偏差的一个直接度量。

### 4.5.3 基线方法

本研究将 HDGFI 模型的推荐结果与 LR<sup>[12]</sup>、FM<sup>[14]</sup>、FFM<sup>[15]</sup>、FmFM<sup>[17]</sup>、NFM<sup>[26]</sup>、FiBiNet<sup>[31]</sup>、GraphFM<sup>[40]</sup>等主要基线方法的结果进行比较。其中，FFM 是在 FM 的基础上增加了特征域的概念；FmFM 在 FM 基础上为每对特征增加一个权重值；FiBiNet 使用压缩激励机制动态学习特征重要性；GraphFM 使用图神经网络克服 FM 的缺陷，并从有益交互学习的角度建模特征交互。其余基线模型见前文 3.5.3 节的介绍。

### 4.5.4 实验设置

本实验均是基于 Pytorch 框架实现，使用 Adam 优化器进行求解。为了公平起见，所有模型都是以 1024 批尺寸的方式进行学习，特征嵌入向量维数为 16，隐藏层单元

数量为 32, L2 正则化权重为  $1.e-4$ 。在 KKBox 和 MovieLens 1M 中, 学习率为  $1.e-3$ , 在 Frappe 数据集中, 学习率为  $5.e-3$ 。对于 AutoInt, Fi-GNN, GraphFM 模型, 迭代层数都设置为三层。

#### 4.5.5 实验结果

表 4.3 HDGFI和基线模型的性能比较

| 模型类型   | 模型       | KKBox          |                | Frappe         |                | MovieLens 1M   |                |
|--------|----------|----------------|----------------|----------------|----------------|----------------|----------------|
|        |          | AUC            | Logloss        | AUC            | Logloss        | AUC            | Logloss        |
| 传统     | LR       | 0.76647        | 0.57593        | 0.93565        | 0.28721        | 0.86949        | 0.43775        |
| 基于 FM  | FM       | 0.78961        | 0.55487        | 0.96571        | 0.20912        | 0.89104        | 0.42229        |
|        | AFM      | 0.79868        | 0.54858        | 0.96534        | 0.21947        | 0.88224        | 0.42861        |
|        | FFM      | 0.79758        | 0.54323        | 0.96871        | 0.19901        | 0.89563        | 0.40881        |
|        | FmFM     | 0.80591        | 0.53465        | 0.96564        | 0.21608        | 0.90181        | 0.39455        |
| 基于 DNN | NFM      | 0.80979        | 0.53088        | 0.97283        | 0.20717        | 0.89975        | 0.40351        |
|        | DeepFM   | 0.81439        | 0.52556        | 0.97551        | 0.18532        | 0.90617        | 0.38856        |
|        | FiBiNet  | 0.81783        | 0.52207        | 0.97554        | 0.18061        | 0.90628        | 0.39021        |
|        | InterHAt | 0.81478        | 0.52477        | 0.97447        | 0.19099        | 0.90616        | 0.38707        |
| 基于 GNN | Fi-GNN   | 0.81831        | 0.52033        | 0.97541        | 0.18431        | 0.90668        | 0.38755        |
|        | GraphFM  | 0.82013        | 0.51872        | 0.9764         | 0.17824        | 0.90782        | 0.38378        |
|        | HDGFI    | <b>0.82278</b> | <b>0.51555</b> | <b>0.97894</b> | <b>0.16495</b> | <b>0.91113</b> | <b>0.37871</b> |

各模型在三个公开数据集上实验评价指标如表 4.3 所示。从表中可以看出, 所提出的 HDGFI 模型实验结果的评价指标均优于对比的基线方法。HDGFI 模型的 AUC 指标较次优模型 GraphFM 模型在 KKBox、Frappe、MovieLens 1M 数据集上分别提升了 3.23%、2.6%和 3.64%, 并且在 Logloss 指标上也优于其它的基线模型。总体来看, 基于 DNN 和基于 GNN 为代表的高阶交互模型效果要优于仅关注一阶、二阶组合特征的模型, 特征之间复杂的高阶组合可以为最终的预测带来超越本身信息的交互增益; 基于 GNN 的特征交互模型整体效果要好于基于 DNN 的模型。证明基于 DNN 进行特征交互的模型是以一种隐式的方式学习特征交互, 而图结构可以以一种显式的方式建模特征交互过程, 具有良好的可解释性。与同为基于 GNN 的模型 Fi-GNN 相比, HDGFI 没有将特征图建模为一个完全连通图, 而是运用了分层邻接矩阵学习策略, 显式地保留 Top-K 个重要特征交互。这样可以减少不必要的特征交互所带来的噪声, 从而更集中地关注重要交互。与 GraphFM 模型相比, HDGFI 从局部和

全局两个角度来考虑特征的重要性,实验结果证明 HDGFI 的推荐效果优于 GraphFM。

#### 4.5.6 参数对比分析

为了进一步研究超参数对 HDGFI 模型的影响,使用控制变量方法对两个主要超参数进行实验。

(1) 各层的邻居采样个数超参数实验。HDGFI 模型将第一层的采样个数固定为特征域的个数,这意味着在第一层建立了一个全连通特征图。各个特征出现在一个输入实例中,通过全连通方式,模型尽可能地捕捉每个节点对之间的交互关系。在第二层和第三层选择了不同的邻居采样个数。采样个数超参数结果如图 4.4 所示。其中, $k_2$  和  $k_3$  分别表示第二层和第三层目标特征节点邻居采样的个数。

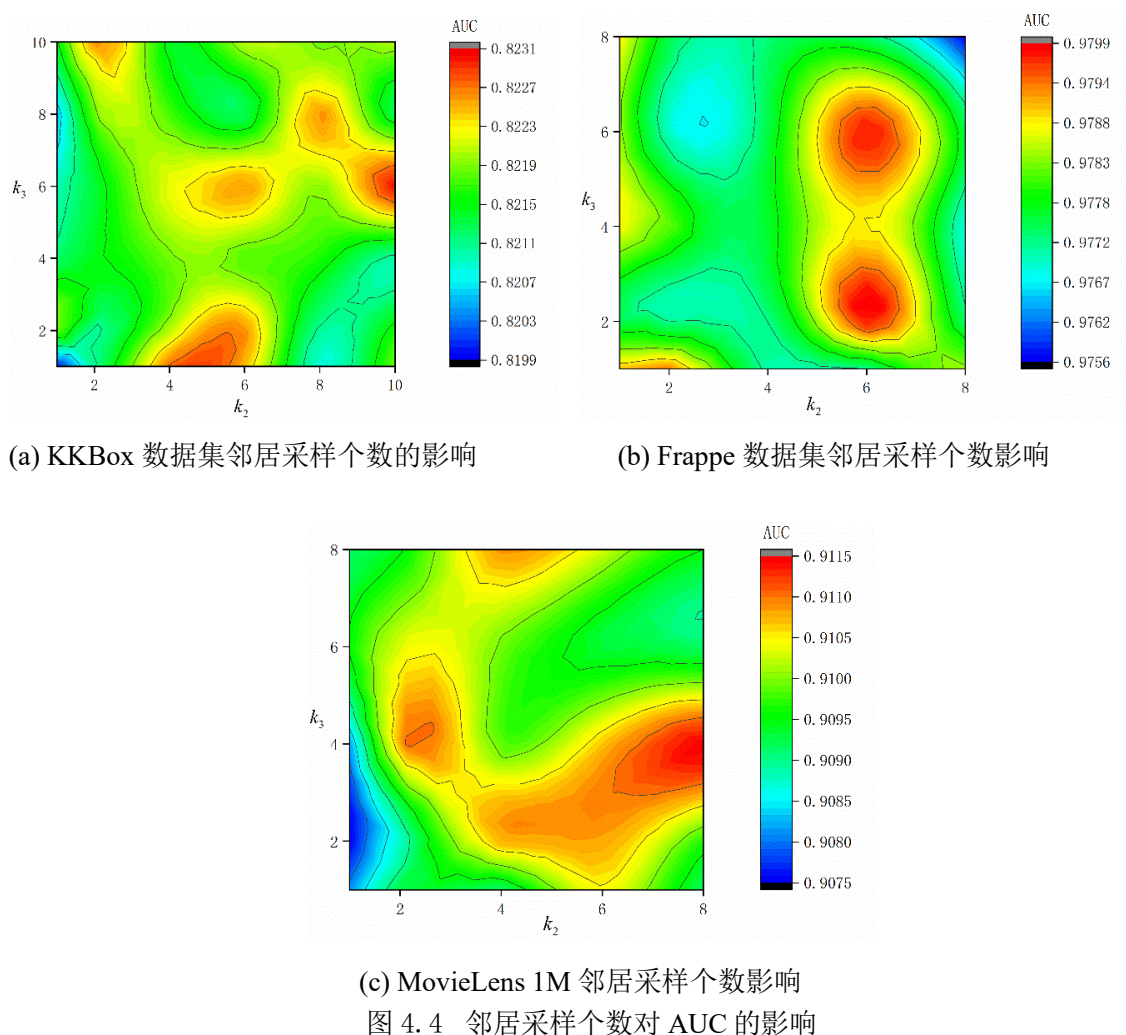


图 4.4 邻居采样个数对 AUC 的影响

图 4.4 (a) 显示的是在 KKBox 数据集上第二层和第三层不同的邻居采样个数对 AUC 的影响。可以看出在  $k_2=10$ ,  $k_3=6$  时性能达到最优值；当第二层和第三层都选择数量为 1 的邻居节点时效果最差。图 4.4 (b) 显示的是 Frappe 数据集上的结果，可以看出当  $k_2=6$ ,  $k_3=2$  时效果达到峰值。图 4.4 (c) 显示的是 MovieLens 1M 数据集上的结果，可以看出在  $k_2=8$ ,  $k_3=4$  时获得最好的性能。总体来说， $k_2$  大于  $k_3$  时效果要好于  $k_2$  小于  $k_3$ ，意味着需要逐层减少邻居采样的个数，才能更好地捕捉特征之间的有益交互。

(2) 全局衰减比率超参实验。针对不同全局衰减比率对模型性能影响，开展了实验研究，结果如图 4.5 所示。

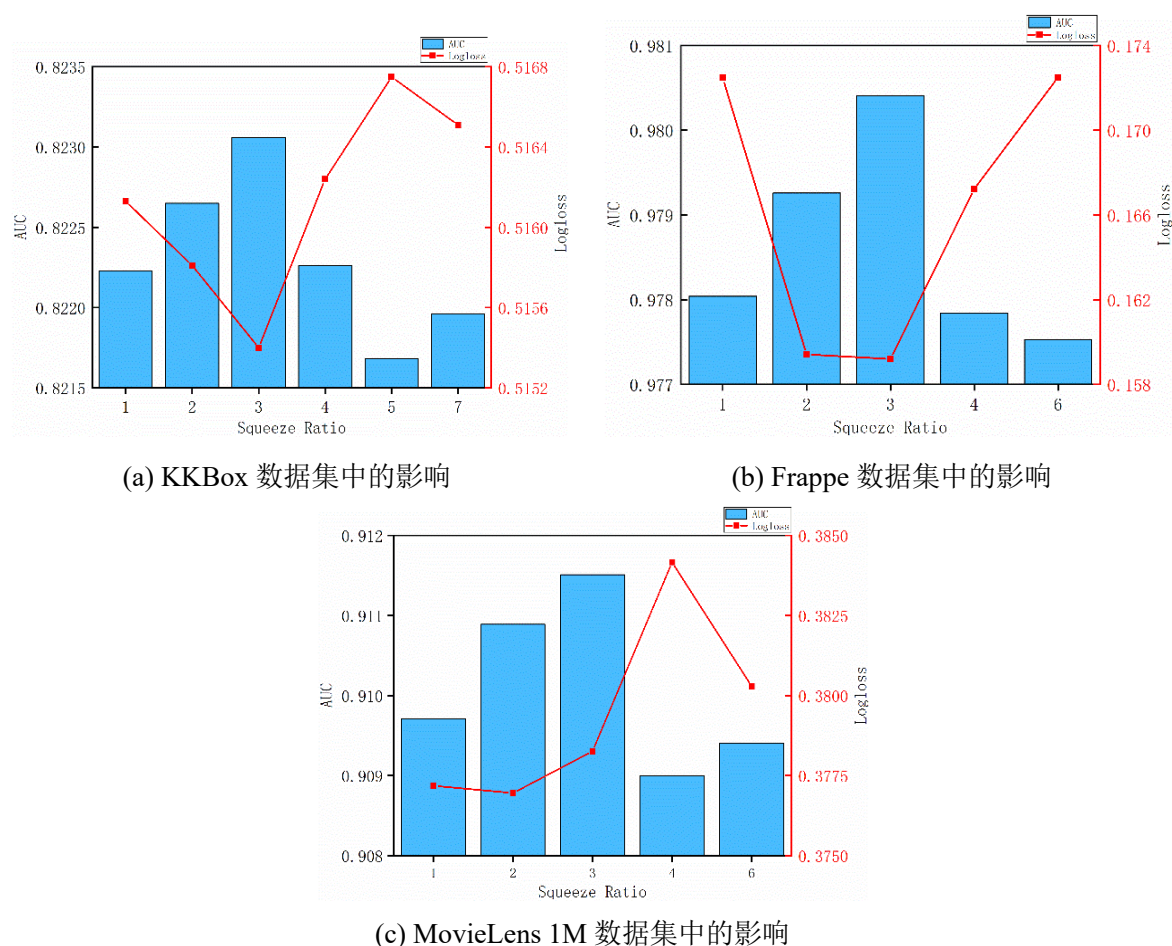


图 4.5 不同全局衰减比例对 AUC 和 Logloss 的影响

从图 4.5 中可以看出，在 KKBox 数据集和 Frappe 数据集中，最佳的衰减比例为



3, MovieLens 1M 数据集在衰减比例分别取 2 和 3 时, 上 Logloss 和 AUC 得到了最佳结果。考虑到三个数据集的特征域数量差别不大, 推断原始特征域的数量压缩 30%-40%达到激励的最优效果。

#### 4.5.7 消融实验

为了更好地对 HDGFI 模型进行性能评估, 进一步开展消融实验研究。HDGFI 模型主要有两大改进: (1) 使用了局部级别的注意力消息传递来捕捉成对的有益特征节点之间的交互; (2) 使用了全局级别的压缩激励模块来保留先前层学到的特征组合信息。

首先, 构建两个变体模型 HDGFI\_L 和 HDGFI\_G, 开展局部特征交互和全局级别动态重要性选择的消融实验。所构建的两个变体模型为:

HDGFI\_L: 去除局部特征交互的 HDGFI 模型, 使用 DNN 来捕捉高阶特征交互;

HDGFI\_G: 去除全局级别动态重要性选择的 HDGFI 模型。

消融实验的结果如表 4.4 所示。

表 4.4 HDGFI与HDGFI\_L和HDGFI\_G变体的性能比较

| 数据集     | KKBox          |                | Frappe         |                | MovieLens 1M   |                |
|---------|----------------|----------------|----------------|----------------|----------------|----------------|
|         | AUC            | Logloss        | AUC            | Logloss        | AUC            | Logloss        |
| 变体方法    |                |                |                |                |                |                |
| HDGFI_L | 0.80779        | 0.53411        | 0.97331        | 0.19246        | 0.90016        | 0.39976        |
| HDGFI_G | 0.81679        | 0.52293        | 0.97598        | 0.17051        | 0.90914        | 0.38133        |
| HDGFI   | <b>0.82278</b> | <b>0.51555</b> | <b>0.97894</b> | <b>0.16495</b> | <b>0.91113</b> | <b>0.37871</b> |

从表 4.4 中可以观察到, 与 HDGFI 模型相比, 两个变体模型的性能有所下降, 这表明局部级别和全局级别的特征交互都是必要的。HDGFI\_L 与 HDGFI 相比, 节点间的成对交互可以反映局部交互的重要性。与 DNN 隐式捕获高阶交互方法相比, 将特征建模为节点并迭代更新节点表示, 可以更好显式地建模特征间的交互。HDGFI\_G 与 HDGFI 相比, 全局级别的特征交互可以自适应地捕捉先前层学习到的特征节点的重要性, 使每一层可以学到更为准确的节点表示, 进而帮助更好地学习当前层的特征图表示和下一层的特征图邻接矩阵。

接下来,进一步开展分层邻接矩阵学习和双线融合的消融实验。所构建的两个变体模型分别是:

HDGFI\_E: 去除分层邻接矩阵学习 Top-k 选择的 HDGFI 模型,在每层都构建一个完全连通特征图;

HDGFI\_B: 去除双线性融合的 HDGFI 模型,使用简单向量拼接替代。

分层邻接矩阵学习和双线性融合消融实验的结果如表 4.5 所示。

表 4.5 HDGFI与HDGFI\_E和HDGFI\_B变体的性能比较

| 变体方法    | KKBox          |                | Frappe         |                | MovieLens 1M   |                |
|---------|----------------|----------------|----------------|----------------|----------------|----------------|
|         | AUC            | Logloss        | AUC            | Logloss        | AUC            | Logloss        |
| HDGFI_E | 0.82149        | 0.51716        | 0.97741        | <b>0.16437</b> | 0.90939        | 0.38051        |
| HDGFI_B | 0.82111        | 0.51738        | 0.97738        | 0.17088        | 0.90975        | 0.38025        |
| HDGFI   | <b>0.82278</b> | <b>0.51555</b> | <b>0.97894</b> | 0.16495        | <b>0.91113</b> | <b>0.37871</b> |

从表 4.5 可以看出,相对于 HDGFI 模型,两个变体模型的性能均有所下降。HDGFI\_E 与 HDGFI 相比,特征被建模为全连通特征图,边的权重是固定的。这种方法在一定程度上保留了所有的特征之间的相互作用,但是不必要的交互对最终的预测结果有负面的影响。分层的邻接矩阵学习和 Top-k 有益交互选择只关注较强的特征节点交互,从而过滤相对不重要的特征之间的交互。HDGFI\_B 与 HDGFI 的实验结果相比显示,双线性交叉聚合函数能够更好地对两个级别的节点表示进行细粒度融合。

## 4.6本章小结

本章提出了分层双级别图融合特征交互推荐算法 HDGFI。该模型采用分层邻接矩阵学习来过滤不必要的特征节点交互,减少交互噪声对预测的干扰;构建了局部消息传递和全局压缩激励两个级别的交互模块,从两个角度捕捉特征的重要性。针对该模型开展了实验研究,并与其它主流模型进行了对比分析;进一步开展了 HDGFI 模型的超参实验和模块消融实验。实验结果表明,提出的 HDGFI 在三个公共数据集上的准确率高于其它算法,模型设计合理有效。

## 5 总结与展望

随着深度学习和大数据的快速发展,推荐系统近年来受到了广泛的关注和研究,特征交互推荐算法是推荐系统领域的研究热点之一。当前众多特征交互推荐模型尚存在一些不足和有待提高之处。首先,现存模型没有对不同级别的特征交互的重要性进行区分,对所有的特征域进行同等重要地建模和交互,然而不同的特征域交互对最终预测结果可能产生不同的影响作用;其次,并不是所有的特征交互都能对最终的预测产生积极的影响,一些不必要的特征交互在训练过程中会带来噪音。本文针对现有特征交互推荐算法存在的局限性开展研究,运用图领域的两大技术,即图相似性学习和图结构学习技术,结合图神经网络,提出了注意力感知的图匹配特征交互推荐算法和分层双级别图融合特征交互推荐算法两种模型。主要工作与研究成果如下:

(1) 提出了一种注意力感知的图匹配特征交互推荐 (AGMRec) 算法。AGMRec 方法将用户和物品的属性信息建模为两个特征图,通过内部和外部两个级别的交互建模特征组合过程。此外,在特征节点交互阶段和节点表示融合阶段都运用注意力机制捕捉交互重要性。在两个公开数据集上进行了实验,结果表明 AGMRec 方法与目前现有的大多数特征交互推荐方法相比具有一定的优越性。

(2) 针对第一个工作未考虑了上下文信息在推荐过程中对预测结果的影响,提出了一种分层双级别图融合特征交互推荐 (HDGFI) 方法。该方法将所有特征域建模为一个特征图,并结合图结构学习思想,在每层选出最重要的交互,消除不相关信息的影响。同时,在每层图神经网络迭代过程中,从局部和全局两个级别动态捕捉特征的重要性。在三个公开数据集进行了推荐结果的预测,并进一步完成了超参实验和模块消融实验,结果表明 HDGFI 方法合理且有效。

虽然本文提出的两个基于图神经网络的特征交互推荐方法取得了一些积极的研究成果,但仍存在一些弊端。首先,AGMRec 方法构建了两个完全连通的特征图,这样可能会使得一些非必要的边带入噪声信息。将来的研究中可以参考 HDGFI 的研究思路,重新设计构建特征图模块并且优化消息传递和图匹配方式,以提高模型的性能和推荐过程的可解释性。其次,HDGFI 模型中目标节点的邻居采样个数  $k$  是一个重要的超参数,网格搜索法耗时且可能会陷入局部最优解,未来研究中计划寻求自适

应最优  $k$  值选择方法，提高模型的效率。

## 参考文献

- [1] Batmaz Z, Yurekli A, Bilge A, et al. A review on deep learning for recommender systems: challenges and remedies[J]. Artificial Intelligence Review, 2019, 52(1): 1-37.
- [2] Zhu J, Han L, Gou Z, et al. A Fuzzy Clustering-Based Denoising Model for Evaluating Uncertainty in Collaborative Filtering Recommender Systems[J]. Journal of the American Society for Information Science and Technology, 2018, 69(9): 1109-1121.
- [3] 魏娟, 李敏. 信息过载影响消费者决策研究的知识图谱分析[J]. 管理现代化, 2022, 42(01): 156-161.
- [4] 管其平. 数字化生存中的信息过载及其空间治理[J]. 昆明理工大学学报(社会科学版), 2021, 21(1): 99-107.
- [5] Varma M. Extreme Classification: Tagging on Wikipedia, Recommendation on Amazon & Advertising on Bing[C]// Proceedings of the Web Conference 2018. 2018: 1897-1897.
- [6] Mooney R J, Roy L. Content-based book recommending using learning for text categorization[C]// Proceedings of the 5th ACM Conference on Digital Libraries. 2000, 195-204.
- [7] Medjahed B, Atif Y. Context-based matching for Web service composition[J]. Distributed and Parallel Databases, 2007, 21(1): 5-37.
- [8] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]// Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. 1998: 43-52.
- [9] Wang X, He X, Wang M, et al. Neural graph collaborative filtering[C]// Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019: 165-174.
- [10] Huang Z, Yu C, Ni J, et al. An Efficient Hybrid Recommendation Model With Deep Neural Networks[J]. IEEE Access, 2019, 7: 137900-137912.
- [11] Hu J, Liu L, Zhang C, et al. Hybrid Recommendation Algorithm Based on Latent Factor Model and PersonalRank[J]. Journal of Internet Technology, 2018, 19(3): 919-926.
- [12] Richardson M, Dominowska E, Ragno R. Predicting Clicks: Estimating the Click-

- Through Rate for New Ads[C]// Proceedings of the 16th International Conference on World Wide Web. 2007: 521-530.
- [13]刘梦娟, 曾贵川, 岳威, 等. 基于融合结构的在线广告点击率预测模型[J]. 计算机学报, 2019, 42(7): 1570-1587.
- [14]Rendle S. Factorization Machines[C]// Proceedings of the 2010 IEEE 10th International Conference on IEEE. 2010: 955-1000.
- [15]Juan Y, Zhuang Y, Chin W S, et al. Field-aware Factorization Machines for CTR Prediction[C]// Proceedings of the 10th ACM Conference. 2016: 43-40.
- [16]Pan J, Xu J, Ruiz A L, et al. Field-weighted factorization machines for click-through rate prediction in display advertising[C]// Proceedings of the 2018 World Wide Web Conference. 2018: 1349-1357.
- [17]Sun Y, Pan J, Zhang A, et al. FM2: Field-matrixed factorization machines for recommender systems[C]// Proceedings of the Web Conference 2021. 2021: 2828-2837.
- [18]Xiao J, Ye H, He X, et al. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks[C]// Proceedings of the 26th International Joint Conference on Artificial Intelligence. 2017: 3119-3125.
- [19]Duan D, Gai X, Han Z, et al. Micro-blog misinformation detection based on gradient boost decision tree[J]. Journal of Computer Applications, 2018, 38(2): 410-414.
- [20]Jin K H, Mccann M T, Froustey E, et al. Deep Convolutional Neural Network for Inverse Problems in Imaging[J]. IEEE Transactions on Image Processing, 2017, 26(9): 4509-4522.
- [21]Lauriola I, Lavelli A, Aiolfi F. An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools[J]. Neurocomputing, 2022, 470(22): 331-336.
- [22]Tianyi Z, Yang H, Valsdottir L R, et al. Identifying drug-target interactions based on graph convolutional network and deep neural network[J]. Briefings in Bioinformatics, 2021, 22(2): 2141-2150.
- [23]Zhang W, Du T, Wang J. Deep learning over multi-field categorical data[C]// Proceedings of the European conference on information retrieval. 2016: 45-57.

- [24]Shan Y, Hoens T R, Jiao J, et al. Deep Crossing: Web-Scale Modeling without Manually Crafted Combinatorial Features[C]// Proceedings of the SIGKDD International Conference. ACM, 2016: 255-262.
- [25]Qu Y, Cai H, Ren K, et al. Product-based Neural Networks for User Response Prediction[C]// Proceedings of the 2016 IEEE 16th International Conference. 2016: 1149-1154.
- [26]He X, Chua T S. Neural Factorization Machines for Sparse Predictive Analytics[C]// Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017: 355-364.
- [27]Cheng H T, Koc L, Harmsen J, et al. Wide & Deep Learning for Recommender Systems[C]// Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM, 2016: 7-10.
- [28]Wang R, Fu B, Fu G, et al. Deep & cross network for ad click predictions[C]// Proceedings of the ADKDD'17. 2017: 1-7.
- [29]Guo H, Tang R, Ye Y, et al. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction[C]// Proceedings of the 26th International Joint Conference on Artificial Intelligence. 2017: 1725-1731.
- [30]Lian J, Zhou X, Zhang F, et al. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems[C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2018: 1754-1763.
- [31]Huang T, Zhang Z, Zhang J. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction[C]// Proceedings of the 13th ACM Conference on Recommender Systems. 2019: 169-177.
- [32]Hu J, Shen L, Sun G, et al. Squeeze-and-Excitation Networks[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 7132-7141.
- [33]Song w, Shi C, Xiao Z, et al. AutoInt: Automatic feature interaction learning via self-attentive neural networks[C]// Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019: 1161-1170.
- [34]Xu Y, Zhu Y, Yu F, et al. Disentangled self-attentive neural networks for click-through

- rate prediction[C]// Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021: 3553-3557.
- [35]Li Z, Cheng W, Chen Y, et al. Interpretable click-through rate prediction through hierarchical attention[C]// Proceedings of the 13th International Conference on Web Search and Data Mining. 2020: 313-321.
- [36]Cheng W, Shen Y, Huang L. Adaptive factorization network: Learning adaptive-order feature interactions[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(04): 3609-3616.
- [37]Li Z, Cui Z, Wu S, et al. Fi-GNN: Modeling Feature Interactions via Graph Neural Networks for CTR Prediction[C]// Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019: 539-548.
- [38]Su Y, Zhang R, Erfani S, et al. Detecting beneficial feature interactions for recommender systems[C]// Proceedings of the AAAI conference on artificial intelligence. 2021, 35(5): 4357-4365.
- [39]Zheng Z, Zhang C, Gao X, et al. HIEN: hierarchical intention embedding network for click-through rate prediction[C]// Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022: 322-331.
- [40]Li Z, Wu S, Cui Z, et al. GraphFM: Graph Factorization Machines for Feature Interaction Modeling[J]. arXiv preprint arXiv: 2105.11866, 2021.
- [41]Su Y, Zhang R, Erfani S M, et al. Neural Graph Matching based Collaborative Filtering[C]// Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021: 849-858.
- [42]黄立威, 江碧涛, 吕守业, 等. 基于深度学习的推荐系统研究综述[J]. 计算机学报, 2018, 41(7): 1619-1647.
- [43]项亮. 推荐系统实践[M]. 北京市:人民邮电出版社, 2012: 1-200.
- [44]吴静, 谢辉, 姜火文. 图神经网络推荐系统综述[J]. 计算机科学与探索, 2022, 16(10): 2249-2263.
- [45]Gao C, Zheng Y, Li N, et al. Graph neural networks for recommender systems: Challenges, methods, and directions[J]. arXiv preprint arXiv:2109.12843,2021.



- [46] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.
- [47] Hamilton W L, Ying R, Leskovec J. Inductive representation learning on large graphs[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 1025-1035.
- [48] Velickovi P, Cucurull G, Casanova A, et al. Graph Attention Networks[C]// Proceedings of the International Conference on Learning Representations. 2018: 1-12.
- [49] Li Y, Gu C, Dullien T, et al. Graph Matching Networks for Learning the Similarity of Graph Structured Objects[C]// Proceedings of the 36th International Conference on Machine Learning. 2019.
- [50] Zhu Y, Xu W, Zhang J, et al. A Survey on Graph Structure Learning: Progress and Opportunities[J]. arXiv preprint arXiv: 2103.03036, 2021.
- [51] Li R, Sheng W, Zhu F, et al. Adaptive Graph Convolutional Neural Networks[J]. arXiv preprint arXiv:1801.03226, 2018.
- [52] Chen Y, Wu L, Zaki M J. Iterative Deep Graph Learning for Graph Neural Networks: Better and Robust Node Embeddings[C]// Proceedings of the Advances in Neural Information Processing Systems. 2020: 19314-19326.
- [53] Jiang B, Zhang Z, Lin D, et al. Semi-Supervised Learning With Graph Learning-Convolutional Networks[C]// Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 11313-11320.
- [54] Luo D, Cheng W, Yu W, et al. Learning to Drop: Robust Graph Neural Network via Topological Denoising[C]// Proceedings of the fourteenth ACM international conference on Web Search and Data Mining. 2021: 779-787.
- [55] Gao X, Hu W, Guo Z. Exploring Structure-Adaptive Graph Learning for Robust Semi-Supervised Classification[C]// Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME). 2020: 1-6.
- [56] Franceschi L, Niepert M, Pontil M, et al. Learning discrete structures for graph neural networks[C]// Proceedings of the international conference on machine learning. PMLR, 2019: 1972-1982.

- [57]<http://www2.informatik.uni-freiburg.de/~ciegler/BX/>
- [58]<http://www.grouplens.org/node/73>
- [59]<https://www.kaggle.com/c/kkbox-music-recommendation-challenge/data>
- [60][http://baltrunas.info/data/CARS2\\_code.zip](http://baltrunas.info/data/CARS2_code.zip)

## 附录一表目录

|       |                                         |    |
|-------|-----------------------------------------|----|
| 表 3.1 | AGMRec 模型的训练.....                       | 28 |
| 表 3.2 | 数据集描述 .....                             | 30 |
| 表 3.3 | AGMRec 和基线模型的性能比较.....                  | 32 |
| 表 3.4 | AGMRec 与三个变体的性能比较.....                  | 34 |
| 表 4.1 | HDGFI 模型的训练.....                        | 45 |
| 表 4.2 | 数据集描述 .....                             | 46 |
| 表 4.3 | HDGFI 和基线模型的性能比较.....                   | 48 |
| 表 4.4 | HDGFI 与 HDGFI_L 和 HDGFI_G 变体的性能比较 ..... | 51 |
| 表 4.5 | HDGFI 与 HDGFI_E 和 HDGFI_B 变体的性能比较 ..... | 52 |



## 附录二图目录

|       |                                   |    |
|-------|-----------------------------------|----|
| 图 2.1 | 推荐系统的三个阶段 .....                   | 7  |
| 图 2.2 | DeepFM 模型框架图 .....                | 10 |
| 图 2.3 | Fi-GNN 模型框架图 .....                | 11 |
| 图 2.4 | 图卷积网络结构示意图 .....                  | 15 |
| 图 2.5 | 图采样和聚集网络结构示意图 .....               | 16 |
| 图 2.6 | 图注意力层结构示意图 .....                  | 17 |
| 图 2.7 | 图相似性计算流程图 .....                   | 18 |
| 图 2.8 | 图结构学习流程图 .....                    | 19 |
| 图 3.1 | AGMRec 模型结构图 .....                | 24 |
| 图 3.2 | 不同嵌入维度对 AUC 和 Logloss 的影响 .....   | 33 |
| 图 3.3 | 不同注意力头数对 AUC 和 Logloss 的影响 .....  | 33 |
| 图 4.1 | HDGFI 模型结构图 .....                 | 38 |
| 图 4.2 | 分层边选择框架图 .....                    | 40 |
| 图 4.3 | 双级节点表征生成的架构 .....                 | 41 |
| 图 4.4 | 邻居采样个数对 AUC 的影响 .....             | 49 |
| 图 4.5 | 不同全局衰减比例对 AUC 和 Logloss 的影响 ..... | 50 |