

分类号

U D C

密 级

编 号 10486

武汉大学

硕 士 专 业 学 位 论 文

基于对比学习的知识感知推荐模型研究

研 究 生 姓 名 : 卢佳瑶

学 号 : 2020282110169

指导教师姓名、职称 : 王健 副教授

专业类别 (领域) : 电子信息 (软件工程)

二〇二四年五月

Research on Knowledge-aware Recommendation Models based on Contrastive Learning

By

Jiayao Lu

Supervised By

Prof. Jian Wang

Wuhan University

May, 2024

论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的研究成果。除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

学位论文作者（签名）：

年 月 日

武汉大学学位论文使用授权协议书

本学位论文作者愿意遵守武汉大学关于保存、使用学位论文的管理办法及规定，即：学校有权保存学位论文的印刷本和电子版，并提供文献检索与阅览服务；学校可以采用影印、缩印、数字化或其它复制手段保存论文；在以教学与科研服务为目的前提下，学校可以在校园网内公布部分或全部内容。

一、在本论文提交当年，同意在校园网内以及中国高等教育文献保障系统（CALIS）、高校学位论文系统提供查询及前十六页浏览服务。

二、在本论文提交☐当年/☐一年/☐两年/☐三年以后，同意在校园网内允许读者在线浏览并下载全文，学校可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。（保密论文解密后遵守此规定）

论文作者（签名）：_____

学 号：_____

学 院：_____

日期： 年 月 日

摘 要

随着互联网和大数据时代的兴起，信息过载逐渐成为一个日益严重的问题。推荐系统作为一种有效的信息过滤手段，能够满足不同用户的个性化需求。为了克服冷启动和数据稀疏对推荐算法性能的影响，推荐模型引入知识图谱作为辅助信息来丰富物品的学习表示，利用图神经网络（Graph Neural Network, GNN）挖掘知识图谱中的高阶连通性信息，在提升推荐性能的基础上，还可增强模型可解释性。然而，融入知识图谱的推荐（又名知识感知推荐，KGR）仍然面临着多方面的挑战。一方面，由于交互数据的极度稀疏，基于 GNN 的知识感知推荐模型存在训练信号稀疏问题，即因为训练标签的稀少无法准确学习用户和项目的特征向量表示，影响最终推荐结果的准确性；另一方面，在 GNN 多层神经网络进行消息聚合后，容易导致用户和项目节点学习表示无法区分的退化问题，即在嵌入空间层面很多不相似的物品的学习表示很集中，在整个训练过程中小部分热门的物品提供了大多数的交互数据，从而产生长尾分布问题，影响模型的性能。

针对上述问题，本文拟充分利用对比学习（CL）在未标记的原始数据中学习通用特征的能力，从数据本身挖掘监督信号，探索对比学习在基于 GNN 的知识感知推荐模型中的应用。本文的研究内容总结如下：

（1）针对基于 GNN 的知识感知推荐模型中存在的训练信号稀疏问题，本文提出了一种多视图对比学习的知识感知推荐模型（MVCK），该方法首先利用不同的数据增强方式构建用户-项目视图以及项目-实体视图，然后通过不同的 GNN 图编码器分别学习用户项目交互数据以及知识图谱中的信息，最后利用多负例对比学习挖掘不同视图之间的信息。

（2）对于多层网络聚合后导致的数据分布不均产生的长尾分布问题，本文提出了一种多视图噪声增强图对比学习的知识感知推荐模型（MNCL），该方法在图卷积神经网络聚合过程中引入一定程度的噪声，通过给节点注入不确定性的噪声因素来优化节点在嵌入空间特征表示下的均匀化分布，从而缓解长尾问题，强化对比学习的性能。

最后，在 Movielens-1M、Last.FM 和 Book-Crossing 三个数据集上对所提两个模型的推荐性能分别进行了评估。进行了充分的模型对比和消融实验，并分析了不同超参数设置下的推荐性能。实验结果表明，所提模型在三个数据集上相较于最新的模型均有一定的提升。

关键词：知识感知推荐；对比学习；多视图；噪声增强

Abstract

With the rise of the Internet and the big data era, information overload has gradually become an increasingly serious problem. As an effective means of information filtration, recommender systems can meet the personalized needs of different users. To overcome the impact of cold start and data sparsity on the performance of recommendation algorithms, recommendation models introduce knowledge graphs as auxiliary information to enrich the learning representations of items. By utilizing Graph Neural Networks (GNN) to explore higher-order connectivity information in knowledge graphs, the recommendation performance can be enhanced, while also improving the interpretability of the model. However, recommendation that integrate knowledge graphs(aka. knowledge-aware recommendation, KGR) still face various challenges. On one hand, due to the extreme sparsity of interaction data, knowledge graph recommender models based on GNNs face the challenge of sparse training signals. This means that due to the scarcity of training labels, accurate learning of feature vector representations for users and items is hindered, impacting the accuracy of the final recommendation results. On the other hand, after message aggregation in the multi-layer neural networks of GNNs, it is prone to encountering the degradation problem of indistinguishable representations for user and item nodes. This implies that many dissimilar items are concentrated in the embedding space, with a few popular items providing the majority of interaction data throughout the training process, resulting in a long-tail distribution problem that impacts the model's performance.

In response to the aforementioned issues, this thesis intends to fully leverage the ability of Contrastive Learning (CL) to learn universal features from unlabeled raw data and mine supervisory signals from the data itself. We aim to explore the application of Contrastive Learning in GNN-based knowledge graph recommendation models. The research content of this thesis is summarized as follows:

(1) To address the issue of sparse training signals in GNN-based knowledge graph recommendation models, this thesis proposes a multi-view contrastive learning approach for knowledge-aware recommendation models (MVCK). This method first utilizes various data augmentation techniques to construct user-item views and item-entity views. Then, different GNN graph encoders are employed to learn the

user-item interaction data and the information in the knowledge graph separately. Finally, multi-negative instance contrastive learning is utilized to explore information between different views.

(2) Regarding the issue of long-tail distribution caused by uneven data distribution after multi-layer network aggregation, this thesis proposes a knowledge-aware recommendation model using multi-view noisy contrastive learning (MNCL). In this approach, a certain degree of noise is introduced during the multi-layer aggregation process of graph convolutional neural networks. By injecting uncertainty into the nodes through noise factors, the model aims to optimize the uniform distribution of node embeddings in feature space, thus alleviating the long-tail problem and enhancing the performance of contrastive learning.

Finally, the recommendation performance of the two proposed models was evaluated on three datasets: Movielens-1M, Last.FM, and Book-Crossing. Sufficient model comparisons and ablation experiments were conducted, and the recommendation performance under different hyperparameter settings was analyzed. The experimental results indicate that the proposed models have achieved certain improvements compared to the latest models on all three datasets.

Keywords: Knowledge-aware recommendation, Contrastive learning, Multi-view, Noise enhancement

目录

摘 要	I
Abstract	II
目录	IV
图目录	VII
表目录	VIII
第 1 章 绪论	1
1.1 研究背景和意义	1
1.2 国内外研究现状和发展趋势	2
1.2.1 传统的推荐算法	2
1.2.2 基于知识图谱的推荐算法	5
1.2.3 基于深度学习的推荐算法	6
1.3 论文的主要内容	7
1.4 论文的结构和框架	8
第 2 章 相关理论和综述	10
2.1 知识图谱及其相关技术	10
2.1.1 知识图谱概念及定义	10
2.1.2 基于嵌入的推荐模型	10
2.1.3 基于路径的推荐模型	12
2.1.4 基于混合的推荐模型	13
2.2 图神经网络	13
2.2.1 图神经网络的定义	14
2.2.2 基于图神经网络的推荐模型	15
2.3 对比学习	16
2.3.1 对比学习框架与步骤	16
2.3.2 对比学习在推荐系统中应用	17
2.4 本章小结	18
第 3 章 多视图对比学习的知识感知推荐模型	19
3.1 研究动机	19
3.2 问题定义	20
3.2.1 用户-相互交互数据	21
3.2.2 知识图谱	21
3.2.3 基于 GNN 的知识感知推荐模型的推荐任务	21

3.3 算法框架	21
3.3.1 子图构建	22
3.3.2 关系感知的 GNN	23
3.3.3 Light-GCN	24
3.3.4 多负例对比学习	25
3.3.5 模型预测	26
3.3.6 算法流程	27
3.4 实验设置	27
3.4.1 实验数据集	27
3.4.2 实验环境	28
3.4.3 评估指标	28
3.4.4 实验模型对比	30
3.5 实验结果以及分析	31
3.5.1 实验结果综合对比	31
3.5.2 消融实验	33
3.5.3 超参敏感性分析	33
3.6 本章小结	37
第 4 章 多视图噪声增强图对比学习的知识感知推荐模型	38
4.1 研究动机	38
4.2 问题定义	39
4.2.1 用户-项目交互数据	39
4.2.2 知识图谱	39
4.2.3 推荐任务	40
4.3 算法框架	40
4.3.1 子图构建	41
4.3.2 关系感知 GNN	41
4.3.3 噪声增强的 Light-GCN	42
4.3.3 路径感知 GNN	43
4.3.5 多负例对比学习	44
4.3.6 模型预测	44
4.3.7 算法流程	45
4.4 实验设置	45
4.4.1 实验数据集	45
4.4.2 实验环境	47

4.4.3 评估指标	47
4.4.4 实验模型对比	47
4.5 实验结果以及分析	49
4.5.1 实验结果综合对比	49
4.5.2 消融实验	50
4.5.3 超参敏感性分析	51
4.6 本章小结	54
第 5 章 总结与展望	56
5.1 论文工作总结	56
5.2 未来工作展望	57
参考文献	58
致 谢	64
硕士期间研究成果	65

图目录

图 1.1 论文主要内容	8
图 1.2 论文主要框架	9
图 2.1 常见的知识图谱嵌入方法	11
图 2.2 对比学习基本步骤	17
图 3.1 MVCK 模型图	22
图 3.2 多负例对比学习	25
图 3.2 AUC 示意图	29
图 3.3 对比学习权重 α 对模型性能的影响	34
图 3.4 用户-项目视图 Light-GCN 层数 L 对模型性能的影响	34
图 3.5 项目-实体视图 Light-GCN 层数 K 对模型性能的影响	35
图 3.6 对比学习原视图权重 ω 对模型性能的影响	36
图 3.7 温度系数 τ 对模型性能的影响	36
图 4.1 MNCL 模型图	40
图 4.2 对比学习权重 α 对模型性能的影响	51
图 4.3 噪声增强系数 β 对模型性能的影响	52
图 4.4 噪声增强的 Light-GCN 层数 L 对模型性能的影响	52
图 4.5 项目视图 Light-GCN 层数 K 对模型性能的影响	54
图 4.6 对比学习原视图权重 ω 对模型性能的影响	54

表目录

表 3.1 MVCK 算法流程	26
表 3.2 实验数据集的基本统计数据	28
表 3.3 实验环境	29
表 3.4 模型 MVCK 的参数设置	31
表 3.5 各模型的实验数据对比	32
表 3.6 MVCK 及其消融变体的影响	33
表 4.1 MNCL 算法流程	46
表 4.2 模型 MNCL 的参数设置	49
表 4.3 各模型的实验数据对比	50
表 4.4 MNCL 及其消融变体的影响	51

第 1 章 绪论

1.1 研究背景和意义

随着互联网的迅速发展，人们在日常生活中产生了大量多样化的数据，包括浏览历史、社交网络互动和在线购物行为等。随之而来的是海量的数据和蓬勃发展的大数据技术^[1]。这些数据不仅反映了用户的个性化兴趣和偏好，还提供了丰富的信息。但是信息爆炸式增长也带来了新的问题，面对海量的数据，人们无法从中快速的获取自己更感兴趣的信息，商家也无法快速的将自己的产品推荐给感兴趣的用户。为了缓解这种信息过载的问题，个性推荐算法应运而生。个性化推荐算法是一种根据用户的个性化需求和偏好，从海量的信息中挖掘并推荐最符合用户兴趣的内容或物品的技术。其核心目标是为每个用户提供个性化、精准的推荐结果，以提升用户体验和满意度。

传统的个性化推荐算法按照计算方式的不同，主要分为下面这三类：基于内容的推荐算法^[2]、基于协同过滤的推荐算法、基于混合的推荐算法。这些算法利用了多种不同的数据结构和多种推荐技术，例如用户的各种行为数据、物品的属性数据、社交网络数据等，来生成更加个性化和精准的推荐结果。然而，由于冷启动问题、数据稀疏性和长尾分布等问题^[3-4]的存在，这些传统的个性推荐算法的精度性都不算很高。

冷启动问题指在推荐模型刚刚启动或新用户加入时，推荐模型缺乏足够的历史数据来做出个性化推荐。而数据稀疏性是因为许多用户仅与少数物品互动，面对千万级甚至上亿的物品来说，模型很难训练出用户的个性化需求，因此降低了推荐的准确性^[5-6]。此外，传统推荐算法可能会导致用户陷入“信息茧房”，只推荐类似之前选择的物品，忽略了物品推荐潜在的多样性^[7]，长期以往会导致用户的审美疲劳。

为了应对上述挑战，研究者开始将知识图谱引入推荐模型，以扩充用户和物品的特征表示，提高推荐的质量和个性化程度^[8]。作为一种结构化的知识表示方式，知识图谱主要由实体和关系组成，并以三元组的形式存储。其中，实体指的是某一具体的实物或者人为定义的概念（包括时间、长度等），关系是两个实体之间的连线，与知识图不同的是，知识图谱的关系是有方向性的。例如三元组（易建联，生日，1987 年 10 月 27 日）指的是“易建联的生日是 1987 年 10 月 27 日”，其中实体分别是“易建联”和“1987 年 10 月 27 日”，实体之间的关系是“生日”，并且实体的方向是从“易建联”指向“1987 年 10 月 27 日”。知识图谱

包含了丰富的跨领域知识，如人物、地点、事件等，以及它们之间的关系。知识图谱作为辅助信息，极大程度上提高了推荐系统的可解释性以及推荐的准确性，而且一定程度上缓解了冷启动问题、以及数据稀疏性问题。

早期的知识图谱推荐系统包括基于知识图谱的特征表示以及融合多源信息的推荐模型。前者主要侧重于利用知识图谱中的实体关系和属性信息，为用户和物品构建特征表示。这些算法通常使用图嵌入技术，如 TransE^[27]、TransH^[28]等，将知识图谱中的实体映射到低维向量空间中，并利用这些向量表示来计算用户和物品之间的相似度。后者通过结合不同数据源的信息，构建更加丰富和准确的用户和物品表示，从而提高推荐的效果和性能。但是都忽略了知识图谱的全局信息，即实体之间的高阶连通性信息。随着图神经网络(GNNs)的产生以及发展，GNNs在推荐算法中的应用也越来越广泛，基于 GNN 的推荐模型能够有效地利用知识图谱中的图结构信息，从而实现更加准确和个性化的推荐^[9]。这些模型通常通过在图上进行消息传递和图卷积操作来学习用户和物品的表示，并利用这些表示进行推荐。

基于 GNN 的推荐模型不断发展，并且取得了巨大的突破。然而基于 GNN 的推荐模型因为用户项目数据交互极度稀疏一直存在问题。其中包括训练信号稀疏问题，节点表示分布不均引起的长尾分布问题等。然而缓解训练信号稀疏问题需要大量的标签。近年来，对比学习(CL)作为一种自监督学习方法，能够从未标记的数据中挖掘出更多的信息，在各个领域取得了巨大的成功，同时也在知识感知推荐领域也得到了广泛的应用。对比学习通过比较正样本和负样本之间的相似度来学习特征表示，从而能够挖掘数据中的潜在结构和关系，提高推荐的性能和效果。

本文主要是探究了对比学习在基于 GNN 的知识感知推荐系统中的应用，利用对比学习方法缓解训练信号稀疏问题，同时探究在多层神经网络聚合过程中加入噪声增强，缓解数据分布不均而导致的长尾分布问题。

1.2 国内外研究现状和发展趋势

推荐系统始于上个世纪 90 年代^[10]，如今已经发展到各行各业，几乎所有的互联网行业，都会涉及到推荐系统的运用。本节，将会从传统的推荐算法、基于知识图谱的推荐算法以及基于深度学习的推荐算法进行综述。

1.2.1 传统的推荐算法

在综述^[11]中，传统的个性化推荐算法按照计算方式的不同，主要分为下面这

三类：基于内容的推荐算法、基于协同过滤的推荐算法以及基于混合的推荐算法。

（1）基于内容的推荐算法

基于内容的推荐算法，根据用户自身的描述（内容）来定义用户的属性。算法会根据用户过去的历史行为包括评分或者点击，来分析用户可能存在的兴趣倾向，从而为用户推荐类似的物品。例如在一个音乐推荐系统中，为了给用户小芳推荐合适的音乐，基于内容的推荐系统会学习小芳的历史播放歌单，并且从中学习到小芳的听歌爱好，然后会推荐小芳与之前歌曲相似度很高的新的歌曲。

基于内容的推荐方法起源于信息检索^[12]和信息过滤^[13]。其主要过程如下：首先，通过分析物品的内容特征和用户的偏好，构建用户以及物品的嵌入，然后计算物品之间的相似度（通常是计算余弦距离）以及用户与物品之间的匹配程度，从而向用户推荐与其过去喜欢的物品相似的新物品。基于内容的推荐算法不仅考虑了物品的内容信息，还充分利用了用户的个性化偏好，原理简单且可解释性强。但是它存在几点不足之处：一、不管是用户还是物品，他们的描述有限，不能够精确地对物品以及用户进行建模。二、只能推荐用户高匹配度的物品，无法发掘用户的新的兴趣，不会充满惊喜。三、存在冷启动问题，新用户加入推荐系统之后，没有历史行为，无法有效的进行推荐。

（2）基于协同过滤的推荐算法

在推荐领域，协同过滤的推荐算法都是用的最广泛的推荐算法之一。其主要思想是，根据其他与某一用户相似的用户对某一物品的评分以及爱好，来预测该用户对该物品的评分。例如，基于协同过滤的推荐系统分析到，小芳与小花的兴趣爱好有很大的相似性，小芳喜欢苹果，那么基于协同过滤的推荐系统会预测小花也会喜欢苹果。协同过滤的方法主要遵循以下两点：首先，用户的兴趣爱好会持续很长一段时间，过去喜欢的未来也会喜欢。其次，用户的兴趣爱好不会突然的消失或者改变，会随着时间相对稳定的变化。根据基本思想以及实现方式不同^[14-15]，协同过滤算法主要分为以下几类：基于用户的协同过滤、基于物品的协同过滤和基于模型的协同过滤。

基于用户的协同过滤推荐算法首先通过计算用户之间的相似度，通常采用余弦相似度或皮尔逊相关系数等指标来衡量用户行为历史的相似程度^[16]。接着，根据相似度计算结果，找出与目标用户最相似的一组用户作为相似用户集合，这些相似用户将成为推荐的依据。随后，在相似用户集合的基础上生成物品候选集合，这些候选物品是相似用户喜欢或者交互过但目标用户尚未接触过的物品。然后，对候选物品集合进行排序，常用的排序方法包括基于评分的排序和基于流行度的

排序，以确定最终推荐给目标用户的物品顺序。最后，将排好序的推荐物品列表展示给用户，完成整个推荐过程。

基于物品的协同过滤主要通过分析物品之间的相似性来进行推荐^[17]。其主要过程如下：首先，构建物品之间的相似度矩阵，常用的相似性度量包括余弦相似度和皮尔逊相关系数。接着，对于目标用户已经有过交互的物品，找出与之相似度最高的一组物品集合，这些相似物品将成为推荐的依据。然后，将目标用户尚未交互过但与其已有行为历史中相似的物品加入候选物品集合。接下来，根据候选物品集合中物品的相似度和用户的历史行为，计算每个候选物品的推荐度。最后，将推荐度高的物品按照一定规则排序，形成最终的推荐列表，将这些物品展示给用户。该算法相比于基于用户的协同过滤具有一定优势，因为物品相对用户更加稳定，计算物品相似度的复杂度相对较低。

基于模型的推荐算法主要是通过机器学习的方法，由于机器学习模型多种多样，基于模型的推荐算法也非常多。但是其大致流程包括以下几个部分：首先对数据预处理；接着进行模型训练，这里需要选择合适的机器学习模型，常见的模型包括基于矩阵分解的模型（如 SVD^[18]、FM^[19]、NeuMF^[20]）、深度学习模型（如神经网络^[21]）等。完成训练后进行评分预测，最后进行推荐。

至今为止，基于协同过滤算法能够广泛应用于各个领域离不开这些优点：简单有效，不需要对物品或用户进行复杂的特征工程，而是直接利用用户的行为数据进行推荐。此外，它能够捕获用户之间的相似性和物品之间的相关性，从而实现个性化推荐。因为其可以分布式计算，因此能够处理大规模的数据，可扩展性好。然而，基于协同过滤的算法也存在一些缺点，包括冷启动问题（对新用户或新物品的推荐效果较差）、数据稀疏性（当用户行为数据较少时，推荐准确度下降）、算法的可解释性较差等。

（3）基于混合的推荐模型

基于混合的推荐模型是为了克服上述两种单一推荐方法的局限性而提出的，它结合了不同推荐算法的优势，以提高推荐系统的性能。通过将多种推荐算法相互结合^[22]，例如将基于内容的推荐与协同过滤相结合，或者将基于内容的推荐与基于模型的推荐相结合，以获得更准确和个性化的推荐结果。其优点在于可以充分利用不同算法的优势，提高推荐的准确性和多样性，但缺点是需要更多的计算资源和算法调优，且模型复杂度较高，不易解释。

1.2.2 基于知识图谱的推荐算法

推荐算法发展至今，一直存在两大难题，分别是“冷启动问题”以及“数据稀疏性问题”。产生问题的主要是数据的质量和数量限制，因此一些辅助信息逐渐引进到推荐系统中，其中知识图谱是最大的热点之一。国内外一些先进的推荐模型已经采用了知识图谱以提高推荐的精度和个性化。例如，知名的电商平台如亚马逊和京东已经开始将知识图谱用于商品推荐，以便更好地理解用户需求和商品属性^[23-24]。根据实现原理和方法的不同，基于知识图谱的推荐算法主要分为三类^[25]：基于知识图谱嵌入的方法、基于路径的方法以及基于混合的方法。

（1）基于知识图谱嵌入的方法

基于知识图谱嵌入的方法是将知识图谱中的实体和关系映射到低维向量空间，旨在捕获实体和关系之间的语义信息。文献^[26]将知识图谱嵌入方法分为基于距离的翻译嵌入和基于语义匹配的嵌入。

基于距离的翻译嵌入主要是在嵌入空间中通过最小化或最大化嵌入向量之间的距离或相似度来学习实体和关系的表示，从而在低维空间中保持图结构的语义一致性，通常情况下，相似的实体在嵌入空间中的距离应该较近，而不相似的实体则应该较远。典型的基于距离的嵌入方法包括 TransE^[27]、TransH^[28]、TransR^[29] 和 TransD^[30]等。

在语义匹配模型中，实体和关系被表示为低维向量，然后通过计算它们之间的相似度来确定它们之间的语义关系。常见的基于语义匹配的嵌入方法例如 DistMult^[31]。

（3）基于路径的方法

基于路径的推荐方法利用实体之间的路径信息进行推荐，将知识图谱视为一个异质信息网络，然后根据路径规则进行匹配计算，以确定最相关的路径，这些规则通常涉及路径选择和路径加权等步骤。路径可以是单个实体之间的直接连接，也可以是通过多个实体之间的关系形成的复杂路径，然后通过计算用户与项目之间的路径来计算用户的偏好兴趣，通常这些路径需要手动设置。例如，PER^[32]模型（Path-based Entity Representation）通过构建用户-项目-项目的元路径来捕获用户和项目之间的关系。它利用不同的元路径扩散用户的偏好，从而生成用户和项目的潜在特征，以提高推荐的准确性和个性化程度。FMG^[33]模型（Factorization Machines with Graph）结合了因子分解机和矩阵分解等方法，集成了不同的元路径信息，以更好地捕获用户和项目之间的关联。RKGE^[34]模型（关系感知

Knowledge Graph Embedding) 通过挖掘具有不同语义的实体路径, 然后利用循环神经网络 (RNN) 技术对这些路径进行编码, 最后通过池化操作来提取不同向量的重要特征, 以增强推荐模型的性能。

基于路径的方法的性能依赖于元路径的质量, 元路径的构建一般需要手动完成, 因此需要工作人员具备相应的专业知识, 耗费了大量的人力成本, 不利于模型的拓展。同时数据稀疏的问题依然存在, 某些路径可能由于数据稀疏性而缺乏有效的信息, 导致模型在这些路径上的性能下降。

(3) 基于混合的方法

为了同时拥有上述两种方法的优点, 一些模型融合了基于知识图谱嵌入的方法和基于路径的方法, 这样可以优化用户和项目的特征表示。基于混合的方法包括例如 RippleNet^[35]和 KGCN^[36]方法。其中 RippleNet 模型通过建模用户与物品之间的复杂关系来提高推荐准确性。该模型利用自适应路径选择和动态路径生成机制, 根据用户和物品的特征动态选择合适的路径进行信息传播, 并通过端到端学习从原始数据中学习模型参数, 从而实现对推荐任务。KGCN 模型将用户和物品表示为图中的节点, 并利用图卷积操作来学习节点之间的表示。该模型通过在知识图谱中捕获用户和物品之间的复杂关系, 提高了推荐的准确性和个性化程度。同时还结合了知识图谱中的实体和关系信息, 并通过端到端的训练过程来学习节点的表示, 从而实现对用户兴趣和物品属性的有效建模。

将知识图谱作为辅助信息引入到推荐系统中, 在一定程度上缓解了推荐系统“冷启动问题”以及“数据稀疏问题”, 提高推荐的个性化程度以及准确性。然而, 引入知识图谱也面临一些挑战。然而知识图谱的构建和维护成本较高, 在聚合邻居信息的过程中, 随着路径的延伸, 计算量呈指数型增大, 不能够高效的利用全局信息。

1.2.3 基于深度学习的推荐算法

深度学习指的是人工神经网络, 因为其强大的学习能力在图像识别、语音识别、计算机视觉以及自然语言处理领域有了重大的发展。至今为止, 很多模型都成功的将深度学习运用在推荐模型上。基于深度学习的推荐算法根据场景和应用的不同分为很多类。循环神经网络 (RNN) 可以通过学习用户的行为序列模式, 能够更好地理解用户兴趣和行为规律, 从而提高推荐效果。多层感知机 (MLP) 和深度神经网络 (DNN) 通过学习用户和物品的特征表示, 可以更准确地捕获它们之间的相互作用, 从而提高推荐准确性。自注意力网络 (Self-Attention

Network) 和注意力机制推荐模型通过注意力机制, 可以更好地捕捉用户和物品之间的关联信息, 提高推荐性能。图神经网络 (Graph Neural Networks, GNN) [62] 和图卷积神经网络 (Graph Convolution Neural Networks, GCN) [63] 通过学习图结构中节点之间的关系, 可以更好地理解用户和物品之间的复杂交互, 从而提高推荐效果。

知识图谱可以看做异质结构图。GNN 和 GCN 能够有效地利用知识图谱中的结构、属性和路径信息, 通过对邻居信息的聚合学习节点的表示来捕捉实体之间的关联性和语义信息, 从而提高推荐的准确性和个性化程度。KGCN 模型就是基于 GNN 的推荐模型。KGAT^[37] (Knowledge Graph Attention Network) 结合了图注意力机制和知识图谱嵌入, 以提高推荐的准确性和个性化。

随着深度学习的兴起, 深度学习技术在推荐领域应用已经成熟。在基于深度学习技术的推荐模型中, 引入知识图谱作为辅助信息的模型也越来越多。然而基于 GNN 的模型存在一些缺陷^[38], 包括训练信号稀疏问题, 以及因为多层神经网络聚合后导致数据分布不均问题, 不利于长尾项目。针对这些问题, 本文提出了两种利用对比学习的基于 GNN 的推荐模型中的应用, 利用对比学习能够从未标记的样本中挖掘项目信息的优点, 探索对比学习方法在基于 GNN 的推荐模型的应用, 缓解训练信号稀疏问题, 同时多层神经网络中加入噪声增强, 缓解多层神经网络聚合后导致数据分布不均引发的长尾问题, 增强模型的可解释性的同时提升推荐的准确性。

1.3 论文的主要内容

针对上述问题, 依据不同的解决思路, 提出两种对比学习在基于知识感知推荐模型中的应用, 如图 1.1 所示。

(1) 针对目前基于 GNN 的知识感知推荐模型中存在的训练信号稀疏问题。提出一种基于多视图对比学习的知识感知推荐模型(MVCK)。因为对比学习拥有从未标记的原始数据中学习通用特征的能力, 挖掘数据本身监督信号, 能够有效的缓解基于 GNN 的推荐模型中存在的训练信号稀疏问题; 其次, 通过构建不同的视图更全面的学习用户/项目的表示。

(2) 针对目前基于 GNN 的知识感知推荐模型中存在数据分布不均问题导致的长尾问题。提出一种多视图噪声增强图对比学习的知识感知推荐模型(MNCL)。其主要思路是在多层神经网络中加入一定程度的噪声增强, 缓解聚合后嵌入空间节点的表示无法区分引起的长尾问题, 同时优化了对比学习正负样本

的均匀性分布，强化对比学习性能。

(3) 两种模型在 Movielens-1M, Book-Crossing 和 Last.FM 三个基准数据集上进行了全面的性能测试，然后进行了超参实验以及消融实验，验证了本文提出的模型具有优秀的性能。与最新的相关模型进行实验比较，本文提出的两种模型在各个数据集上面均有不错的表现。

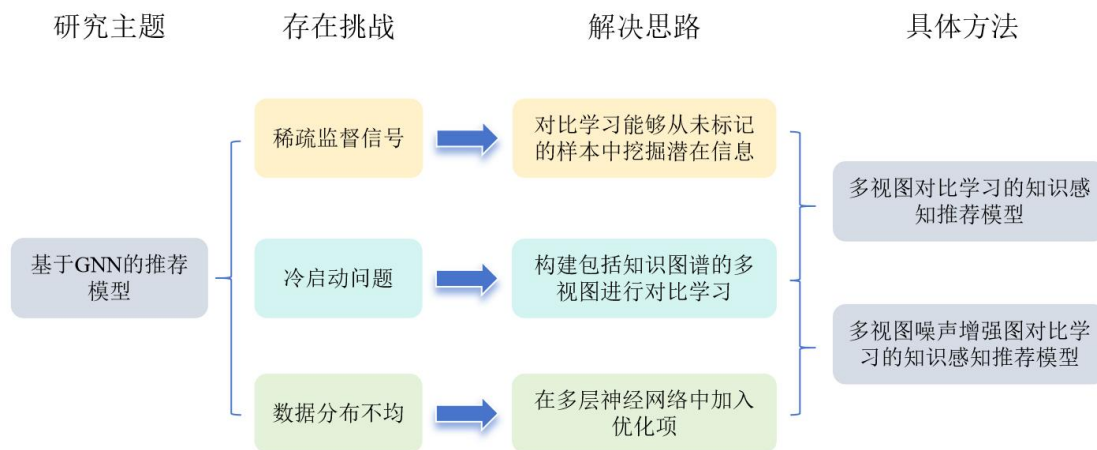


图 1.1 论文主要内容

1.4 论文的结构和框架

论文的整体框架如图 1.2 所示。论文划分为以下几个章节，具体如下：

第 1 章：绪论

绪论部分首先从整体阐述了本文的研究背景与意义；然后介绍了国内外推荐模型的现状以及发展趋势，重点介绍了三种不同阶段的推荐算法，并且说明了现存的推荐模型中面临的挑战。然后针对这些挑战，说明了本文的主要研究内容。

第 2 章：相关理论和综述

本章阐述了推荐系统目前流行的相关技术及理论。包括对知识图谱及其相关技术、图神经网络以及对比学习等技术的介绍。

第 3 章：多视图对比学习的知识感知推荐模型

本章提出了多视图对比学习的知识感知推荐模型。首先分析了基于 GNN 推荐模型现存的问题，针对这些问题，利用对比学习优秀的自监督学习方式，探索了对比学习在基于 GNN 推荐模型中的应用。接着提出了研究动机、并将定义问题，接着提出解决方案以及模型的算法框架。最后介绍实验所有的数据集以及对比的模型，并且对实验结果做出了总结和分析。

第 4 章：多视图噪声增强图对比学习的知识感知推荐模型

本章提出了多视图噪声增强图对比学习的知识感知推荐模型。首先提出了研究动机、并将定义问题，接着提出解决方案以及模型的算法框架。最后介绍实验所有的数据集以及对比的模型，并且对实验结果做出了总结和分析。

第 5 章：总结与展望

在此章节，对全文主要研究内容进行了总结，分析了此次论文中模型存在的不足以及对未来的展望。

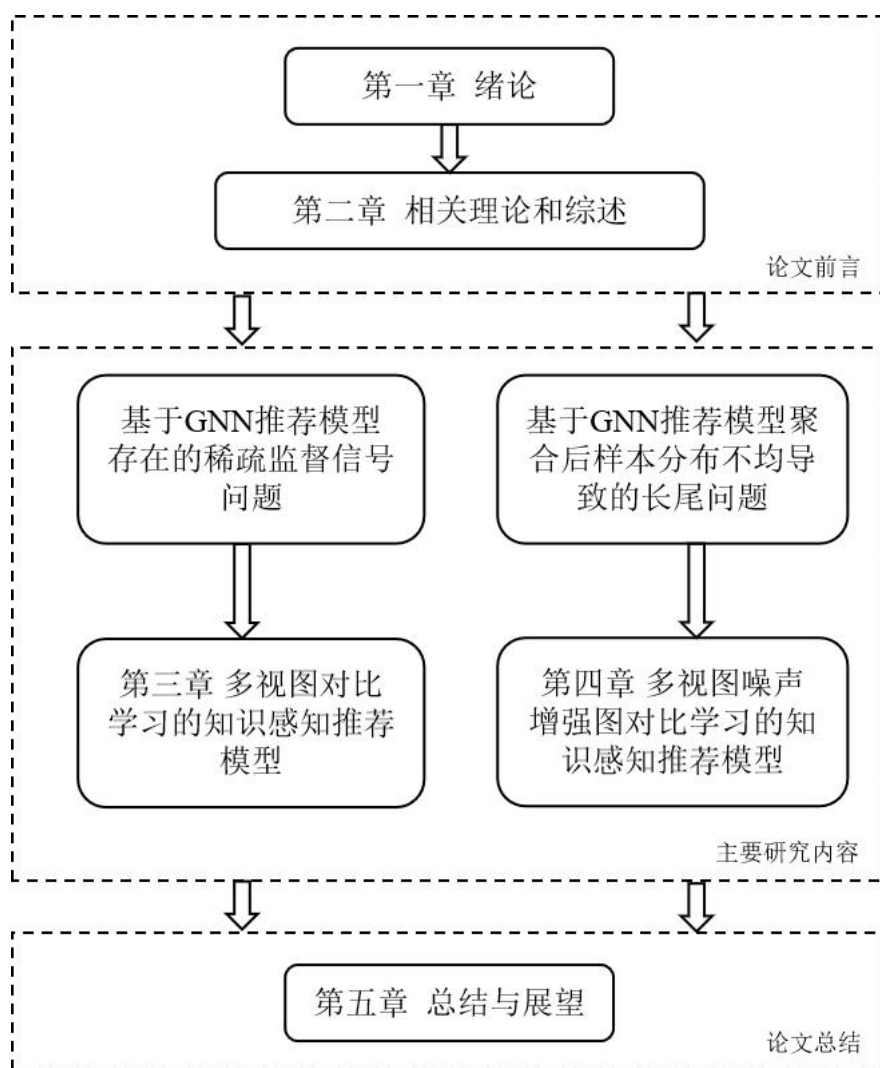


图 1.2 论文主要框架

第 2 章 相关理论和综述

绪论简单介绍了国内外推荐模型的研究现状以及发展趋势。本章将详细介绍基于对比学习的知识感知推荐模型研究中涉及到的基础概念和相关方法。本章首先重点介绍了知识图谱以及相关技术，接着对图神经网络、对比学习等技术要点的基础概念和相关应用进行了详细的陈述。

2.1 知识图谱及其相关技术

传统协同过滤的推荐算法往往因为数据稀疏以及冷启动问题导致推荐的结果不够准确。为了缓解这些问题，知识图谱被引入推荐中。接下来首先介绍知识图的概念以及定义。

2.1.1 知识图谱概念及定义

知识图谱 (Knowledge Graph, KG) 是一种高效的知识存储、组织和管理工具，能直观的将现实世界的概念以及关系结构化地描述出来。它包含实体、关系和其他相关信息，以便人类更容易理解知识之间的联系。在知识图谱中，实体被表示为节点，关系则是连接这些节点的边。核心是由三元组 (即头实体、关系、尾实体) 组成，例如 (湖北, 省会, 武汉)。谷歌创建的 FreeBase^[64] 就是一个庞大的知识图谱。谷歌利用知识图谱建立实体对象之间的联系，以更好地理解用户的搜索意图，增加用户的搜索体验，并将搜索结果以结构化的方式呈现，帮助用户发现新的知识，优化搜索结果。知识图谱可表示为：

$$\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}) \quad (2-1)$$

其中 \mathcal{G} 代表知识图谱， \mathcal{E} 代表实体集合， \mathcal{R} 代表关系集合， \mathcal{T} 代表知识三元组集合。三元组 \mathcal{T} 的定义为 $\{(h, t, r) | h, t \in \mathcal{E}, r \in \mathcal{R}\}$ ，其中 h, t, r 分别表示头实体，尾实体以及关他们之间的关系。

知识图谱在推荐模型的表现形式，主要为三种：基于知识图谱嵌入、基于路径以及基于混合方法。

2.1.2 基于嵌入的推荐模型

常见的知识图谱嵌入方法通常利用知识图谱中的实体和关系来生成低维向量表示，以便用于丰富推荐系统中的用户或项目信息，并同时保存图的结构信息，以便后续操作，这类模型的主要包括图嵌入和推荐两个模块。前者用于对知识图

谱的表示学习，其中知识图谱表征（KGE）算法是常用的方法之一，它将知识图谱中的实体转换为表征嵌入，并将这些嵌入向量应用于模型中。这类推荐模型可以分为依次学习、联合学习以及交替学习三种模型，如图 2.1 所示。

根据算法的思想不同，经典的知识图谱的表示学习主要分为两类：翻译距离模型和语义匹配模型。翻译距离模型通过将知识图谱中的实体和关系表示为低维向量，并利用它们之间的距离来捕捉实体之间的语义关系。这些模型尝试将知识图谱中的三元组转换成向量空间中的距离或者相似度，通常的做法是通过学习一个映射函数，将实体和关系映射到低维向量空间中。具体来说，翻译距离模型会定义实体和关系之间的转换方式，比如在 TransE 模型中，它简单地将头实体的向量与关系的向量相加来预测尾实体的向量。而在其他变体模型中，如 TransH、TransR 和 TransD，会引入额外的约束或转换矩阵来更好地捕捉实体和关系之间的语义信息。通过这种方式，翻译距离模型可以应用于诸如实体关系预测、知识推理等任务中。

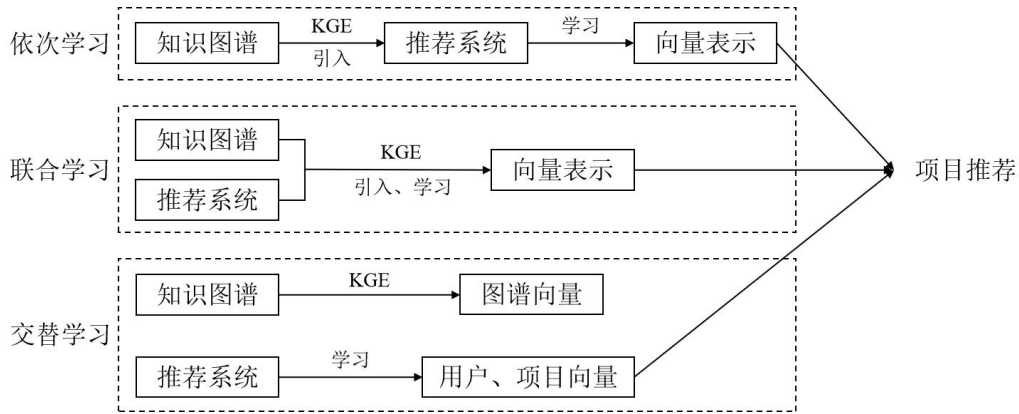


图 2.1 常见的知识图谱嵌入方法

TransE 将三元组 (h, t, r) 中的头实体 h 和关系 r 映射到同一向量空间中，使得 $h + r \approx t$ 成立。模型的打分函数为：

$$f_r(h, t) = - \|h + r - t\|_{1/2} \quad (2-2)$$

打分函数表示的是 $h + r$ 和 t 之间的距离，函数得分越大，表示距离越接近，越接近事实。由上可知，TransE 模型具有结构简单、易实现并且计算简单等优点，但是不能够处理复杂的关系，如一对多、多对一以及多对多的关系等。为了解决这些复杂的关系问题，有研究者提出了 TransH 模型。

TransH 模型通过引入关系特征向量和嵌入空间的超平面约束，可以更好地处理复杂的关系，从而提高了知识图谱表示学习的能力。在 TransH 模型中，每个知识图谱三元组的头实体 h 和尾实体 t 都被映射到与其所属关系 r 对应的超平面

M_r 上。对于每个关系，头尾实体都通过该关系对应的超平面进行映射，得到对应的特征向量。这些特征向量分别表示头尾实体在关系超平面上的投影，符号为 h_{\perp} 和 t_{\perp} ，再训练使得 $h_{\perp} + d_r \approx t_{\perp}$ 。最终，不同的实体映射到不同的关系法平面内，都具有不同的语义含义。其中头尾实体沿着法线 ω_r 在关系超平面 M_r 上的映射为 h_{\perp} 和 t_{\perp} 。

TransH 只能将实体跟关系映射到同一语义空间中，但是实体可能有很多不同的关系，不同的关系具有不同的语义关系。TransR 基于不同的关系，每种关系对应着不同的关系矩阵 M_r ，通过关系矩阵 M_r 将实体映射在了不同的关系空间上，头尾实体映射后的特征表示为 h_r 和 t_r 。进一步的，TransD 认为 TransR 中的映射关系对应的关系矩阵 M_r 应该由实体和关系两者共同决定。

Wang^[39]等人提出的 DKN (Deep Knowledge-Aware Network) 在新闻推荐中运用了实体特征和标题特征。该模型利用了实体特征和标题特征，并运用了 TransD 算法从知识图谱中获取特征。在推荐过程中，DKN 利用 TransD 算法获得的知识图谱特征，结合用户和候选新闻之间的关系，计算候选新闻对用户的注意力系数，进而形成了最终的新闻表征，提供更加个性化的新闻推荐服务。而 CKE^[55]则是将实体特征、文本特征以及图像特征融合成项目特征，通过协同过滤进行推荐。

基于语义匹配模型通过将实体和关系映射到连续的语义空间中，并通过语义匹配来学习实体之间的语义关系。常见的语义匹配模型包括 DistMult、RESCAL^[40]、HolE^[41]等。

2.1.3 基于路径的推荐模型

基于路径的推荐模型将用户和项目的历史交互构建成异质的信息网络 (Heterogeneous information networks)^[42]，用户和项目作为网络中的结点，而不同结点的连边表示用户和项目之间的潜在特征信息。其核心思想是通过考虑用户与物品之间的连通的实体语义相似性进行推荐。该类模型利用图结构来表示用户行为数据，其中节点表示用户和物品，边表示用户与物品之间的交互关系或路径。在知识图谱中，元路径是指图中连接用户和物品的不同类型的路径，例如用户-物品-用户 (U-I-U) 和用户-物品-类别-物品 (U-I-C-I)。Hete-MF^[43]模型通过分析和利用不同类型的元路径 (Meta-path)，来捕捉用户和物品之间的复杂关系，从而进行推荐。这类模型适用于处理具有丰富关系类型的异质信息网络，如社交网络、电子商务平台等。

还有的模型是基于嵌入的方法,不需要手动定义元路径,通过学习知识图谱中用户-项目或者项目-项目之间的路径信息,然后将路径信息嵌入到低维向量空间中,从而实现对用户行为的表示和建模。通过学习用户行为的嵌入向量,基于路径嵌入的方法能够更好地捕捉用户的兴趣和行为模式,从而进行推荐。 $RKGE^{[44]}$ 模型可以自动挖掘实体和项目之间具有不同语义的路径,随后使用 RNN 技术编码这些路径,并通过池化操作找到不同向量的重要特征。

2.1.4 基于混合的推荐模型

基于嵌入的推荐方法和基于路径的推荐方法在推荐模型中都取得了显著进展。然而,它们各自存在一些局限性,无法充分利用知识图谱中的丰富信息。因此,许多研究者尝试将这两种方法的优点相结合,以获得更为有效的推荐结果。这种基于混合的推荐方法首先利用知识图谱中的传播机制获取用户偏好信息,然后利用图嵌入技术进行用户偏好的特征学习,最终利用推荐模块完成推荐。

基于混合的推荐模型包括由 Wang 等人提出的 RippleNet 等。其核心思想是利用图结构中节点之间的传播过程来进行推荐,类似于信息在社交网络中的传播。RippleNet 的推荐过程可以分为两个阶段:路径传播和推荐生成。在路径传播阶段, RippleNet 利用图结构中节点之间的传播过程来捕捉用户和物品之间的关系。具体来说,它通过随机游走的方式在图中进行路径传播,获取用户与物品之间的关联信息。在推荐生成阶段, RippleNet 利用路径传播阶段获取的关联信息,结合用户和物品的特征,生成最终的推荐结果。举例说明,对于集合中的每一个三元组 (h, t, r) , 用 $h * r$ 与项目 v 相乘得到项目 v 与每个 (h_i, r_i) 的相关性得分,最后对得分进行归一化得到 p_i 。

2.2 图神经网络

知识图谱推荐(知识感知推荐)主要任务是如何将知识图谱高效的应用于推荐中,丰富用户/项目节点的学习表示,然而上述大部分方法以及模型对知识图谱的利用率极低。如何更加高效的利用知识图谱中丰富的辅助信息变得极为重要。

随着技术迭代更新,神经网络被广泛运用于各个领域。图结构包含丰富的信息,其中元素以节点表示,关系以边表示。很多真实场景中的任务都可以抽象为图结构用来推理。图神经网络就是通过神经网络利用图结构更好的学习特征表示,在推荐领域引起了广泛的关注。知识图谱作为图结构的一种,图神经网络极大提升了知识感知推荐模型的性能。根据图神经网络在应用中的实现方式不同分为不同类型,其中最主要的包括图卷积网络(GCN)以及图注意力网络(Graph Attention

Networks, GAT) [65]等。

2.2.1 图神经网络的定义

图神经网络 (GNN) 能够直接处理图结构数据, 对节点和边进行特征学习和信息传递, 从而实现对整个图的表示学习和预测任务。图神经网络的核心思想是通过迭代地聚合节点周围的邻居信息来更新节点的表示, 使得节点的表示能够综合考虑其局部邻居的信息, 并反映出节点在整个图中的语义含义。具体而言, 图神经网络通常包括消息传递 (Message Passing) 和节点更新 (Node Updating) 两个主要步骤: 在消息传递阶段, 节点会从其邻居节点接收消息, 并根据接收到的消息更新自己的表示; 在节点更新阶段, 节点根据自身的特征和邻居节点传递的消息, 更新自己的表示。通过多轮消息传递和节点更新, 图神经网络能够逐步融合全局和局部信息, 从而实现对整个图的高效表示学习和预测。图神经网络已被广泛应用于各个领域 (包括推荐领域), 并取得了显著的成果。

以图 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 为例, 其中 \mathcal{V} 是节点的集合, \mathcal{E} 是边的集合。图神经网络的消息传递和节点更新传播公式如下所示:

$$\mathbf{n}_e^{(l)} = \text{aggregate}(\mathbf{h}_e^{(l)}, \forall e \in \mathcal{N}_v) \quad (2-3)$$

$$\mathbf{h}_v^{(l+1)} = \text{update}(\mathbf{h}_v^{(l)}, \mathbf{n}_e^{(l)}) \quad (2-4)$$

其中, $\mathbf{n}_e^{(l)}$ 表示第 l 层的节点 e 的表示, $\text{aggregate}(\cdot)$ 是聚合操作, $\text{update}(\cdot)$ 是更新操作。 \mathcal{N}_v 表示节点 v 的领域, 其中 $\mathcal{N}_v = \{u \in \mathcal{V} | (u, v) \in \mathcal{E}\}$ 。

在 GNN 的众多衍生模型中, GCN 是运用最广泛的一种。GCN 的核心思想是利用节点的邻居信息来更新节点的表示, 从而实现对整个图的表示学习和预测任务, 被广泛应用于图形数据的学习和推理任务中。GCN 的主要结构包括输入层、图卷积层以及输出层。GCN 最核心的部分为卷积层。输入为邻接矩阵 \mathbf{A} 和特征矩阵 \mathbf{X} 。图卷积层执行 GCN 的主要任务消息聚合以及节点更新。图卷积层中第 l 层的传播公式如:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}^{(l)}) \quad (2-5)$$

其中, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, \mathbf{I} 是单位矩阵, $\tilde{\mathbf{D}}$ 是 $\tilde{\mathbf{A}}$ 的度矩阵。 \mathbf{H} 是每一层的特征, 下一层的输入 \mathbf{X} 就是上一层的输出 \mathbf{H} 。 σ 是非线性激活函数。

GCN 的输出层通常是一个全连接层, 用于将 GCN 中学习到的节点表示映射到目标任务的输出空间。这一层通过线性变换和非线性激活函数将节点表示转换

为最终的预测结果，输出每个节点在每个不同类别上的分布概率结果以完成不同的图数据挖掘或预测任务。

2.2.2 基于图神经网络的推荐模型

近年来，随着图神经网络（Graph Neural Networks, GNN）在深度学习领域的兴起，图数据挖掘和分析成为了研究的热点之一。图神经网络以其对图结构数据的高效处理能力和强大的表示学习能力，在推荐系统领域取得了显著的成果。在推荐领域，传统的基于矩阵分解的方法往往忽视了用户与物品之间的复杂交互关系，无法充分挖掘用户行为数据中的潜在信息。为了更好地解决这一问题，研究人员开始探索将图神经网络引入到推荐系统中，形成了一系列基于图神经网络的推荐模型。这些模型通过充分利用用户和物品之间的图结构信息，能够更加准确地捕捉用户的兴趣和物品之间的关联性，从而提高了推荐系统的性能和效果。

在推荐中，大多数数据都可以看做是基于图结构的。比如用户/项目的交互数据可以表示为包含用户以及用户节点的交互图。其他的还包括序列图、社交关系图以及知识图谱等。由于推荐领域中的数据结构都可以被看做图结构。基于GNN的推荐模型取得了巨大成功，其中包括由Wang等人提出的KGCN模型，KGCN模型利用图卷积神经网络（GCN）来汇聚与节点关联的知识图谱的邻居信息，同时计算实体之间的关系来衡量这段关系对于用户的重要性，以此赋予实体传播过程中的权重。通过多次迭代，KGCN获取知识图谱的高阶关系，从而丰富项目的潜在特征表示。最后，KGCN将项目特征与用户特征进行内积操作，以获取最终的预测概率。对于三元体 (u, r, v) 关系对于用户的得分重要性公式如下：

$$\pi_r^u = g(u, r) \quad (2-6)$$

函数 $g(\cdot)$ 表示内积操作。 π_r^u 表示关系 r 对于用户 u 的重要性得分。

为了描述项目 v 的拓扑邻近结构，项目 v 的邻居线性组合计算公式为：

$$v_{\mathcal{N}(v)}^u = \sum_{e \in \mathcal{N}(v)} \tilde{\pi}_{r_{v,e}}^u \mathbf{e} \quad (2-7)$$

其中 \mathbf{e} 是项目 v 的邻居节点的向量表示。 $\tilde{\pi}_{r_{v,e}}^u$ 为归一化得分，计算公式为：

$$\tilde{\pi}_{r_{v,e}}^u = \frac{\exp(\pi_{r_{v,e}}^u)}{\sum_{e \in \mathcal{N}(v)} \exp(\pi_{r_{v,e}}^u)} \quad (2-8)$$

基于GNN的推荐模型除了KGCN，还包括KGAT、GraphSAGE^[45]等。这些模型将深度学习运用在推荐领域上都取得了巨大的成功，都很好的扩展了用户的

偏好以及丰富了项目的特征，在模型性能上都有着不小的提升，在推荐领域做出了巨大的贡献。

2.3 对比学习

深度学习拥有在大量数据中自动学习的能力，各种领域广泛应用，例如计算机视觉和自然语言处理等。然而传统的监督学习模型极度依赖于大量的有标签数据，然而在实际场景中，交互数据往往是非常稀疏的。因此相关工作者想研究出一种新的方法，如何能够利用大量的无标签数据。因此自监督学习（Self-supervised learning, SSL）得到了广泛的应用，自监督学习可以从数据本身中寻找标签来监督模型的训练。这种不需要大量标签的训练范式在各个领域引起了大量的关注。自监督学习在训练过程中为数据生成额外的监督信号（伪标签），然后对原本稀疏的数据进行充分的学习表示，继而用到下游的任务中去。无监督学习根据实现方式跟架构不同，可以分为生成式学习以及对比式学习。其中，对比学习（Contrastive Learning, CL）作为自监督学习中重要的组成部分，在推荐领域取得了巨大的成功，很好缓解数据稀疏产生的问题。对比学习的思想是：将与样本相似的样本视为正样本，与样本不相似样本视为负样本，最大化正样本之间的一致性，最小化负样本之间的相似性。

2.3.1 对比学习框架与步骤

如图 2.2 所示。样本是一组动物照片，包括狗的图片以及猫头鹰图片（代指其他动物的所有图片）。接下来将通过下图介绍对比学习的一般步骤。对比学习过程可以简单的描述为三个基本步骤：

（1）数据增强：首先对图中狗的图片 I^a 进行数据增强（包括旋转、裁剪、重新着色、调整大小等等），获得了正样本 I^+ ，猫头鹰的图片则为负样本 I^- 。

（2）特征学习：将数据增强后的图片输入到深度学习模型中，获得每个样本的特征向量表示，样本 I^+ 通过 θ 获得 f^+ ，样本 I^a 通过 θ 获得 f^a ，样本 I^- 通过 θ 获得 f^- 。

（3）损失最小化：通过最小化对比损失函数 InfoNCE 来最大化正样本两个向量的相似性。满足式子：

$$\delta(f^a, f^+) \ll \delta(f^a, f^-)$$

其中函数 δ 表示样本之间的距离公式。一般采用欧几里得距离，余弦相似度。

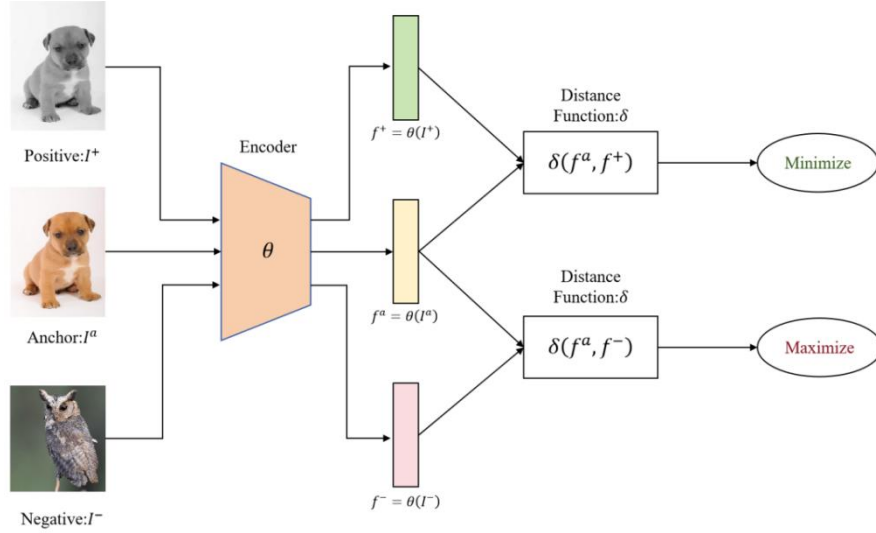


图 2.2 对比学习基本步骤

2.3.2 对比学习在推荐系统中应用

对比学习因为自监督学习方法在其他领域取得了巨大的成功，近几年，在推荐领域也备受注。对比学习通过比较正样本和负样本之间的相似性来学习模型，从而提高推系统的性能。本文将从对比学习的基本概念出发，介绍其在推荐系统中的应用及相关方法。

首先，对比学习是一种自监督学习方法，它不需要标记的样本，而是通过最大化正样本与负样本之间的差异来学习模型。在推荐系统中，正样本通常表示用户与其喜欢的物品之间的交互，负样本则表示用户与未喜欢的物品之间的交互或者是与其他用户的交互。通过对比学习，即使交互数据仍然稀疏，推荐系统仍然可以很好的理解用户的兴趣和偏好，学习到更接近事实的特征表示，提高推荐的准确性和个性化程度。

SGL (Self-supervised Graph Learning) ^[46]模型是最新的基于对比学习的推荐模型。SGL 模型旨在通过自监督学习任务来增强图卷积网络 (GCN) 在推荐任务中的性能。其核心思想是将经典的推荐监督任务与辅助的自监督任务相结合，通过自我区分来加强节点表示学习。SGL 通过生成节点的多个视图，并最大化同一节点的不同视图之间的一致性。为了生成这些视图，SGL 引入了三种操作：节点丢弃、边丢弃和随机游走，以便以不同的方式改变图的结构。通过这种方式，SGL 能够自动挖掘困难的负样本，从而提高了推荐准确性，尤其是在长尾物品方面，同时也提高了模型对于交互噪声的鲁棒性。

SGL 提出的三种操作：节点丢弃、边丢弃和随机游走被应用于各种主流的

推荐模型中。这三种方法主要是为了创建节点的不同视图。这些操作可以统一表示为：

$$\mathbf{z}_1^{(l)} = H(\mathbf{z}_1^{(l-1)}, s_1(\mathcal{G})), \mathbf{z}_2^{(l)} = H(\mathbf{z}_2^{(l-1)}, s_2(\mathcal{G})), s_1, s_2 \sim \mathcal{S} \quad (2-9)$$

首先在图 \mathcal{G} 上独立应用两个随机选择 s_1 和 s_2 ，并建立节点 $\mathbf{z}_1^{(l)}$ 和 $\mathbf{z}_2^{(l)}$ 的两个相关视图。

建立了节点的增强视图后，将相同节点的视图视为正样本对（即 $\{(\mathbf{z}_u', \mathbf{z}_u'') | u \in \mathcal{U}\}$ ），将不同节点的视图视为负样本对 $\{(\mathbf{z}_u', \mathbf{z}_v'') | u, v \in \mathcal{U}, u \neq v\}$ 。正样本对的辅助监督鼓励了同一节点不同视图之间的一致性，而负样本对的监督则强制了不同节点之间的不一致性。形式上，遵循 SimCLR^[47]并采用对比损失 InfoNCE^[48]，以最大化正样本对的一致性并最小化负样本对的一致性：

$$\mathcal{L}_{ssl}^{user} = \sum_{u \in \mathcal{U}} -\log \frac{\exp(s(\mathbf{z}_u', \mathbf{z}_u'')/\tau)}{\sum_{v \in \mathcal{U}} \exp(s(\mathbf{z}_u', \mathbf{z}_v'')/\tau)} \quad (2-10)$$

其中， $s(\cdot)$ 表示两个向量之间的相似度，设置为余弦相似度函数； τ 是超参数，表示 softmax 中的温度参数。类似地，可以得到物品的对比损失。将这两个损失结合起来最终得到了自监督任务的目标函数。

对比学习因为其优秀的自监督学习能力被广泛运用在最新的推荐模型中。Lin 等人^[66]将潜在的结构邻居节点和语义邻居节点纳入对比学习中来获取高阶邻居的结构信息和语义信息。SEPT^[67]提出了挖掘用户的社交信息来增强用户的数据视图从而来进行社交感知的对比学习方法框架。考虑到在推荐系统场景中，对比损失函数相较于数据增强的方式对性能的影响更关键，XSimGCL 模型^[71]并没有依赖于对图数据的增强来提高模型准确性，而是专注于优化对比损失函数，从而减少表示分布的偏差，提高了推荐模型的性能。

2.4 本章小结

在本章中，首先描述了知识图谱的概念及其相关技术。之后从基于知识图谱嵌入的方法、基于路径的推荐模型以及基于混合的推荐算法三个方面介绍了知识图谱在推荐模型中的应用。接着，概述了图神经网络的定义以及基于图神经网络的推荐模型。最后介绍了最近大热的对比学习方法，以及对比学习方法在推荐模型中常用几种数据增强方法，包括节点丢失，边丢失以及随机游走三种方法。这些内容为后面的研究工作提供了理论基础。

第3章 多视图对比学习的知识感知推荐模型

传统的协同过滤推荐算法在融入知识图谱丰富的信息之后有效的缓解了数据稀疏性和冷启动问题，然而大部分方法对知识图谱的利用率很低。因此基于知识图谱的推荐模型的主要任务是将知识图谱如何更加高效的运用到推荐模型中。随着图神经网络（GNNs）在各个领域获得巨大成功，其优秀的高阶结构信息捕获能力在知识图谱推荐领域也备受关注，已成为基于知识图谱推荐的主流方向。但是由于交互数据的极度稀疏得问题一直存在，基于 GNN 的知识感知推荐模型依然存在着训练信号稀疏问题。最近，基于自监督学习方式的对比学习由于能够从未标记的原始数据中学习特征向量，即从数据本身挖掘监督信号，在各个领域取得了巨大的成功。利用对比学习这一特点，本文提出了一种基于多视图对比学习的知识感知推荐模型（MVCK）。MVCK 模型构建了不同的视图用来对比学习，其中包括用户与项目的视图，以及项目与知识图谱中实体的视图。接着在多视图的基础上，运用对比学习中边丢失、节点丢失等方法构建了不同程度的对比图进行对比学习，以对比学习自监督的学习方式挖掘不同视图中特征结构信息。最后，MVCK 模型在三个基准数据集 MovieLens、Last.FM 和 Book-Crossing 上面进行了大量的实验，验证了模型的有效性。

3.1 研究动机

随着日常数据的剧增，信息过载日益严重。性能更加优秀的个性化推荐系统成为用户发现兴趣项目的关键。传统的推荐算法按照计算方式的不同，主要分为下面这三类：基于内容的推荐算法、基于协同过滤的推荐算法、基于混合的推荐算法。协同过滤算法从发起至今在推荐领域广受欢迎，早已经被广泛应用于各个现实领域，推动了推荐系统的重大进展。协同过滤算法假设相似的用户对项目都有着相似的兴趣，传统的协同过滤算法依赖于用户-物品的交互数据来进行推荐。然而，用户-项目交互的稀疏性从根本上限制了传统的推荐算法性能的改进。为了解决这个问题，很多研究将知识图谱作为侧信息引入到推荐模型中，用来充分的学习用户和项目的特征表示，来提高推荐的质量以及个性化程度。

基于知识图谱的推荐主要包括基于嵌入的方法和基于路径的方法。基于知识图谱嵌入的方法将高维的知识图谱三元组中的实体以及关系映射成低维的向量，一些研究^[49]将基于转换的知识图谱嵌入纳入到物品嵌入中去，如 TransR^[29]，来丰富用户跟物品的特征表示。基于路径的推荐模型将用户和项目的历史交互构建成交异的异质信息网络，研究集中于从知识图谱中提取具有语义意义的元路径，并沿

着这些元路径对用户和项目进行复杂的建模。然而, 这些方法均没有很好的表现, 因为它们独立地处理每个项目关系以进行学习。因此, 学习过程无法提取足够的协作信号。随着图神经网络 (GNN) 的产生以及发展, GNN 在推荐算法和知识图谱中的应用也越来越广泛, 基于 GNN 的推荐模型能够有效地利用知识图谱中的图结构信息, 从而能够有效挖掘图数据之间的高阶连通性信息, 实现更加准确的个性化推荐。

图卷积神经网络将基于嵌入的方法和基于路径的方法合理的利用起来, 通过在知识图谱上的传播和聚合来捕获多跳高阶信息。尽管 GNN 在这方面取得了显著的成果, 但仍然存在训练信号稀疏的问题, 特别是在极端稀疏的用户-项目交互情况下。这种极端的稀疏性还可能导致生成的节点表示无法区分, 造成更加严重的退化问题^[59]。缓解训练信号稀疏问题面临着一个重大的挑战, 即训练标签的不足, 因为在真正的推荐系统中标签通常是很少的。一个经典的自我监督学习 (SSL) 方法—对比学习, 提出了一种不需要明确标签的模型训练方法^[50], 通过最大化负样本之间的距离而最小化正样本之间的距离, 从未标记的样本数据中学习出具有区分性的嵌入表示。对比学习这种能够从未标记的原始数据中学习特征向量、从数据本身挖掘监督信号的能力很好的缓解了基于 GNN 推荐模型存在的训练信号稀疏问题。基于此本章提出了基于多视图对比学习的知识感知推荐模型 (MVCK)。MVCK 模型主要是探索对比学习在基于 GNN 的知识感知推荐模型中的应用。利用对比学习解决基于知识感知推荐模型存在的训练信号稀疏问题, 并且探索更加有效的不同视图对比方法。本章的主要贡献如下:

(1) 提出了一种新的推荐模型 MVCK, 该模型运用对比学习方法有效缓解了基于知识感知推荐模型存在的训练信号稀疏问题, 并且探索了不同视图之间的对比方法。

(2) 构建了对比学习需要的不同子图, 并通过不同的 GNN 来学习用户和项目的嵌入表示。最后设计了多负例对比学习来充分学习不同子图。

(3) 在 Movielens、Last.FM 和 Book-Crossing 三个数据集上进行了充分的实验, 验证了模型的有效性。

3.2 问题定义

本节首先介绍了两种关键的结构化数据, 即用户-物品交互矩阵和知识图谱。随后, 描述了基于 GNN 的知识感知推荐模型的推荐任务。

3.2.1 用户-相互交互数据

在典型的推荐场景中，定义用户和物品的集合分别为 $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$ 和 $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ ，其中 M 为用户数量， N 为物品数量。用户-物品交互矩阵 $\mathbf{Y} \in \mathbb{R}^{M \times N}$ 根据用户的隐式反馈定义，其中 $y_{uv} = 1$ 表示用户与物品进行了互动，例如点击或购买等行为；否则 $y_{uv} = 0$ 。

3.2.2 知识图谱

在知识图谱方面，除了考虑历史交互外，现实世界中与物品相关的信息（例如物品属性或其他的常识知识）以异构图的形式存储在知识图谱中。设 $\mathcal{G} = \{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{R}\}$ 为知识图谱。其中， \mathcal{E} 表示实体集合， $\mathcal{E} = \{e_1, e_2, \dots, e_n\}$ ； \mathcal{R} 表示关系集合， $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$ 。 (h, r, t) 是单个三元组， h, r, t 分别代表知识三元组的头实体、关系和尾实体，例如，三元组（《惊天魔盗团》，导演，易斯·卡卡杜瑞尔）表示易斯·卡卡杜瑞尔是电影《惊天魔盗团》的导演。在许多推荐场景中，物品 $v \in \mathcal{V}$ 对应于知识图谱中的实体 $e \in \mathcal{E}$ 。例如，在电影推荐中，“蜘蛛侠”这个物品也在知识图谱中以相同的名称作为一个实体出现。因此，本章建立了一个物品-实体对齐的集合 $\mathcal{A} = \{(v, e) | v \in \mathcal{V}, e \in \mathcal{E}\}$ ，其中 (v, e) 表示物品 v 可以与知识图中的实体 e 对齐。通过物品和知识图实体之间的对齐，知识图能够为交互数据提供补充信息并对物品进行描述。

3.2.3 基于 GNN 的知识感知推荐模型的推荐任务

在给定用户-物品交互矩阵 \mathbf{Y} 和知识图 \mathcal{G} 的情况下，本章的目标是学习一个函数 $\hat{y}_{uv} = f(u, v)$ ，该函数的目标是预测用户 u 对于项目 v 的感兴趣程度。

3.3 算法框架

本章介绍提出的 MVCK 模型。MVCK 旨在将自监督学习融入知识感知推荐，以改进用户/物品表示学习。图 3.1 展示了 MVCK 的模型图，整个框架主要由四个部分模块组成：不同子图构建模块、消息聚合模块、对比学习模块以及预测模块。

（1）不同子图构建模块：在此模块中，分别对用户-项目的交互图以及项目-实体视图分别进行不同程度的节点以及边的丢失，构建不同的子图。

(2) 消息聚合模块：将构建的不同子图通过不同的 GNN 进行消息聚合，知识图谱的子图经过关系感知 GNN 之后，还要经过多层 Light-GCN，来学习更好的项目特征表示。

(3) 对比学习模块：将学习到的用户以及项目特征分别进行多负例对比学习。从有限的交互数据中挖掘更多的信息。

(4) 预测模块：将得到的最终用户和项目特征向量进行连接，然后作内积进行最后的评分预测。

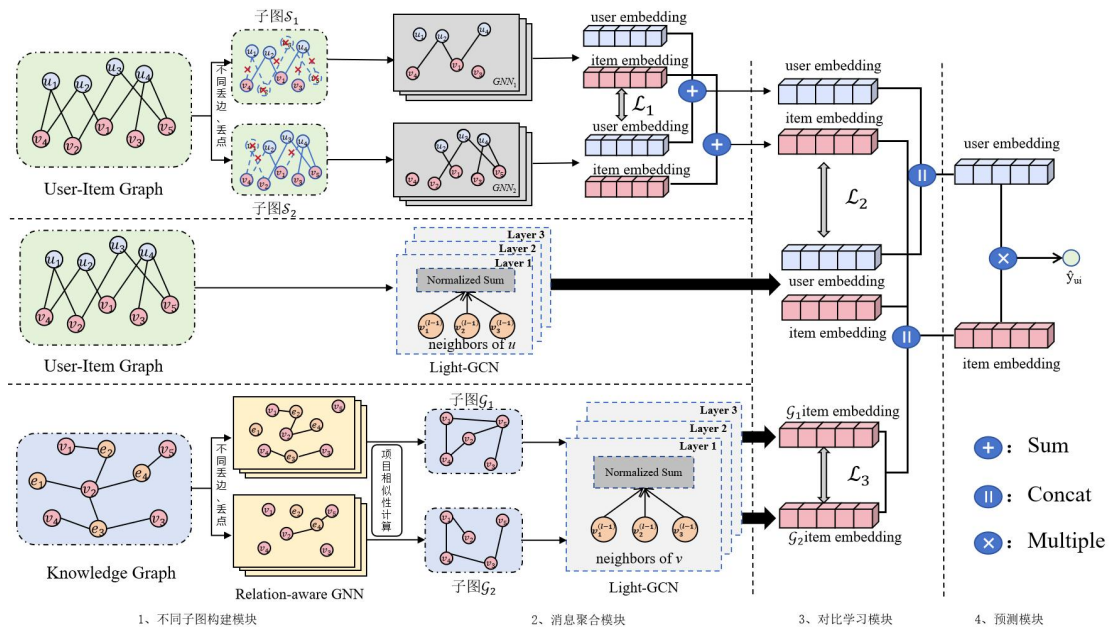


图 3.1 MVCK 模型图

3.3.1 子图构建

在数据样本中，用户跟项目的数量一般都很巨大，但是在实际场景中能够观测到的数据间的交互矩阵非常稀疏，导致挖掘数据的隐含信息非常受限。本章采用数据增强的方式来处理样本，这样可以大大的节约训练成本以及训练时间。

数据增强的方式包括节点丢失、边丢失以及随机游走。本章主要采用节点丢失以及边丢失两种方法来设计数据增强。节点丢失的方法主要是从原始的交互图中以一定比例 p 来随机删除图中的一些节点，剩余的部分节点构建出不一样的图结构来进行特征学习，边丢失的方法与节点丢失的方法类似。在 MVCK 模型中，对于用户-相互的交互图 \mathcal{S} 。运用这两种方法得到了不同的子图，具体方法如下：

$$\mathcal{S}_1 = (\mathbf{M}' \odot \mathcal{V}, \mathbf{M}_1 \odot \mathcal{E}, p_1, \rho_1), \mathcal{S}_2 = (\mathbf{M}'' \odot \mathcal{V}, \mathcal{E}, \mathbf{M}_2 \odot \mathcal{E}, p_2, \rho_2) \quad (3-1)$$

其中, \mathbf{M}_1 和 $\mathbf{M}_2 \in \{0,1\}^{|\mathcal{E}|}$ 是边缘集合 \mathcal{E} 上的两个掩码向量, $\mathbf{M}', \mathbf{M}'' \in \{0,1\}^{|\mathcal{V}|}$ 是应用在节点集 \mathcal{V} 上的两个掩码向量。 p_1, ρ_1 分别是子图 \mathcal{S}_1 上的节点丢失率以及边丢失率。 p_2, ρ_2 分别是子图 \mathcal{S}_2 上的节点丢失率以及边丢失率。

3.3.2 关系感知的 GNN

知识图谱中除了实体的信息外, 还有丰富的实体之间的关系信息。为了能够充分的利用知识图谱带来的辅助信息, 缓解原始交互图中的数据稀疏问题本章采用了本章采用了一个关系感知的 GNN 模块, 该模型在消息聚合的过程中, 不仅能够汇聚节点周围实体的信息, 同时也能够保留实体之间关系的信息。具体如下, 首先从知识图谱 \mathcal{G} 中递归地学习 K 次的项目表示, 其中第 k 次关系感知聚合机制公式为:

$$\mathbf{e}_i^{(k+1)} = \frac{1}{|\mathcal{N}_i|} \sum_{(r,j) \in \mathcal{N}_i} \mathbf{e}_r \odot \mathbf{e}_j^{(k)} \quad (3-2)$$

$$\mathbf{e}_j^{(k+1)} = \frac{1}{|\mathcal{N}_j|} \left(\sum_{(r,j) \in \mathcal{N}_j} \mathbf{e}_r \odot \mathbf{e}_j^{(k)} + \sum_{(r,i) \in \mathcal{N}_j} \mathbf{e}_r \odot \mathbf{e}_i^{(k)} \right) \quad (3-3)$$

其中, $\mathbf{e}_i^{(k)}$ 和 $\mathbf{e}_j^{(k)}$ (对于 $\forall k \in K$) 分别表示项目 i 和实体 j 的向量表示, 它们存储了 $(k-1)$ 跳邻居以及关系传播的信息。

在 K 次消息聚合后, 最终知识图谱中的邻居实体信息和关系信息都被编码到项目的特征表示中。接着, 基于余弦相似度构建了项目-实体相似性, 计算公式如下:

$$\mathcal{G}_{ij} = \frac{(\mathbf{e}_i^K)^T \mathbf{e}_j^K}{\|\mathbf{e}_i^K\| \|\mathbf{e}_j^K\|} \quad (3-4)$$

其中, $\mathbf{e}_i^{(K)}$ 和 $\mathbf{e}_j^{(K)}$ 分别表示项目 i 和实体 j 在 K 次消息聚合后的向量表示, 在知识图 \mathcal{G} 中的每项 \mathcal{G}_{ij}^j 表示项目 i 和实体 j 之间的语义相似性, $\mathcal{G}_{ij}^j=0$ 意味着它们之间没有关联。接着, 对全连接的项目-实体图进行 kNN 稀疏化^[51], 用来减少计算负担以及噪声较大和不重要的边^[52]:

$$\hat{\mathbf{G}}_{ij} = \begin{cases} \mathbf{G}_{ij}, & \mathbf{G}_{ij} \in \text{top} - k(\mathbf{G}_i) \\ 0, & \text{otherwise} \end{cases} \quad (3-5)$$

其中, $\hat{\mathbf{G}}_{ij}$ 是一个经过稀疏化的图邻接矩阵。为了缓解梯度爆炸或消失的问题, 需要对邻接矩阵进行必要的正则化处理, 具体如下:

$$\tilde{\mathbf{G}} = (\mathbf{D})^{-\frac{1}{2}} \hat{\mathbf{G}}_{ij} (\mathbf{D})^{-\frac{1}{2}} \quad (3-6)$$

其中, $\mathbf{D} \in \mathbb{R}^{N \times N}$ 是 $\tilde{\mathbf{G}}$ 的对角度数矩阵, $\mathbf{D}_{i,i} = \sum_j \hat{\mathbf{G}}_{ij}$ 。因此, 最终得到了归一化的稀疏邻接矩阵 $\tilde{\mathbf{G}}$ 。

经过上述处理之后, 得到项目-实体视图 \mathcal{G} 以及其归一化的稀疏邻接矩阵 $\tilde{\mathbf{G}}$ 。

3.3.3 Light-GCN

为了获得用户-项目视图的不同的特征向量用来进行对比学习, 本文采用了不同的图编码器学习用户和项目的特征向量。除了关系感知 GNN 外, 还主要采用了 Light-GCN^[53]。Light-GCN 包含简单的消息传递和聚合机制, 没有特征变换和非线性激活, 非常的简单有效并且减少计算成本。它递归地执行 L 次聚合, 在第 l 层, 聚合过程可以被表述如下:

$$\mathbf{e}_u^{(l+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u| |\mathcal{N}_i|}} \mathbf{e}_i^{(l)} \quad (3-7)$$

$$\mathbf{e}_i^{(l+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u| |\mathcal{N}_i|}} \mathbf{e}_u^{(l)} \quad (3-8)$$

其中, $\mathbf{e}_u^{(l)}$ 和 $\mathbf{e}_i^{(l)}$ 表示用户 u 和物品 i 在第 l 层的嵌入, \mathcal{N}_u , \mathcal{N}_i 表示用户 u 和物品 i 的邻居。在 Light-GCN 中, 共执行 L 次聚合, 每次聚合会得到一个本层的用户以及项目的特征向量。然后将不同层的表示相加, 得到最终的特征向量 \mathbf{e}_u 和 \mathbf{e}_i , 如下所示:

$$\mathbf{e}_u = \mathbf{e}_u^{(0)} + \dots + \mathbf{e}_u^{(L)}, \mathbf{e}_i = \mathbf{e}_i^{(0)} + \dots + \mathbf{e}_i^{(L)} \quad (3-9)$$

在关系感知 GNN 模块获得了项目视图 \mathcal{G} 以及其归一化的稀疏邻接矩阵 $\tilde{\mathbf{G}}$, 接着采用 Light-GCN 进行 L 次聚合操作, 来学习更好的项目表示。在每层 Light-GCN 中的第 l 次消息传递和聚合过程可以表示为:

$$\mathbf{e}_i^{(l+1)} = \sum_{j \in \mathcal{N}(i)} \tilde{\mathbf{G}} \mathbf{e}_i^{(l)} \quad (3-10)$$

其中, $\mathcal{N}(i)$ 是邻居物品, $\tilde{\mathbf{G}}$ 是公式中 (3-6) 的归一化稀疏图邻接矩阵, $\mathbf{e}_i^{(l)}$ 是物品 i 在每层 Light-GCN 中的第 l 次消息聚合后的表示。然后, 将 L 次聚合操作后的物品表示相加, 得到最终的项目特征表示 \mathbf{e}_i :

$$\mathbf{e}_i = \mathbf{e}_i^{(0)} + \dots + \mathbf{e}_i^{(L)} \quad (3-11)$$

3.3.4 多负例对比学习

经过不同的 GNN 消息集合之后, 得到了各个子图中的用户、项目向量。子图 \mathcal{S}_1 中用户、项目最终向量分别为 $\mathbf{e}_u^{\mathcal{S}_1}, \mathbf{e}_i^{\mathcal{S}_1}$, 子图 \mathcal{S}_2 中用户、项目最终向量分别为 $\mathbf{e}_u^{\mathcal{S}_2}, \mathbf{e}_i^{\mathcal{S}_2}$ 。同理, 子图 \mathcal{G}_1 中项目最终向量分别为 $\mathbf{e}_i^{\mathcal{G}_1}$, 子图 \mathcal{G}_2 中项目最终向量分别为 $\mathbf{e}_i^{\mathcal{G}_2}$, 将它们映射到计算对比损失的空间中。然后定义正样本和负样本, 对于一个视图中的任何节点, 由另一个视图学习到的相同节点嵌入形成正样本; 而在两个视图中, 除了它之外的节点嵌入自然被视为负样本, 如下图 3.2 所示。

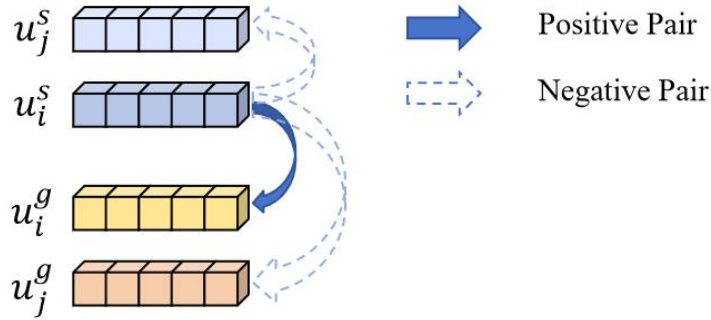


图 3.2 多负例对比学习

同时本文认为在不同视图中负样本的权重不一样, 因此定义原视图中的权重为 ω 。定义好正样本和负样本后, 本章将对比损失函数分为交互图对比 \mathcal{L}_1 以及知识图谱项目对比 \mathcal{L}_2 , 其中 \mathcal{L}_1 包含用户以及项目的对比损失, 以项目的对比损失为例, 具体如公式 (3-12) 所示:

$$\mathcal{L}_1 = -\log \frac{\exp(s(\mathbf{e}_i^1, \mathbf{e}_i^2)/\tau)}{\exp(s(\mathbf{e}_i^1, \mathbf{e}_i^2)/\tau) + \omega \sum_{i \neq j} \exp(s(\mathbf{e}_i^1, \mathbf{e}_j^1)/\tau) + \sum_{i \neq j} \exp(s(\mathbf{e}_i^1, \mathbf{e}_j^2)/\tau)} \quad (3-12)$$

其中 τ 表示温度系数。 $s(\cdot)$ 表示余弦相似度计算。表示除了以外的节点。即总体的对比损失为:

$$\mathcal{L}_{CL} = \mathcal{L}_1 + \mathcal{L}_2 \quad (3-13)$$

表 3.1 MVCK 算法流程

算法 1: MVCK 模型训练过程 (伪代码)

输入:

- 用户-物品交互矩阵 $\mathbf{Y} \in \mathbb{R}^{M \times N}$,
- 知识图谱 $\mathcal{G} = \{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{R}\}$
- 模型总训练次数 \mathcal{N}_e
- 批处理大小 \mathcal{N}_b
- 对比学习损失权重 α

- 1) 创建知识图谱项目-实体稀疏图 \mathcal{G} , 用户-项目交互稀疏图 \mathcal{S}
- 2) FOR epoch IN \mathcal{N}_e DO
- 3) 从图 \mathcal{G} 和图 \mathcal{S} 中采样批次大小的 \mathcal{N}_b 的数据
- 4) 初始化神经网络参数、用户特征 \mathbf{e}_u 以及项目特征 \mathbf{e}_i
- 5) //构建不同子图
- 6) 数据增强获得子图 $\mathcal{S}_1, \mathcal{S}_2, \mathcal{G}_1, \mathcal{G}_2$
- 7) //GNN 层消息聚合
- 8) 子图 $\mathcal{S}_1, \mathcal{S}_2 \rightarrow \mathbf{e}_u^{s1}$ 和 \mathbf{e}_u^{s2} , \mathbf{e}_i^{s1} 和 \mathbf{e}_i^{s2}
- 9) 子图 $\mathcal{G}_1, \mathcal{G}_2 \rightarrow \mathbf{e}_i^{g1}$ 和 \mathbf{e}_i^{g2}
- 10) //对比任务
- 11) $\mathcal{L}_1 \leftarrow \mathbf{e}_u^{s1}$ 和 \mathbf{e}_u^{s2}
- 12) $\mathcal{L}_2 \leftarrow \mathbf{e}_i^{s1}$ 和 \mathbf{e}_i^{s2} , \mathbf{e}_i^{g1} 和 \mathbf{e}_i^{g2}
- 13) $\mathcal{L}_{CL} = \mathcal{L}_1 + \mathcal{L}_2$
- 14) //推荐任务
- 15) $\mathbf{e}_u^{s1} \parallel \mathbf{e}_u^{s1} \parallel \mathbf{e}_u^{s2} \parallel \mathbf{e}_u^{s2} \rightarrow \mathbf{e}_u$, $\mathbf{e}_i^{s1} \parallel \mathbf{e}_i^{s2} \parallel \mathbf{e}_i^{g1} \parallel \mathbf{e}_i^{g2} \rightarrow \mathbf{e}_i$
- 16) $\mathcal{L}_{BPR} \leftarrow$ 推荐任务损失计算
- 17) //多任务学习
- 18) $\mathcal{L}_{MVCK} = \mathcal{L}_{BPR} + \alpha \mathcal{L}_{CL}$
- 19) END FOR

输出: MVCK 模型

3.3.5 模型预测

用户-项目视图、知识图谱项目-实体视图经过多层聚合后, 获得了用户多个表示 \mathbf{e}_u^{s1} 和 \mathbf{e}_u^{s2} 以及多个项目表示分别为 \mathbf{e}_i^{s1} 、 \mathbf{e}_i^{s2} 、 \mathbf{e}_i^{g1} 和 \mathbf{e}_i^{g2} 。将这些优化后的特征向量进行串联操作, 得到最终的用户和项目的特征向量, 并通过预测函数获得最

终的预测评分，如下所示：

$$\mathbf{e}_u = \mathbf{e}_u^{s1} \|\mathbf{e}_u^{s1}\| \mathbf{e}_u^{s2} \|\mathbf{e}_u^{s2}\|, \mathbf{e}_i = \mathbf{e}_i^{s1} \|\mathbf{e}_i^{s1}\| \mathbf{e}_i^{g1} \|\mathbf{e}_i^{g1}\| \mathbf{e}_i^{g2} \|\mathbf{e}_i^{g2}\| \quad (3-14)$$

$$\hat{y}(u, i) = \mathbf{e}_u^T \mathbf{e}_i \quad (3-15)$$

本章采用了多任务训练策略来优化整个模型。为了重构历史数据，鼓励用户历史项目的预测得分高于未观察到的项目，本章采用了 BPR 损失：

$$\mathcal{L}_{BPR} = \sum -\log \sigma(\hat{y}(u, i^+) - \hat{y}(u, i^-)) \quad (3-16)$$

其中， i^+ 表示训练中观察到的与用户 u 有交互的项目， i^- 表示训练中未观察到的项目。 σ 是 sigmoid 函数。通过将对比损失与 BPR 损失相结合，优化以下目标函数来学习模型参数：

$$\mathcal{L}_{MVCK} = \mathcal{L}_{BPR} + \alpha \mathcal{L}_{CL} + \lambda \|\Theta\|_2^2 \quad (3-17)$$

其中， Θ 是模型参数集， α 和 λ 是两个超参数，分别用于控制对比损失和 L_2 正则化项。

3.3.6 算法流程

MVCK 推荐算法如表 3.1 所示。

3.4 实验设置

3.4.1 实验数据集

本章算法采用的数据集分别为 Movilens-1M 数据集、Book-Crossing 数据集和 Last.FM 数据集，如表 3.2 所示：

1. MovieLens-1M 数据集。该数据集广泛应用于推荐系统研究中，其中包含了一百万个用户对电影的评分数据。该数据集包含了电影评分、电影元数据以及用户属性等信息。每个用户对多部电影进行了评分，并且用户和电影之间的关系是显式地表示的。

2. Book-Crossing 数据集。该数据集包含了来自 Book-Crossing 网站的用户对图书的评分数据。它包括了来自 17860 位用户对 14967 本书籍的 139746 个评分。除了评分数据外，该数据集还包括了有关书籍和用户的一些元数据信息，如书名、作者、出版社以及用户的年龄、地区等信息。

表 3.2 实验数据集的基本统计数据

	Movilens-1M	Book-Crossing	Last.FM
用户数量	6036	17860	1872
项目数量	2445	14967	3846
交互数量	753772	139746	42364
实体数量	182011	77903	9366
关系数量	12	25	60
三元组数量	1241996	151500	15518

3. Last.FM 数据集。该数据集由 Last.FM 音乐社交网站提供，包括了用户以及他们播放的音乐曲目和播放次数。除此之外，Last.FM 数据集还包含了有关音乐曲目的元数据信息，如音乐标题、艺术家、流派等。

由于 MovieLens-1M、Book-Crossing 和 Last.FM 数据集中的用户与项目的交互数据是显式反馈的形式，本章节采用了 RippleNet 的方法将其转换为隐式反馈。在这种隐式反馈中，标签为 1 表示用户对项目的反馈是正向的，标签为 0 表示反馈是负向的。对于 MovieLens-1M 数据集，本文将评分阈值设为 4，即用户对项目的评分超过 4 则被标记为正样本（标签为 1）。然而，由于 Last.FM 和 Book-Crossing 数据集较为稀疏，无法设置评分阈值。因此，任何用户与项目的交互记录（如播放音乐或阅读书籍）都被视为正样本（标签为 1），否则为负样本（标签为 0）。对于负样本，针对每个用户，本章从他未观看或未播放的项目中进行随机抽样，抽样数量与他的正样本数量相等。

本章还利用 FreeBase 构建知识图谱，具体过程如下：首先，根据数据集中项目的 ID 与 FreeBase 中的 ID 进行配对，然后筛选出与这些项目相关的子知识图谱。在筛选过程中，排除了涉及多个实体或无法匹配到实体的项目。接着，将项目的 ID 与子知识图谱中的三元组头实体的 ID 进行匹配，并从中选择已匹配的三元组。这三个数据集的基本统计数据如表 3.2 所示。

3.4.2 实验环境

所有实验均在表 3.3 所示的环境下进行。

3.4.3 评估指标

本文在以下实验场景中对所提出的方法进行评估：在点击率（CTR）预测中，

将训练好的模型应用于测试集中的每个交互来进行预测。本文采用了两个广泛使用的指标 AUC 和 F1 来评估 CTR 预测的性能。

表 3.3 实验环境

实验环境	参数
操作系统	Windows11 专业版
CPU	Intel(R) Core(TM) i7-13620H
GPU	NVIDIA GeForce RTX 3090
内存	16GB
算法语言	Python 3.9
学习框架	PyTorch 2.2

ROC (receiver operating characteristic) 曲线的横轴是“假正例率”(False Positive Rate, FPR)，纵轴是“真正例率”(True Positive Rate, TPR)。AUC (Area Under Curve) 被定义为 ROC 曲线下与坐标轴围成的面积，这个面积范围在 0~1 之间，如图 3.2 所示。一般情况下，ROC 曲线处于 $y=x$ 这条对称轴的上方，因此 AUC 的值范围在 0.5~1 之间。AUC 的值越高，说明模型性能越优秀，分类更接近于现实。

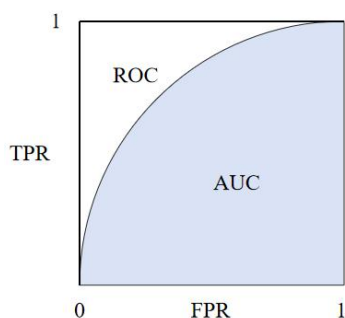


图 3.2 AUC 示意图

召回率 (Recall) 指的是模型推荐给用户真正喜欢的物品与用户实际喜爱物品总数的比例，也被称为查全率。Recall 的取值范围在 0~1 之间，Recall 的值越大表示模型预测的越准确，Recall 的公式 (4-26) 如下所示：

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3-18)$$

其中，TP (真正) 表示推荐的物品中用户真正喜欢的物品数，FN (假负) 表示用户实际上喜欢但是模型并未推荐给用户的物品数，Recall 的值是样本中的正例

预测对的比例。

准确率（Precision）表示为在推荐的列表中，用户真正喜欢的物品数与被推荐的物品总数的比例，也被称为查准率。Precision 的取值范围在 0~1 之间，Precision 的值越大表示模型预测的越准确，Precision 公式如（3-19）所示：

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3-19)$$

其中，TP（真正）表示推荐的物品中用户真正喜欢的物品数，FP（误报）表示被推荐的物品中用户不喜欢的物品数。

F1-score 评价指标综合考虑了 Precision 和 Recall 的结果而计算的衡量分数，为二者的加权调和均值，公式如（3-20）所示：

$$F_1 = \frac{2PR}{P + R} \quad (3-20)$$

3.4.4 实验模型对比

为了验证 MVCK 模型的性能，本文选取了 11 个模型进行对比分析，这些模型包括基于协同过滤算法的推荐模型（BPRMF）、基于知识图谱的推荐模型（PER,CKE,RippleNet,CurvRec）、基于 GNN 的推荐模型（KGCN,KGNN-LS,KGAT,KGIN）以及运用了对比学习方法的知识感知推荐模型（KGIC,KGCL）。

- **BPRMF**^[54]：这是一种典型的基于协同过滤的方法，它使用两两矩阵分解来进行由 BPR 损失优化的隐式反馈。
- **PER**^[56]：这是一种典型的基于路径的方法，它提取基于元路径的特征来表示用户和项目之间的连接。
- **CKE**^[57]：这是一种基于嵌入的方法，它将结构知识、文本知识和视觉知识结合在一个框架中。
- **RippleNet**^[35]：这是一种经典的基于嵌入的方法，可以在 KG 上传播用户的偏好。
- **KGCN**^[36]：基于 GNN 的模型，通过迭代集成邻近信息来丰富项嵌入。
- **KGNN-LS**^[58]：这是一个基于 GNN 的模型，利用 GNN 和标签平滑正则化方法丰富了项目嵌入。
- **KGAT**^[37]：这是一种基于 GNN 的方法，它迭代地集成了用户-项目-实体

图上的邻居与一个注意机制，以获得用户/项目表示。

- **KGIN**^[59]: 这是一种最先进的基于 GNN 的方法，它在用户意图的粒度上分离用户-项目交互，并对提出的用户-意图-实体图执行 GNN。
- **KGIC**^[60]: 这是一种基于知识感知推荐模型，该模型运用了对比学习方法，探索了交互式对比学习机制。
- **KGCL**^[70]: 是一种面向知识图谱感知推荐的新型对比学习框架。它旨在通过所提出的知识图谱增强模式来解决实体稀疏性的限制。
- **CurvRec**^[72]: 基于知识图谱推荐模型，通过融合混合曲率流形和图卷积网络，有效地利用知识图谱的几何结构来提升推荐性能。

以上所有模型的代码实现和参数选择都来自于论文中给出的代码链接或者 GitHub 上面其他作者的实现，并仔细调整了关键参数。本章将模型的嵌入大小固定为 64，嵌入参数使用 Xavier 方法进行初始化。本章使用 Adam 优化器来优化算法，同时将批量大小设置为 4096。利用网格搜索确认最佳参数设置，调整学习率 η_1 在 $\{0.001, 0.003, 0.01, 0.03\}$ 之间，以及 L2 正则化项 η_2 在 $\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$ 之间。表 3.4 中提供了其他超参数设置。

表 3.4 模型 MVCK 的参数设置

Dataset	α	K	L	ω	τ
MovieLens-1M	0.1	2	4	0.9	0.6
Book-Crossing	0.1	2	4	0.9	0.7
Last.FM	0.1	2	4	0.8	0.6

其中， α : 对比损失权重， K : 项目-实体视图 Light-GCN 层数， L : 用户-物品 Light-GCN 层数， ω : 对比学习中原视图权重， τ : 对比损失函数中 softmax 的温度系数。

3.5 实验结果以及分析

3.5.1 实验结果综合对比

在本节中，将 MVCK 模型与上述对比实验模型在三个基准数据集上面进行性能对比。为了保证实验结果的可靠性，所有的模型都取其最优结果。实验数据对比结果如表 3.5 所示。

表 3.5 各模型的实验数据对比

Model	MovieLens-1M		Book-Crossing		Last.FM	
	AUC	F1	AUC	F1	AUC	F1
BPRMF	0.8923	0.7894	0.6586	0.6117	0.7562	0.7014
PER	.0.7126	0.6670	0.6052	0.5732	0.6408	0.6042
CKE	0.9072	0.8026	0.6752	0.6098	0.7423	0.6715
RippleNet	0.9195	0.8420	0.7209	0.6472	0.7765	0.7021
KGCN	0.9095	0.8375	0.6842	0.6321	0.8025	0.7095
KGNN-LS	0.9152	0.8413	0.6765	0.6269	0.8053	0.7219
KGAT	0.9146	0.8433	0.7318	0.6524	0.8148	0.7426
KGIN	0.9162	0.8435	0.7264	0.6608	0.8292	0.7508
KGIC	0.9252	0.8559	0.7589	0.6712	0.8592	0.7653
KGCL	0.9287	0.8578	0.7590	0.6678	0.8352	0.7596
CurvRec	0.9349	0.8619	0.7607	0.6726	0.8530	0.7658
MVCK	0.9360	0.8632	0.7629	0.6781	0.8754	0.7985

本章通过将对比模型以及实验模型在上述的三个数据集上面进行比较，来测试本章实验模型的性能。

(1) 将 CKE 和 BPRMF 相比较可知，引入知识图谱作为辅助信息的重要性。通过将知识图谱中的辅助信息嵌入合并到 MF 中之后，不管在稀疏的 MovieLens-1M 数据集以及 Book-Crossing 数据集上，还是在并不稀疏的数据集 Last.FM 上，CKE 的性能均优于 BPRMF，由此表明引入知识图谱作为辅助信息能有效的提升推荐性能。通过将 KGCN、KGAT 与 CKE 比较可知，前者不管在 AUC 还是 F1 上都有很好的表现，证明了 GNN 具有很强的图学习能力。大多数基于 GNN 的方法比嵌入目标方法和基于路径的方法表现得更好，这表明了引入 GNN 汇聚项目高阶信息的有效性。

(2) 将 KGIC、KGCL 与 KGCN、KGAT 相比可知，结合了对比学习方法的知识感知模型性能更好，相比于单独的推荐任务，加上对比任务的多任务推荐模式更加先进。对比学习自监督的学习方法能够很好的缓解数据稀疏性问题。

(3) 由表 3.5 可知，本章提出的 MVCK 模型在三个基准数据集上面均有良

好的表现。同时, MVCK 模型与最近的同样利用对比学习的模型 KGIC 对比有不错的表现。在较稀疏的数据集 Last.FM 上 AUC 指标有 1.62%的提升、F1 指标有 3.32%的提升。在较稠密的数据集 MovieLens-1M 上 AUC 指标依然有着 0.11%的提升、F1 指标有 0.13%的提升。相比于最新的知识感知推荐模型 CurvRec, 在三个数据集上面同样实现了优秀的推荐性能, 证明了 MVCK 模型的先进性。

3.5.2 消融实验

为了验证模型的每个模块对最终结果的影响, 本节进行了消融实验。比较 MVCK 模型的几个变体, 并分析了这些模块的影响。MVCK_{w/o}^G表示去掉项目-实体模块, 来评估知识图谱对模型的影响。MVCK_{w/o}^L表示去掉对比学习模块, 来评估对比学习对模型的影响。实验结果如表 3.6 所示。由表 3.6 的消融实验结果可以看出。项目-实体模块以及对比学习模块对实验结果都有影响。其中项目-实体模块消融之后, 模型的性能大幅减低, 充分说明知识图谱的信息在学习中的重要性。其次, 消融对比学习模块之后, 模型性能也随之下降, 能够说明对比学习模块能够充分挖掘数据的潜在信息, 联合对比学习的多任务学习比单独的推荐任务更好。

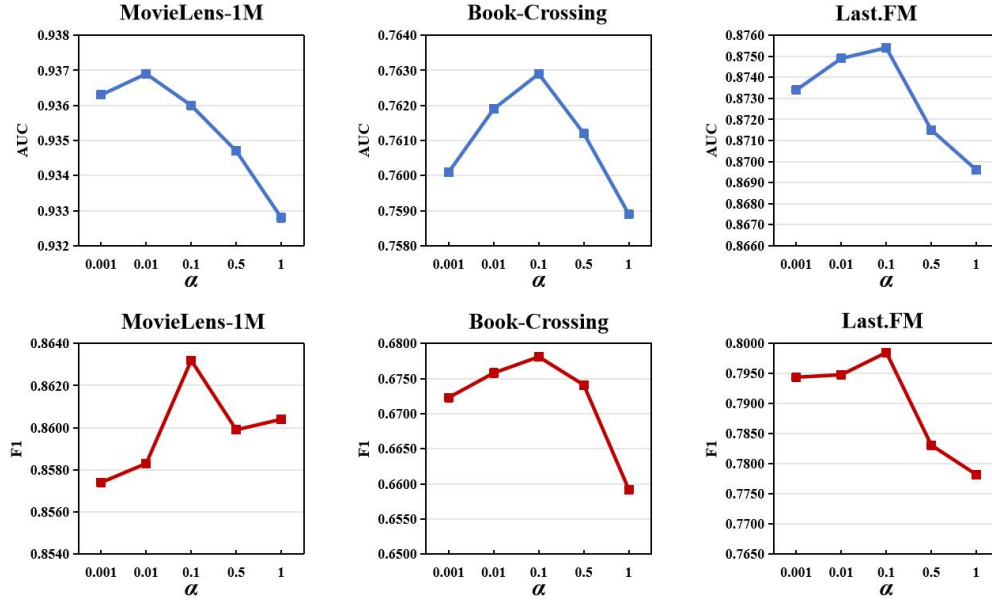
表 3.6 MVCK 及其消融变体的影响

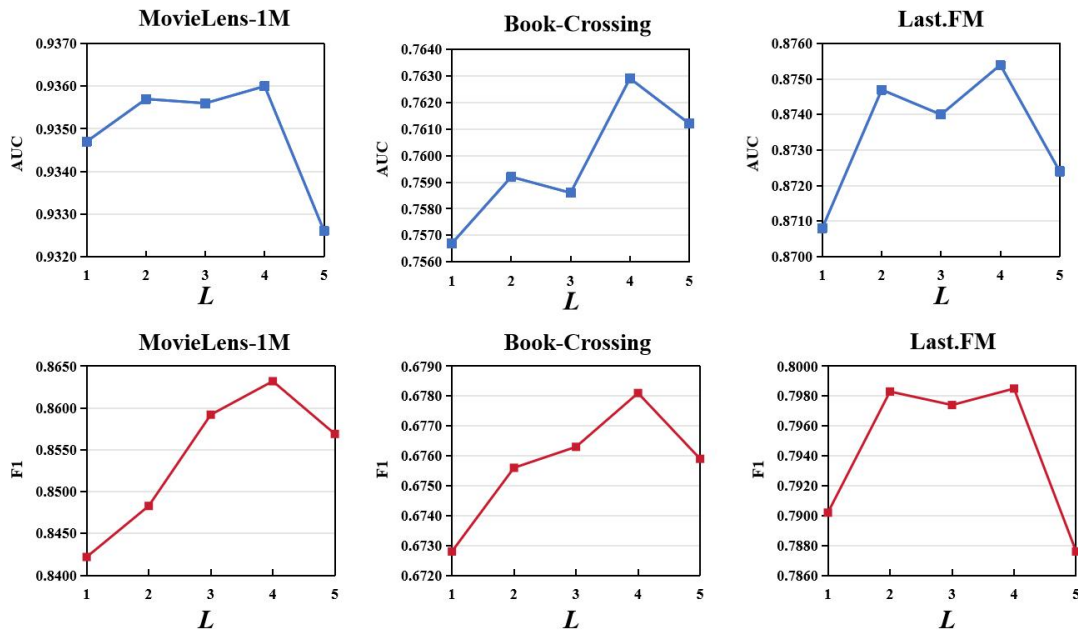
Model	MovieLens-1M		Book-Crossing		Last.FM	
	AUC	F1	AUC	F1	AUC	F1
MVCK _{w/o} ^G	0.9290	0.8011	0.7591	0.6709	0.8590	0.7711
MVCK _{w/o} ^L	0.9315	0.8376	0.7602	0.6723	0.8607	0.7825
MVCK	0.9360	0.8632	0.7629	0.6781	0.8754	0.7985

3.5.3 超参敏感性分析

(1) 对比学习权重 α 的敏感性分析

本小节分析了对比权重 α 对于模型性能的影响程度, 具体如图 3.3 所示。本文让对比权重 α 在{0.001,0.01,0.1,0.5,1}之间取值, 根据结果可以得知在三个公开的数据集 MovieLens-1M、Book-Crossing 以及 Last.FM 上当 $\alpha=0.1$ 时模型性能最佳。 α 表示的是多任务学习中的对比学习损失的权重, 当 $\alpha=0.1$ 时模型性能最佳, 随着 α 的值增加模型性能下降迅速, 可以得知, 在多任务学习中, 应该以推荐任务为主。


 图 3.3 对比学习权重 α 对模型性能的影响

 (2) 用户-项目视图 Light-GCN 层数 L 的敏感性分析

 图 3.4 用户-项目视图 Light-GCN 层数 L 对模型性能的影响

MVCK 模型有两种不同的数据结构, 针对用户-项目视图中 Light-GCN 层数 L 对于模型性能的影响, 本节调整了 Light-GCN 层数 L 的在 $\{1, 2, 3, 4, 5\}$ 之间取值。图 3.4 展示了在三个公开的基准数据集下不同层数 L 的模型的性能表现。从图中可以看出, MVCK 模型对 Light-GCN 层数 L 很敏感, 当层数为 4 时, 模型性能总能达到最优。但是在数据稠密的数据集 MovieLens-1M 和 Book-Crossing 上, 随着层数的增加, 性能持续上升, 造成原因主要是因为更深的网络能够充分学习到

稠密的数据集中丰富的信息,而在稀疏的数据集 Last.FM 上,因为数据信息较少,只需要 2 层网络就充分学习到用户以及项目特征表示。但是当层数超过 4 时,模型的性能都开始显著的下降,因为较大的深度会给模型带来大量的噪声。

(3) 项目-实体视图 Light-GCN 层数 K 的敏感性分析

MVCK 模型纳入了知识图谱作为辅助信息来丰富项目的表示学习。在大多数基于知识谱图模型中证明了利用好知识图谱中丰富的节点信息以及关系信息能够提升推荐的准确性。本节对项目-实体视图的 Light-GCN 层数 K 对于模型性能的影响进行了分析,Light-GCN 层数 K 范围在 $\{1,2,3,4\}$ 之间取值。实验结果如图 3.5 所示。从图中可以知道对于知识图谱中的信息只需要 2 次消息聚合,模型性能达到最佳。随着网络层数增加,不管是稀疏数据集还是稠密的数据集,模型性能随之下下降,因为知识图谱中含有大量的噪音,消息聚合的次数越多,噪音干扰的影响越大,模型的性能越低,因此只需要 2 次消息聚合是足够的。

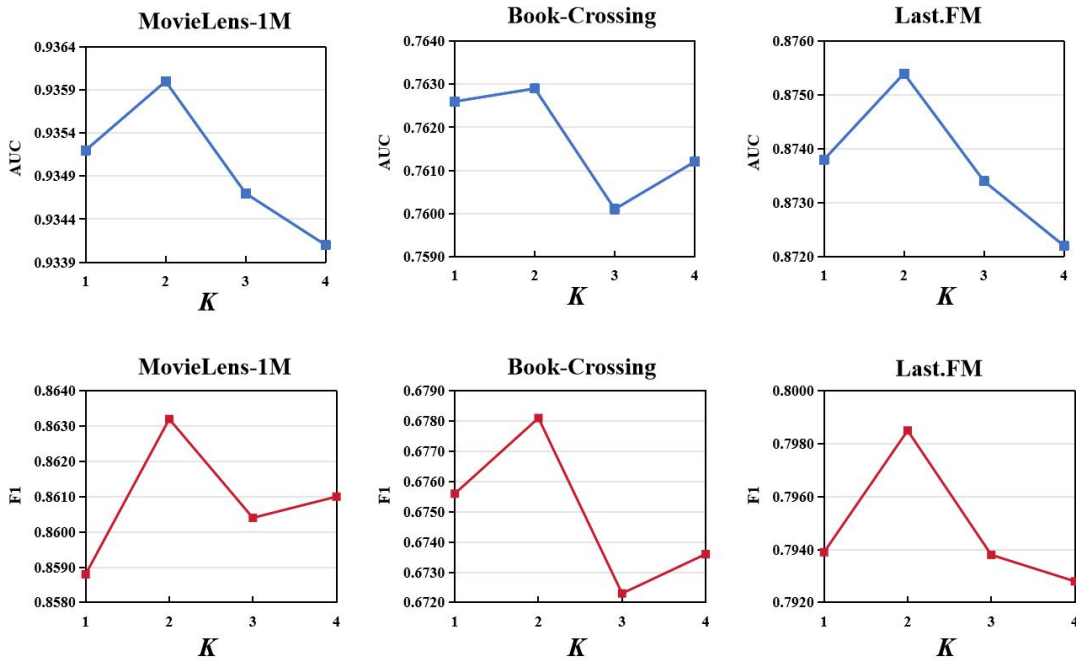
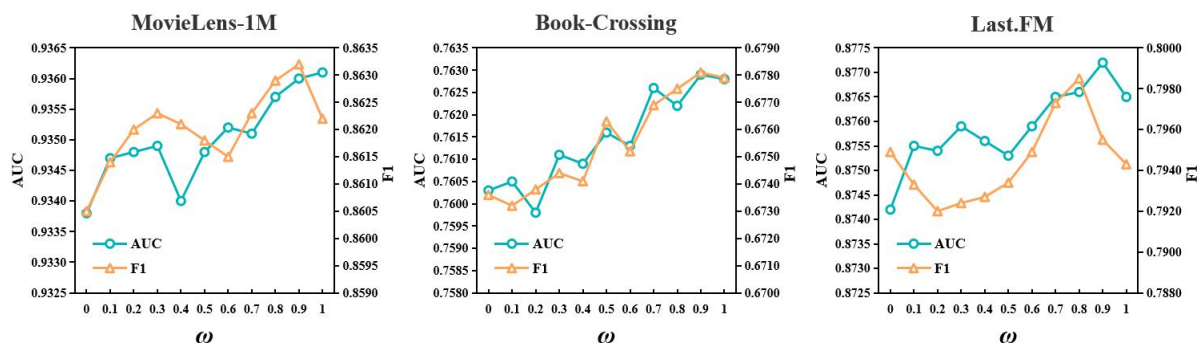


图 3.5 项目-实体视图 Light-GCN 层数 K 对模型性能的影响

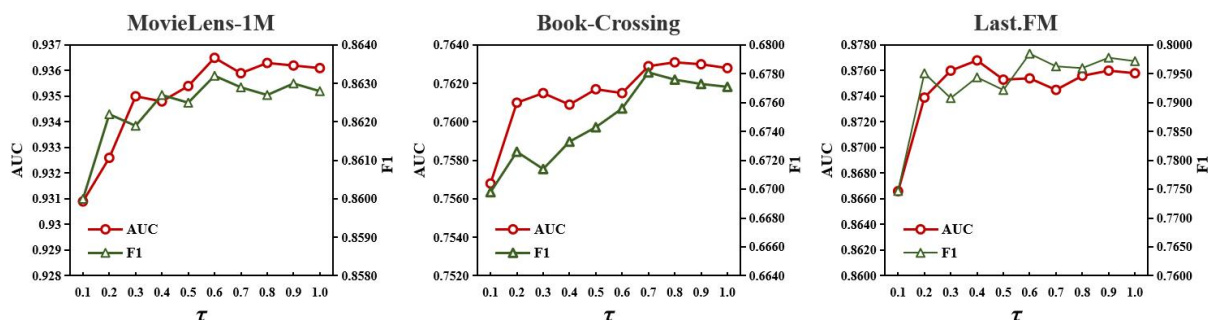
(4) 对比学习原视图权重 ω 的敏感性分析

本章提出的多负例对比学习利用了不同视图中的负例,从不同角度学习来自不同视图中潜在的信息,对于来自不同视图的信息本文设置了不同的权重,其中来自原视图的权重为 ω ,本节对原视图权重 ω 对于模型性能的影响设置了参数实验,原视图权重 ω 的取值在 $\{0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0\}$ 中,实验结果如图 3.6 所示。


 图 3.6 对比学习原视图权重 ω 对模型性能的影响

图中结果表明, 在三个数据集上随着 ω 增大, 模型的性能大致呈增加的趋势。证明不同视图对于对比学习最终的结果均产生贡献, 在数据集 MovieLens-1M 和 Book-Crossing 上当 ω 接近于 1 时, 模型的性能达到最大, 因为这两个数据集中的交互数据更多, 不同视图中的信息均在对比学习中利用到, 丰富了用户项目最终的表示学习, 因此在对比学习中应该把不同视图放在同一水平上面。而在数据集 Last.FM 上, 当 ω 等于 0.8 时, 模型性能最佳, 随着 ω 继续增大, 模型性能反而下降, 因此在稀疏的数据集上面应该更注意对比视图上面的信息。

(5) 温度系数 τ 的敏感性分析


 图 3.7 温度系数 τ 对模型性能的影响

温度系数 τ 是用来控制对比学习中相似度计算结果分布形状的参数, 同时也控制对负样本的惩罚程度。当温度系数 τ 取值越小, 模型越关注与正样本相似的负样本, 并拉远负样本与正样本之间的距离。当 τ 取值较大时, 模型区分所有负样本的能力变差。

为了寻找合适的温度系数 τ , 本节对其设置了一系列参数实验, 度系数 τ 在 $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ 中取值, 实验结果如图 3.7 所示。从图中可以得知当温度系数 τ 取最小值 0.1, 模型的性能最差, 此时对于负样本惩罚过大,

导致样本之间的距离过于远离,无法学习到很好的向量表示,导致推荐的结果下降。但是随着温度系数 τ 取值增大,在数据集 Last.FM 上,相比另外两个数据集结果变化趋势更加平缓,此时对比学习任务在多任务学习中主要其辅助作用,模型的性能在 τ 的值取 0.6 时最佳。在数据集 MovieLens-1M 和 Book-Crossing 上,随着 τ 的值增加,模型性能变化趋势明显,其中在 MovieLens-1M 上,当 τ 取 0.6 综合性能最佳,在 Book-Crossing 上 τ 取 0.7 综合性能最佳。

3.6 本章小结

在本章中主要聚焦于对比学习在基于 GNN 的推荐模型中的应用,并提出了基于多视图对比学习的知识感知推荐模型(MVCK)。一方面,用户项目视图以及知识图谱视图都能作为图结构,利用不同的数据增强策略来获得这两个视图的子图,用来充分利用对比学习。其次,利用不同的 GNN 来充分学习用户和项目的嵌入向量。最后,设计多负例对比学习方法来探索新的对比学习方法。

本章提出的基于多视图对比学习的知识感知推荐模型(MVCK)包括以下四个模块:子图构建模块、消息聚合模块、对比学习模块以及评分预测模块。首先利用不同的数据增强策略构建子图,接着不同子图在对应的 GNN 模块中进行消息聚合获得最终的嵌入向量,然后进入对比学习模块挖掘更深层次的信息。最后对模型进行评分预测。MVCK 模型在三个基准数据集上面进行了大量的实验,实验结果表明了 MVCK 模型的有效性。

第4章 多视图噪声增强图对比学习的知识感知推荐模型

第三章中 MVCK 模型探索了在基于 GNN 的知识感知推荐模型中运用了对比学习方法，有效的提高了推荐模型的性能。在本章中将保留 MVCK 模型中的部分模块，同时针对多层神经网络聚合后导致数据分布不均问题提出新的推荐模型多视图噪声增强图对比学习的知识感知推荐模型（MNCL）。该方法通过在图卷积神经网络中引入噪声增强，优化每层神经网络聚合后数据的均匀分布，缓解长尾问题的同时也有益于对比学习的性能。最终，MNCL 模型在 MovieLens-1M、Last.FM 和 Book-Crossing 三个数据集上面验证了模型的有效性。

4.1 研究动机

在推荐领域，数据稀疏一直是亟待解决的问题，对比学习作为自监督学习方法，具有从未标记的样本中学习用户项目特征表示的潜力，因此在推荐领域中受到了广泛关注。KGCL^[70]模型利用对比学习方法，从知识图谱中提取对用户偏好预测有用的信号,从而减少噪音信息对推荐系统的影响。MCCLK^[68]模型以分层的方式进行对比学习，考虑了用户-项目-实体图的结构信息。KACL^[69]给知识图中每个关系加上一个可学习的嵌入向量，并利用对比学习缓解训练中用户-项目视图占主导的问题。这些推荐模型利用对比学习解决了知识图谱带了的噪音影响以及数据稀疏等问题。然而，本文认为对比学习方法在基于 GNN 的推荐模型应用仍然存在以下两个问题：由于数据的稀疏性，在 GNN 信息聚合过程中，所有物品的学习表示均会像热门物品靠拢，会导致生成的节点特征表示无法区分的退化问题^[59]，不利于长尾项目。其次，对比学习方法的学习性能受到样本质量的影响，样本在嵌入空间的均匀化分布也会影响对比学习模型性能。XSimGCL^[71]提出一种新颖的图对比学习方法，文中提到了嵌入空间样本的均匀化是让对比学习方法更有益的影响。因此，考虑到上述问题，本文提出了一个基于图对比学习的多视图噪声增强知识感知推荐模型，称为 MNCL。首先，通过用户-项目、用户-项目-实体、项目-实体三个不同视图来学习用户和项目多个视角下的特征表示，以此学习到更加全面的知识信息。此外，MNCL 模型进一步采用了多视图以及多负例对比学习方法，通过多视图生成高质量的正负样本，然后采用多负例对比学习方法充分的学习用户项目的嵌入向量来提高推荐系统的准确性。更重要的是，为了缓解退化问题对模型的影响，模型在图卷积神经多层聚合过程中引入一定程度的噪音，通过给节点注入不确定性的噪声因素来优化节点在嵌入空间特征表示下

的均匀化分布。最后，在多个公开数据集上的实验结果也证明了所提模型的有效性。本章的主要贡献如下：

(1) 本文提出了一种新的推荐模型 MNCL，利用多视图以及噪音增强为对比学习生成优质的正负样本，最后通过多负例对比学习方法来充分学习不同视图之间的信息。

(2) 考虑到信息聚合过程中样本均匀化分布的重要性，我们通过在信息聚合过程中引入噪音增强的方法来优化样本的均匀性分布。

(3) 在 MovieLens-1M、Last.FM 和 Book-Crossing 上进行了广泛的实验。结果表明模型 MNCL 在更好的表示学习方面的优势，表明了优化对比学习方法的有效性。

4.2 问题定义

与第三章相似，输入的数据包括两种关键的结构化数据，即用户-物品交互数据和知识图谱三元组数据。最后说明了模型的推荐任务。

4.2.1 用户-项目交互数据

在大多数推荐场景中，用户集合表示为 $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$ ，其中 M 表示用户的个数；项目的集合表示为 $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ ， N 为项目的数量。将 M 个用户与 N 个项目构建为用户-项目的交互矩阵 $\mathbf{Y} \in \mathbb{R}^{K \times M}$ 。在矩阵 \mathbf{Y} 中每一个元素都表示用户对项目的隐式反馈，其中， $\mathbf{Y} = \{y_{uv} | u \in \mathcal{U}, v \in \mathcal{V}\}$ 。 y_{uv} 的取值为 0 或 1，当 $y_{uv} = 1$ 表示用户与项目有互动。

4.2.2 知识图谱

在知识图谱方面，除了考虑历史交互外，现实世界中与物品相关的事实（例如物品属性或外部常识知识）以异构图的形式存储在知识图中。设 $\mathcal{G} = \{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{R}\}$ 为知识图谱，其中 h, r, t 分别代表知识三元组的头、关系和尾； \mathcal{E} 和 \mathcal{R} 分别是 \mathcal{G} 中的实体和关系集。例如，三元组（《惊天魔盗团》，导演，易斯·卡卡杜瑞尔）表示易斯·卡卡杜瑞尔是电影《惊天魔盗团》的导演。在许多推荐场景中，物品 $v \in \mathcal{V}$ 对应于知识图谱中的实体 $e \in \mathcal{E}$ 。例如，在电影推荐中，“蜘蛛侠”这个物品也在知识图谱中以相同的名称作为一个实体出现。因此，本章建立了一个物品-实体对齐的集合 $\mathcal{A} = \{(v, e) | v \in \mathcal{V}, e \in \mathcal{E}\}$ ，其中 (v, e) 表示

物品 v 可以与知识图中的实体 e 对齐。通过物品和知识图实体之间的对齐，知识图能够为交互数据提供补充信息并对物品进行描述。

4.2.3 推荐任务

同第三章一样，输入为用户-物品交互矩阵 \mathbf{Y} 和知识图 \mathbf{G} ，最后的推荐任务是函数 $\hat{y}_{uv} = f(u, v)$ ，该函数的目标是预测用户 u 对于项目 v 的感兴趣程度。

4.3 算法框架

本节将介绍提出的多视图噪声增强图对比学习的知识感知推荐模型（MNCL）。图 4.1 显示了 MNCL 模型的流程图。主要组成部分由三个模块构成，包括多视图的构建、噪音增强的消息聚合以及多负例对比学习模块。首先构建包括用户-项目交互图、项目-实体视图以及用户-项目-实体的多视图。然后采用不同的数据增强方式，其中用户-项目视图通过在 Light-GCN^[29]中进行噪音增强的方式优化数据均匀化分布，用户-项目-实体的视图采用随机丢边、丢点的方式进行数据增强。接着通过不同的图编码器学习用户、项目的节点表示。其中用户-项目视图模块采用带噪音增强的 Light-GCN 来学习更加均匀化分布的用户项目节点表示，用户-项目-实体视图采用路径感知的 GNN 学习更多结构化信息的项目表示，项目-实体视图采用关系感知 GNN 利用知识图谱中关系的信息挖掘更丰富的项目表示。接下来对这三部分进行详细的介绍。

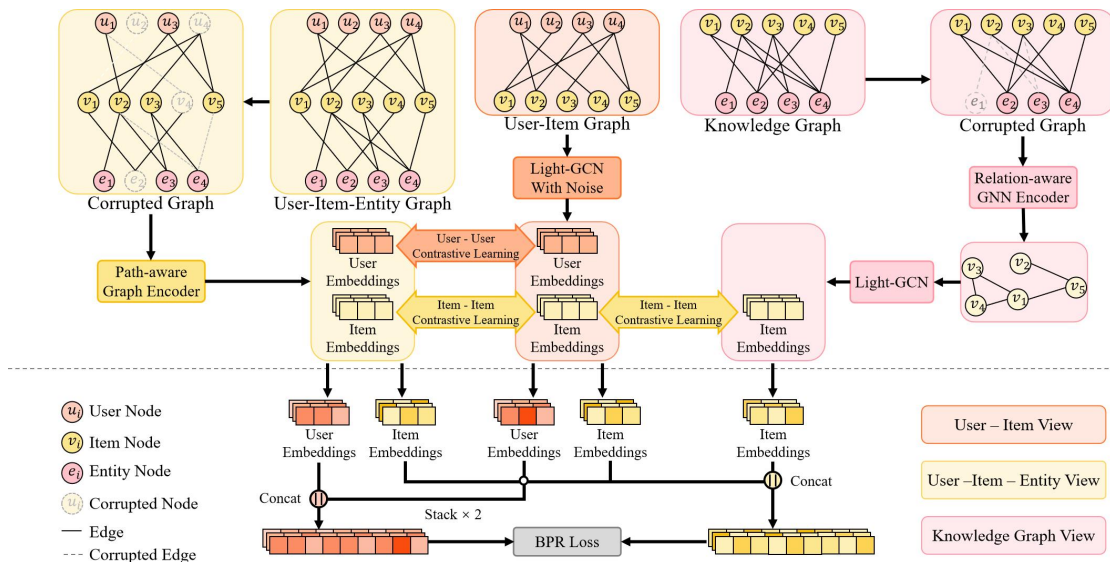


图 4.1 MNCL 模型图

4.3.1 子图构建

由于真实场景中能够观察到的数据间的交互非常稀疏,挖掘数据的隐含信息非常的困难,因此需要采用数据增强的方式来处理样本。本章采用节点丢失、边丢失两种数据增强的方式对用户-项目视图 \mathcal{S} 、项目-实体视图 \mathcal{G} 以及用户-项目-实体视图 \mathcal{M} 进行处理。分别得到对用的子图,以户-项目-实体视图 \mathcal{M} 的子图 m 为例,具体方法如下:

$$m = (\mathbf{M}' \odot \mathcal{V}, \mathbf{M}_1 \odot \mathcal{E}, p, \rho) \quad (4-1)$$

其中, $\mathbf{M}_1 \in \{0,1\}^{|\mathcal{E}|}$ 是边缘集合 \mathcal{E} 上的掩码向量, $\mathbf{M}' \in \{0,1\}^{|\mathcal{V}|}$ 是应用在节点集 \mathcal{V} 上的掩码向量。 p, ρ 分别是子图 \mathcal{S} 上的节点丢失率以及边丢失率。

4.3.2 关系感知 GNN

为了学习知识图谱中项目与实体丰富交互信息,本章沿用了 MVCK 模型中的关系感知 GNN。具体如下,首先从知识图谱 \mathcal{G} 中递归地学习 K 次的项目表示,关系感知 GNN 聚合机制公式为:

$$\mathbf{e}_i^{(k+1)} = \frac{1}{|\mathcal{N}_i|} \sum_{(r,j) \in \mathcal{N}_i} \mathbf{e}_r \odot \mathbf{e}_j^{(k)} \quad (4-2)$$

$$\mathbf{e}_j^{(k+1)} = \frac{1}{|\mathcal{N}_j|} \left(\sum_{(r,i) \in \mathcal{N}_j} \mathbf{e}_r \odot \mathbf{e}_i^{(k)} + \sum_{(r,i) \in \mathcal{N}_j} \mathbf{e}_r \odot \mathbf{e}_i^{(k)} \right) \quad (4-3)$$

其中 $\mathbf{e}_i^{(k)}$ 和 $\mathbf{e}_j^{(k)}$ (对于 $\forall k \in K$) 分别表示物品 i 和实体 j 的表示,它们存储了从它们的 $(k-1)$ 跳邻居传播的关系信号。知识图谱中的邻居实体和关系都被编码到物品表示中。最后同第三章中一样得到正则化的稀疏邻接矩阵 $\tilde{\mathbf{G}}$ 。

项目-实体知识图谱视图经过关系感知 GNN 聚合消息后,采用 Light-GCN 进行 K 次聚合,来学习更好的项目表示。在每层 Light-GCN 中的第 k 消息传递和聚合过程可以表示为:

$$\mathbf{e}_i^{(k+1)} = \sum_{j \in \mathcal{N}(i)} \tilde{\mathbf{G}} \mathbf{e}_j^{(k)} \quad (4-4)$$

其中, $\mathcal{N}(i)$ 是邻居物品, $\tilde{\mathbf{G}}$ 是公式中的规范化稀疏图邻接矩阵, $\mathbf{e}_i^{(k)}$ 是物品 i 每层 Light-GCN 的第 k 次消息聚合后的表示。然后,将 K 次聚合操作后的物品表示相加,得到最终的特征向量 \mathbf{e}_i :

$$\mathbf{e}_i = \mathbf{e}_i^{(0)} + \dots + \mathbf{e}_i^{(K)} \quad (4-5)$$

为了来学习更好的项目表示。同样采用 Light-GCN 图编码器进行学习。在每层 Light-GCN 中的第 k 消息传递和聚合过程可以表示为：

$$\mathbf{e}_i^{(k+1)} = \sum_{j \in \mathcal{N}(i)} \tilde{\mathbf{S}} \mathbf{e}_i^{(k)} \quad (4-6)$$

其中， $\mathcal{N}(i)$ 是邻居物品， $\tilde{\mathbf{S}}$ 是公式中（4-5）的规范化稀疏图邻接矩阵， $\mathbf{e}_i^{(k)}$ 是物品 i 每层 Light-GCN 的第 k 次消息聚合后的表示。然后，将 K 次聚合操作后的物品表示相加，得到最终的项目-实体视图的项目特征向量 \mathbf{e}_i^g ：

$$\mathbf{e}_i^g = \mathbf{e}_i^{(0)} + \dots + \mathbf{e}_i^{(K)} \quad (4-7)$$

知识图谱中存在过多的噪音干扰，如果添加噪声增强反而会引入更多的干扰信息，反而会降低模型的性能，同时这也能反应在用户-项目视图中增加噪声增强能够影响消息聚合的最终结果，适当的噪声增强能够优化样本的均匀化分布。

4.3.3 噪声增强的 Light-GCN

已知用户和相互的交互矩阵为 $\mathbf{R} \in \mathbb{R}^{M \times N}$ ，其中 M 为用户数量， N 为项目数量。如果用户与项目进行交互则 $R_{ui}=1$ ，否则等于 0。然后可以得到用户项目的邻接矩阵：

$$\mathbf{A} = \begin{pmatrix} 0 & \mathbf{R} \\ \mathbf{R}^T & 0 \end{pmatrix} \quad (4-8)$$

本章中同样主要采用了 Light-GCN。Light-GCN 包含简单的消息传递和聚合机制，没有特征变换和非线性激活，非常的简单有效并且减少计算成本。它递归地执行 K 次聚合，在每层网络的第 k 次聚合过程可以被表述如下：

$$\mathbf{e}_u^{(k+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \mathbf{e}_i^{(k)} \quad (4-9)$$

$$\mathbf{e}_i^{(k+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \mathbf{e}_u^{(k)} \quad (4-10)$$

其中， $\mathbf{e}_u^{(k)}$ 和 $\mathbf{e}_i^{(k)}$ 表示用户 u 和物品 i 在第 k 层的嵌入， \mathcal{N}_u ， \mathcal{N}_i 表示用户 u 和物品 i 的邻居，然后将不同层的表示相加，得到最终的特征向量 \mathbf{e}_u 和 \mathbf{e}_i ，如下所示：

$$\mathbf{e}_u = \mathbf{e}_u^{(0)} + \dots + \mathbf{e}_u^{(K)}, \mathbf{e}_i = \mathbf{e}_i^{(0)} + \dots + \mathbf{e}_i^{(K)} \quad (4-11)$$

为了缓解多层聚合导致的退化问题，增加样本的多样性，在每层 Light-GCN

消息聚合中加入噪声增强 φ ，其中第 l 层的 φ 表达式为：

$$\varphi = \text{sign}(\mathbf{e}^{(l)}) \odot \mathbf{Y} \odot \beta \quad (4-12)$$

其中 $\mathbf{e}^{(l)}$ 表示第 l 层的输入向量矩阵。 \mathbf{Y} 为与 $\mathbf{e}^{(l)}$ 具有一样维度的正则化随机矩阵， β 为噪音增强系数。加入噪声增强 φ 后，在第 l 层，Light-GCN 聚合过程可以被表述如下：

$$\mathbf{e}^{(l+1)} = \tilde{\mathbf{A}}(\mathbf{e}^{(l)} + \varphi) \quad (4-13)$$

其中 $\tilde{\mathbf{A}}$ 是规范化稀疏图邻接矩阵， $\tilde{\mathbf{A}} = (\mathbf{D})^{-\frac{1}{2}} \mathbf{A} (\mathbf{D})^{-\frac{1}{2}}$ 。最终将经过 L 层 Light-GCN 聚合后的矩阵表示相加并且取平均之后得到最终的向量表示为：

$$\mathbf{e} = \frac{\mathbf{e}^{(0)} + \mathbf{e}^{(1)} + \dots + \mathbf{e}^{(L)}}{L} \quad (4-14)$$

$$\mathbf{e} = \frac{\mathbf{e}^{(0)} + (\tilde{\mathbf{A}}\mathbf{e}^{(0)} + \varphi) + \dots + (\tilde{\mathbf{A}}^L\mathbf{e}^{(0)} + \tilde{\mathbf{A}}^L\varphi + \dots + \varphi)}{L} \quad (4-15)$$

得到的最终特征向量 \mathbf{e} 中包含用户-项目视图的用户最终特征向量 \mathbf{e}_u^s 以及项目的最终特征向量 \mathbf{e}_i^s 。

4.3.3 路径感知 GNN

虽然采用关系感知 GNN 利用了知识图谱中的信息，但是大量的研究表明知识图谱中的信息在训练中利用率低，因此本文采用一种路径感知的 GNN^[62]，该方法学习聚合 h 次的相邻信息，同时保留路径信息，即用户-项目-实体连接信息。第 h 层聚集过程可以表示为：

$$\mathbf{e}_u^{(h+1)} = \frac{1}{|\mathcal{N}_u|} \sum_{i \in \mathcal{N}_u} \mathbf{e}_i^{(h)} \quad (4-16)$$

$$\mathbf{e}_i^{(h+1)} = \frac{1}{|\mathcal{N}_i|} \sum_{(r,u) \in \mathcal{N}_i} \delta(i, r, u) \mathbf{e}_r \odot \mathbf{e}_u^{(h)} \quad (4-17)$$

其中， $\mathbf{e}_u^{(h)}$ 和 $\mathbf{e}_i^{(h)}$ 分别表示用户 u 和项目 i 的表示，它们储存了 $(h-1)$ 次聚合邻居传播的关系信号，从而存储多跳路径的整体语义。为了对每个关系和实体进行加权，注意力权重 $\delta(i, r, u)$ 计算如下：

$$\begin{aligned} \delta(i, r, u) &= \text{softmax}((\mathbf{e}_i \parallel \mathbf{e}_r)^T \cdot (\mathbf{e}_u \parallel \mathbf{e}_r)) \\ &= \frac{\exp((\mathbf{e}_i \parallel \mathbf{e}_r)^T \cdot (\mathbf{e}_u \parallel \mathbf{e}_r))}{\sum_{(u', r) \in \widehat{\mathcal{N}}(i)} \exp((\mathbf{e}_i \parallel \mathbf{e}_r)^T \cdot (\mathbf{e}_{u'} \parallel \mathbf{e}_r))} \end{aligned} \quad (4-18)$$

其中, \parallel 表示连接操作, $\widehat{\mathcal{N}}(i)$ 表示相邻实体 $\mathcal{N}(i)$ 和物品 i 本身的集合。之后将所有层的表示相加, 得到全局的用户以及项目表示 \mathbf{e}_u^m 和 \mathbf{e}_i^m :

$$\mathbf{e}_u^m = \mathbf{e}_u^{(0)} + \dots + \mathbf{e}_u^{(H)} \quad \mathbf{e}_i^m = \mathbf{e}_i^{(0)} + \dots + \mathbf{e}_i^{(H)} \quad (4-19)$$

4.3.5 多负例对比学习

以上得到了用户-项目视图以及项目-实体视图的嵌入向量。用户-项目视图的用户最终特征向量为 \mathbf{e}_u^s 、项目的最终特征向量为 \mathbf{e}_i^s 以及项目-实体视图中的项目的最终特征向量为 \mathbf{e}_i^g , 用户-项目-实体视图中用户最终特征向量为 \mathbf{e}_u^m 、项目的最终特征向量为 \mathbf{e}_i^m 。将它们映射到计算对比损失的空间中。然后定义正样本和负样本, 对于视图 S 中的任意用户节点 \mathbf{u}_i^s , 其对比的正样本是另一个视图 G 学习到的相同节点嵌入 \mathbf{u}_i^g ; 而在两个视图中, 除了 \mathbf{u}_i^s 之外的其他节点嵌入 \mathbf{u}_j^s 以及 \mathbf{u}_j^g 被视为负样本, 同时本文认为在不同视图中负样本的权重不一样, 定义原视图中的权重为 ω 。定义好正样本和负样本后, 本章将对比损失函数分为全局对比损失 \mathcal{L}^G 以及包含知识图对比 \mathcal{L}^{KG} , 以项目的损失为例, 具体公式如下:

$$\mathcal{L}_i^G = -\log \frac{\exp(s(\mathbf{e}_i^s, \mathbf{e}_i^m)/\tau)}{\exp(s(\mathbf{e}_i^s, \mathbf{e}_i^m)/\tau) + \omega \sum_{i \neq j} \exp(s(\mathbf{e}_i^s, \mathbf{e}_j^s)/\tau) + \sum_{i \neq j} \exp(s(\mathbf{e}_i^s, \mathbf{e}_j^m)/\tau)} \quad (4-20)$$

$$\mathcal{L}_i^{KG} = -\log \frac{\exp(s(\mathbf{e}_i^s, \mathbf{e}_i^g)/\tau)}{\exp(s(\mathbf{e}_i^s, \mathbf{e}_i^g)/\tau) + \omega \sum_{i \neq j} \exp(s(\mathbf{e}_i^s, \mathbf{e}_j^s)/\tau) + \sum_{i \neq j} \exp(s(\mathbf{e}_i^s, \mathbf{e}_j^g)/\tau)} \quad (4-21)$$

其中, τ 表示温度系数。 $s(\cdot)$ 表示余弦相似度计算。 j 表示除了 i 以外的节点, 用户特征向量计算的对比损失 \mathcal{L}_u^G 与 \mathcal{L}_i^G 类似, 需要将公式中的 i 换成 u 即可。总体的对比损失为:

$$\mathcal{L}_{CL} = \mathcal{L}_i^G + \mathcal{L}_u^G + \mathcal{L}_i^{KG} \quad (4-22)$$

4.3.6 模型预测

用户-项目视图、项目-实体视图以及用户-项目-关系视图经过多层聚合以及对比学习优化后, 获得了用户多个表示 \mathbf{e}_u^s 和 \mathbf{e}_u^m 以及多个项目表示分别为 \mathbf{e}_i^s 、 \mathbf{e}_i^m 和 \mathbf{e}_i^g 。将这些优化后的特征向量进行串联操作, 得到最终的用户和项目的特征向量, 并通过预测函数获得最终的预测评分, 如下所示:

$$\mathbf{e}_u = \mathbf{e}_u^s \parallel \mathbf{e}_u^m, \quad \mathbf{e}_i = \mathbf{e}_i^s \parallel \mathbf{e}_i^m \parallel \mathbf{e}_i^g \quad (4-23)$$

$$\hat{y}(u, i) = \mathbf{e}_u^T \mathbf{e}_i \quad (4-24)$$

本章采用了多任务训练策略来优化整个模型。为了重构历史数据, 鼓励用户

历史项目的预测得分高于未观察到的项目，本章采用了 BPR 损失：

$$\mathcal{L}_{BPR} = \sum -\log \sigma(\hat{y}(u, i^+) - \hat{y}(u, i^-)) \quad (4-25)$$

其中， i^+ 表示训练中观察到的与用户 u 有交互的项目， i^- 表示训练中未观察到的项目。 σ 是 sigmoid 函数。通过将对比损失与 BPR 损失相结合，优化以下目标函数来学习模型参数：

$$\mathcal{L}_{MVCK} = \mathcal{L}_{BPR} + \alpha \mathcal{L}_{CL} + \lambda \|\Theta\|_2^2 \quad (4-26)$$

其中， Θ 是模型参数集， α 和 λ 是两个超参数，分别用于控制对比损失和 L2 正则化项。

4.3.7 算法流程

MNCL 推荐算法如表 4.1 所示。

4.4 实验设置

4.4.1 实验数据集

本章算法采用的数据集分别为 Movilens-1M 数据集、Book-Crossing 数据集和 Last.FM 数据集，详细信息如表 3.2 所示：

1. MovieLens-1M 数据集。在 MovieLens-1M 数据集中有 6036 名用户以及 2445 个项目。包含用户对项目的评分，范围在 1~5 之间。
2. Book-Crossing 数据集。Book-Crossing 数据集来自于 Book Crossing 社区 17860 个用户对大约 1.5 万册图书的真实评分，评分范围在 0~10 之间。
3. Last.FM 数据集。Last.FM 数据集是来自 Last.FM 社区中 1872 个听众对于 3846 首音乐的收听信息，其中包括用户对音乐的打分以及用户和音乐的属性信息。

由于 MovieLens-1M、Book-Crossing 和 Last.FM 中的交互是显式反馈，本章遵循 RippleNet 的方法，将它们转换为隐式反馈，标签为 1 时表示该用户对项目的反馈是正向的，反之标签为 0。其中，对于 MovieLens-1M，本文将评分阈值设为 4，即用户对项目的评分超过 4 则为样本标签设为 1。但由于 Last.FM 和 Book-Crossing 的稀疏性，它们没有设置阈值，产生交互记录（打分为行）的即为 1，否则为 0。至于负样本，对于每个用户，本章从他未观看的物品中随机抽

样，抽样数量与他的正样本相等。本章使用 FreeBase 构建知识图谱，大致过程如下：首先，根据数据集中项目的 ID 和 FreeBase 中的 ID 进行配对，随后筛选出知识库中与这些项目相关的子知识图谱。在筛选过程中，排除了涉及多个实体或者无法匹配到实体的项目。接下来，将项目的 ID 与子知识图谱中的三元组头实体的 ID 进行匹配，并从中选择出已匹配的三元组。

表 4.1 MNCL 算法流程

 算法 1: MNCL 模型训练过程（伪代码）

输入：

- 用户-物品交互矩阵 $\mathbf{Y} \in \mathbb{R}^{M \times N}$,
- 知识图谱 $\mathcal{G} = \{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{R}\}$
- 模型总训练次数 \mathcal{N}_e
- 批处理大小 \mathcal{N}_b
- 对比学习损失权重 α

 1) 创建项目-实体稀疏图 \mathcal{G} , 用户-项目交互稀疏图 \mathcal{S} , 用户-项目-实体稀疏图 \mathcal{M}

 2) FOR epoch IN \mathcal{N}_e DO

 3) 从图 \mathcal{G} 、图 \mathcal{S} 和图 \mathcal{M} 中采样批次大小的 \mathcal{N}_b 的数据

 4) 初始化神经网络参数、用户特征 \mathbf{e}_u 以及项目特征 \mathbf{e}_i

5) //构建不同子图

 6) 数据增强获得子图 \mathcal{g}, m

7) //GNN 层消息聚合

 8) 用户-项目交互稀疏图 $\mathcal{S} \rightarrow \mathbf{e}_u^s, \mathbf{e}_i^s$

 9) 子图 $\mathcal{g} \rightarrow \mathbf{e}_i^g$, 子图 $m \rightarrow \mathbf{e}_u^m, \mathbf{e}_i^m$

10) //对比任务

 11) $\mathcal{L}^G \leftarrow \mathbf{e}_u^s, \mathbf{e}_i^s$ 和 $\mathbf{e}_u^m, \mathbf{e}_i^m$

 12) $\mathcal{L}^{KG} \leftarrow \mathbf{e}_i^g, \mathbf{e}_i^s$

 13) $\mathcal{L}_{CL} \leftarrow \mathcal{L}^G + \mathcal{L}^{KG}$

14) //推荐任务

 15) $\mathbf{e}_u^s \| \mathbf{e}_u^m \| \mathbf{e}_i^m \rightarrow \mathbf{e}_u$, $\mathbf{e}_i^s \| \mathbf{e}_i^m \| \mathbf{e}_i^g \rightarrow \mathbf{e}_i$

 16) $\mathcal{L}_{BPR} \leftarrow$ 推荐任务损失计算

17) //多任务学习

 18) $\mathcal{L}_{MNCL} \leftarrow \mathcal{L}_{BPR} + \alpha \mathcal{L}_{CL}$

19) END FOR

 输出: MNCL 模型

4.4.2 实验环境

所有实验使用集成了 2 块 GPU (NVIDIA GeForce RTX3090, 16GB) 的工作站上实现, 编程语言为 Python, 具体设置如表 3.3 所示。

4.4.3 评估指标

本文在以下的实验场景中评估了本文的方法: 在点击率 (CTR) 预测中, 将训练好的模型应用于测试集中的每个交互来进行预测。本文采用了两个广泛使用的指标 AUC 和 F1 来评估 CTR 预测的性能。

4.4.4 实验模型对比

为了验证 MNCL 模型的性能, 本文选取了 14 个模型进行对比分析, 这些模型包括基于协同过滤算法的推荐模型、基于知识图谱的推荐模型、基于 GNN 的推荐模型以及最新的运用了对比学习方法的知识感知推荐模型, 同时为了验证路径感知模块对模型的提升以及噪音增强方式的可泛化能力, 本章增加了第三章的模型以及其变体进行相应的对比实验。

(一) 模型比较

基于协同过滤算法的推荐模型:

- **BPRMF**^[54]: 这是一种典型的基于 CF 的方法, 它使用两两矩阵分解来进行由 BPR 损失优化的隐式反馈。

基于知识图谱的推荐模型:

- **PER**^[56]: 这是一种典型的基于路径的方法, 它提取基于元路径的特征来表示用户和项目之间的连接。
- **CKE**^[57]: CKE 模型将协同过滤算法和文本、结构化信息以及视觉信息融合在统一的推荐框架中。
- **RippleNet**^[35]: RippleNet 模型将知识图谱结合到推荐算法中, 利用偏好传播方法, 自动对用户兴趣特征进行建模。
- **CurvRec**^[72]: 基于知识图谱推荐模型, 通过融合混合曲率流形和图卷积网络, 有效地利用知识图谱的几何结构来提升推荐性能。

基于 GNN 的推荐模型：

- **KGCN**^[36]: KGCN 利用图卷积神经网络，挖掘项目在知识图谱中的高阶信息，对于不同实体特征进行重要性区分。
- **KGNN-LS**^[58]: KGNN-LS 考虑了在图卷积中用户对不同知识图三元组的偏好，并引入标签平滑作为正则化，以确保在知识图中相邻项目之间具有相似的用户偏好权重。
- **KGAT**^[37]: KGAT 引入了协同知识图的概念，在用户-物品-实体图上应用注意力聚合，注意力分数反映了知识图三元组的重要性。
- **KGIN**^[59]: KGIN 模型对用户和项目的交互建模成用户意图图，通过挖掘图中长范围连接的语义性以及捕捉用户和项目的交互特征，来提升推荐性能。

基于对比学习方法的知识感知推荐模型：

- **KGIC**^[60]: 这是一种基于知识感知推荐模型，该模型运用了对比学习方法，探索了交互式对比学习机制。
- **MCCLK**^[68]: MCCLK 以分层的方式进行对比学习，从而考虑用户-项目-实体图的结构信息。
- **KACL**^[69]: KACL 给知识图中每个关系加上一个可学习的嵌入向量，并利用对比学习缓解训练中用户-项目视图占主导的问题。

论文第三章模型及其变体：

- **MVCK**: 此论文第三章推荐模型，MVCK 利用多视图构建不同的子图，这些子图经过不同的图编码器之后进行多负例对比学习充分学习用户-项目的特征表示。
- **MVCK⁺**: 在 MVCK 模型上面增加了噪声增强。

(二) 参数设置

本章实验部分使用 Pytorch 实现了 MNCL 和所有基准模型，并仔细调整了关键参数。本章将所有模型的嵌入大小固定为 64，嵌入参数使用 Xavier 方法进行初始化。本章使用 Adam 优化器来优化算法，同时将批量大小设置为 4096。利用网格搜索确认最佳参数设置，调整学习率 η_1 在 $\{0.001, 0.003, 0.01, 0.03\}$ 之间，以及 L2 正则化项 η_2 在 $\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$ 之间。表 4.2 中提供了其他超参

数设置。对于所有比较方法中的超参数，通过经验研究或遵循原始论文，找到了最佳的设置。

表 4.2 模型 MNCL 的参数设置

Dataset	α	K	L	β	ω
MovieLens-1M	0.1	2	4	1.5	0.7
Book-Crossing	0.1	2	4	1.5	0.8
Last.FM	0.1	2	4	1.5	0.8

其中, α : 对比损失权重, K : 项目视图 Light-GCN 层数, L : 加入优化项的 Light-GCN 层数, β : 优化项参数, ω : 对比学习中原视图权重。

4.5 实验结果以及分析

4.5.1 实验结果综合对比

在本节中，将 MNCL 模型与上述对比实验模型在三个基准数据集上面进行性能对比。为了保证实验结果的可靠性，所有的模型都取的最优结果。实验数据对比结果如表 4.3 所示。

本章通过将对比模型以及实验模型在上述的三个数据集上面进行比较，来测试本章实验模型的性能。

(1) 将 CKE 和 BPRMF 相比较可知，引入知识图谱作为辅助信息的重要性。通过将 KGCN、KGAT 与 CKE 比较可知，GNN 具有很强的图学习能力。大多数基于 GNN 的方法比嵌入目标方法和基于路径的方法表现得更好，这表明了引入 GNN 汇聚项目高阶信息的有效性。

(2) MVCK+模型与 MVCK 比较可知，噪音增强的方式具有泛化性，再其他模型上也同样适用。在神经网络汇聚过程中增加一定的噪音增强能够优化样本的均匀性分布，提升对比学习的性能。双视图模型 MVCK 与三视图模型 MNCL 比较可知。增加合适的视图能够更好的挖掘用户以及项目的学习表示。同时，从不同的视角出发学到更加全面更加丰富的样本进行对比学习，更有益于对比学习挖掘信息的潜力。

(3) 本文提出的 MNCL 模型在三个数据集上都有很好的表现。相较于最新的 MCCLK 模型在三个数据集上面都有更好的表现。实验结果证明在图神经网络聚合过程中加入一定程度的噪音增强并没有降低模型的性能，反而能够提升模型

性能,证明了噪音增强方式的有效性,能够很好的缓解节点在嵌入空间中无法区分的退化问题。更进一步,优化嵌入空间重节点的均匀化分布有益于对比学习的性能,从而学习到更加准确的节点表示,同时也能够缓解长尾问题,增加推荐模型的准确性。

表 4.3 各模型的实验数据对比

Model	MovieLens-1M		Book-Crossing		Last.FM	
	AUC	F1	AUC	F1	AUC	F1
BPRMF	0.8923	0.7894	0.6586	0.6117	0.7562	0.7014
PER	0.7126	0.6670	0.6052	0.5732	0.6408	0.6042
CKE	0.9072	0.8026	0.6752	0.6098	0.7423	0.6715
RippleNet	0.9195	0.8420	0.7209	0.6472	0.7765	0.7021
KGCN	0.9095	0.8375	0.6842	0.6321	0.8025	0.7095
KGNN-LS	0.9152	0.8413	0.6765	0.6269	0.8053	0.7219
KGAT	0.9146	0.8433	0.7318	0.6524	0.8148	0.7426
KGIN	0.9162	0.8435	0.7264	0.6608	0.8492	0.7608
KGIC	0.9252	0.8559	0.7589	0.6712	0.8592	0.7753
KACL	0.9318	0.8589	0.7602	0.6721	0.8615	0.7786
CurvRec	0.9349	0.8619	0.7607	0.6726	0.8530	0.7658
MCCLK	0.9351	0.8631	0.7625	0.6777	0.8763	0.7964
MVCK	0.9360	0.8632	0.7629	0.6781	0.8754	0.7985
MVCK ⁺	0.9377	0.8659	0.7702	0.6822	0.8829	0.7986
MNCL	0.9396	0.8700	0.7736	0.6892	0.8951	0.8095

4.5.2 消融实验

为了验证模型的每个模块对最终结果的影响,本节进行了消融实验。比较 MNCL 模型的几个变体,并分析了这些模块的影响。其中 $MNCL_{w/o}^L$ 表示去掉对比学习来评估对比学习方法对模型的影响。 $MNCL_{w/o}^N$ 表示将噪音增强 Light-GCN 的模块换成一般的 Light-GCN 模块。 $MNCL_{w/o}^G$ 表示将用户-项目-实体视图去掉来评估多视图对模型的影响。最后的结果如表 4.4 所示。

由表 4 的消融实验结果可以得出,不同的模块对于 MNCL 模型均有贡献。

从中可以得到如下结论：（1）去掉用户-项目-实体视图模块后模型性能下降可知，每部分视图都对更加准确的推荐均有贡献，用户和项目可以从不同的视图学习不同角度的信息来丰富最终的向量表示，并且更全面更丰富的样本更加有利于对比学习的能力。（2）去掉噪音增强后，模型推荐性能会明显下降，特别是对于较稀疏的数据集 MovieLens-1M 以及 Book-Crossing，因为稀疏数据集上面消息聚合之后节点退化问题更加严重，因此证明我们提出的数据增强方式能够有效的缓解节点无法区分导致的退化问题，优化节点在嵌入空间层面的均匀性分布，有利于长尾项目，提高推荐的准确性。（3）融入对比学习进行多任务学习比单独的推荐任务有更好的性能。

表 4.4 MNCL 及其消融变体的影响

Model	MovieLens-1M		Book-Crossing		Last.FM	
	AUC	F1	AUC	F1	AUC	F1
$MNCL_{w/o}^{CL}$	0.9343	0.8641	0.7615	0.6779	0.8905	0.8025
$MNCL_{w/o}^N$	0.9351	0.8655	0.7624	0.6798	0.8837	0.7992
$MNCL_{w/o}^G$	0.9367	0.8659	0.7656	0.6813	0.8829	0.7986
MNCL	0.9396	0.8700	0.7736	0.6892	0.8951	0.8095

4.5.3 超参敏感性分析

（1）对比学习权重 α 的敏感性分析

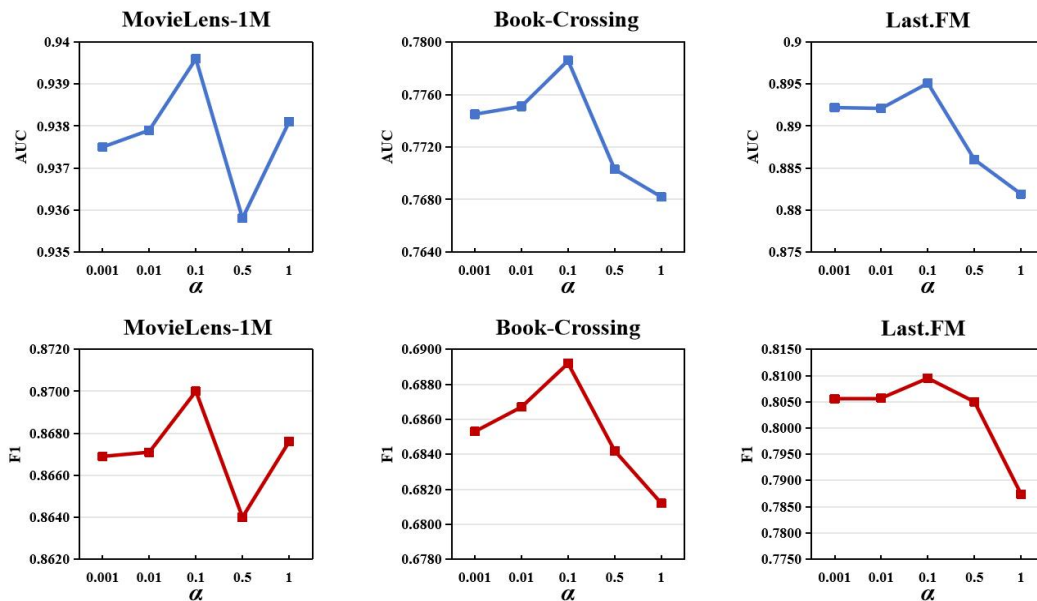


图 4.2 对比学习权重 α 对模型性能的影响

本章采用的是多任务训练策略来优化整个模型。本节根据对比学习权重占比 α 设置了调参实验， α 在 $\{0.001, 0.01, 0.1, 0.5, 1\}$ 之间取值，来探究对比学习在整个多任务学习过程中的影响。实验结果如图 4.2 所示。在三个公开的基准数据集上面，当 α 等于 0.1 时，模型的推荐效果最好，当 α 继续增强效果反而会急剧下降，说明在多任务中，当以推荐任务作为主导时更有利于模型的推荐性能。

(2) 噪声增强系数 β 的敏感性分析

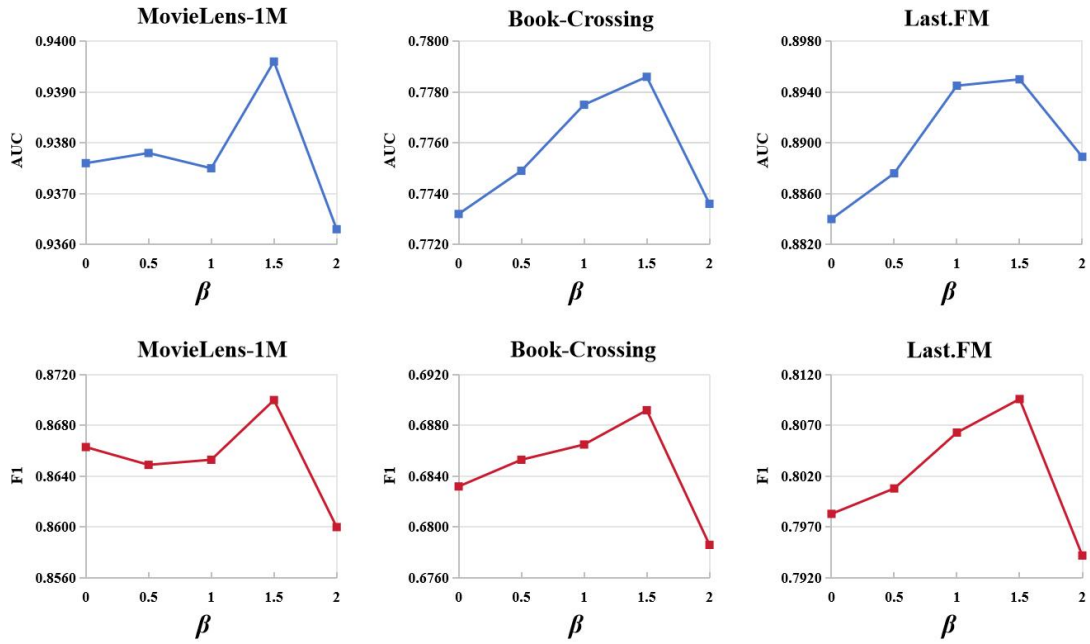


图 4.3 噪声增强系数 β 对模型性能的影响

为了研究噪音增强系数 β 对于模型性能的影响，本节将噪音增强系数 β 在 $\{0, 0.1, 0.5, 1, 1.5, 2\}$ 之间取值。实验结果如图 4.3 所示，从图中可以得知在数据集 MovieLens-1M、Book-Crossing 和 Last.FM 上当 $\beta=1.5$ 时模型性能均能最佳。随着 β 的增大，不管是稠密的数据集 Last.FM 还是极度稀疏的数据集 MovieLens-1M 都能够发现模型性能显著上升，当 β 的值到达 1.5 的时候，模型性能最佳，证明了数据增强方式的可泛化性。然而随着 β 的值持续增加，模型的性能反而会显著下降，造成这样的结果可能的原因是当 β 的值越大时，模型引入了过多的噪音干扰，影响了最终的推荐性能，也解释了本文没有在知识图谱中的 GNN 加入噪音增强方式来学习最终表示，因为知识图谱中已经存在很多噪音，采用噪音增强的方式反而会影响知识图谱中项目的最终向量表示的学习。

(3) 带优化项的 Light-GCN 层数 L 的敏感性分析

针对用户-项目视图中噪音增强的 Light-GCN 层数 L 对于模型性能的影响，本节调整了 Light-GCN 层数 L 的范围变化在 $\{1, 2, 3, 4, 5\}$ 之间。实验结果如图 4.4 所示，

当 $L=4$ 时，模型在三个公开数据集上面性能最佳。由于用户-项目交互数据的极度稀疏，要学习到更加准确的嵌入表示往往需要更深的图神经网络，特别是加入噪音增强之后，嵌入空间中节点分布更均匀，更深的网络优势明显。然而随着网络层数超过4层之后，模型性能反而下降的明显，分析造成这个结果的原因可能是随着嵌入向量在多层图神经网络不断深入的学习，节点学习表示重新变得无法区分，影响了节点的学习，从而降低了模型预测的准确性。上述分析反映了基于GNN的推荐模型存在的退化问题一直存在，本文提出的噪音增强方式在合适的情境下能够很好的缓解这个问题。

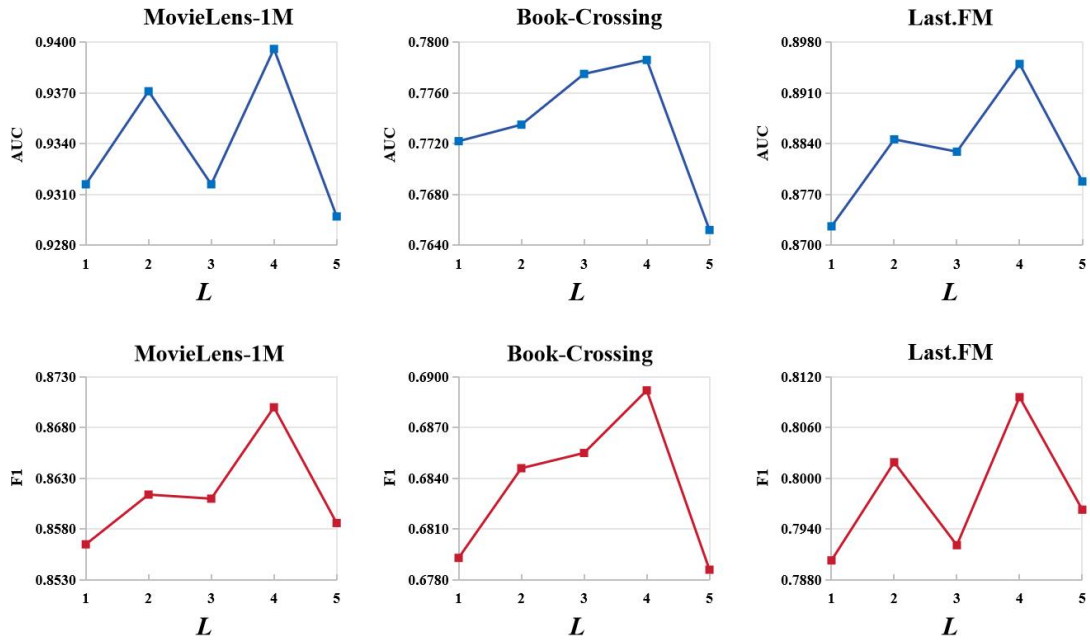


图 4.4 噪音增强的 Light-GCN 层数 L 对模型性能的影响

(6) 项目视图 Light-GCN 层数 K 的敏感性分析

图 4.5 显示了项目-实体视图模块的 Light-GCN 层数 K 对于模型性能的影响。从图中可以得知在三个数据集上当 $K=2$ 时模型性能最佳。由此可知，一到两跳对于在项目-实体视图中聚合邻居信息是足够的，随着项目-实体视图中 Light-GCN 神经网络层数升高性能下降很明显。最主要的原因是知识图谱中包含了很多的噪音信息，这些冗余的信息被编码到嵌入向量中，影响了最终的评分预测，因此本文并没有考虑在项目-实体视图中的 Light-GCN 加入噪音增强的方式在，这样做很可能导致模型的性能下降。

(7) 对比学习原视图权重 ω 的敏感性分析

本文设计了多负例的对比学习方法，同时分析了原视图中的负例权重对于对比学习最终性能的影响。由图 4.6 可以得出结论，在数据集 Last.FM 和数据集上

在 $\omega=0.8$ 时对比学习性能最佳,在数据集 MovieLens-1M 和 Book-Crossing 时 $\omega=0.7$ 对比学习性能最佳。权重 ω 解释了不同视图中样本对于对比学习性能的影响,在经典的对比学习方法中,大部分模型采用的对比方法只利用了对比视图中的负样本,往往忽略正样本视图中存在其他潜在信息。本文提出的多负例对比学习方法与多视图的思想不谋而合,都是希望通过不同角度来学习到更加全面更加丰富的用户以及项目向量表示,从而提升推荐的准确性。

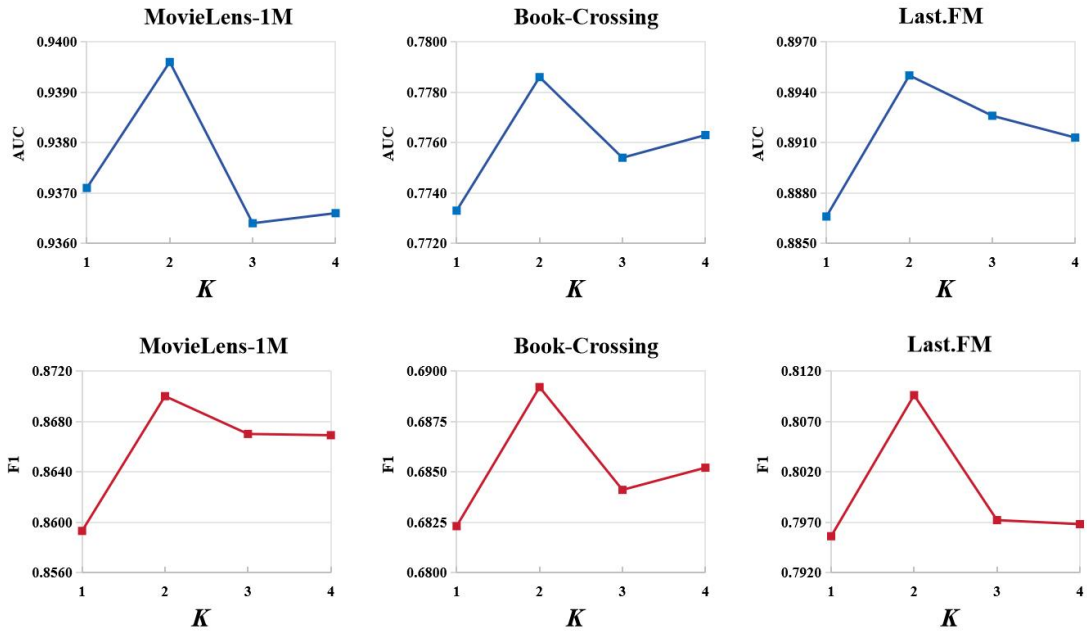


图 4.5 项目视图 Light-GCN 层数 K 对模型性能的影响

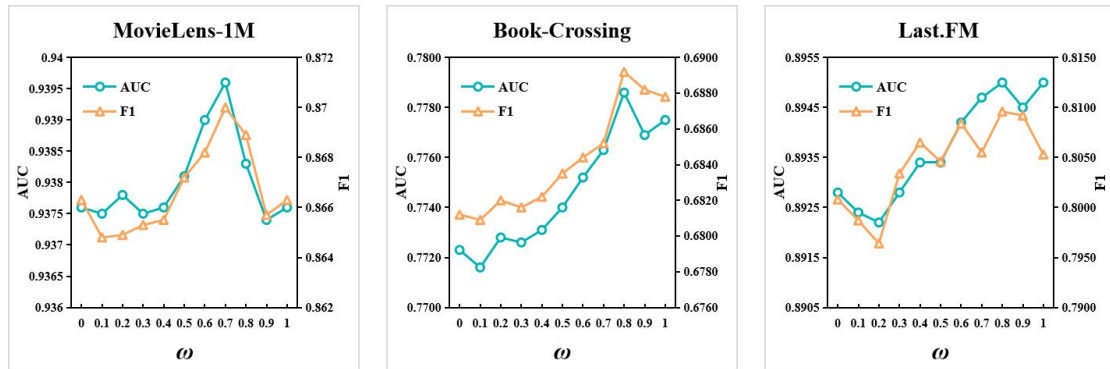


图 4.6 对比学习原视图权重 ω 对模型性能的影响

4.6 本章小结

在本章中,针对目前图卷积神经网络聚合后,样本分布不均匀导致的长尾问题,提出了优化图卷积神经网络的知识图谱对比学习推荐模型 MNCL,并进行了充分的实验验证了模型的性能。模型 MNCL 主要模块包括多视图的构建、

Light-GCN 层、噪声增强的 Light-GCN 层、对比学习模块、预测模块。模型的关键是多视图的构建以及噪声增强 Light-GCN 层、对比学习模块。首先需要利用知识图谱构建不同的视图，其中包括用户-项目视图，项目-实体视图以及用户-项目-实体视图。然后经过不同的图编码器获得最后的表征特征，进行不同的对比学习之后，进行最后推荐预测评分。本模型最终在三个公开基准数据集上与不同方法的多种优秀的模型进行对比，并设计了消融实验，验证了模块不同模块都具有优秀的贡献，以及模型整体具有优秀的推荐性能。

第 5 章 总结与展望

5.1 论文工作总结

本文主要探索了对比学习在基于 GNN 的推荐模型中的应用，利用对比学习能够挖掘数据本身监督信号的能力，缓解当前基于 GNN 的推荐模型中依然存在的问题。接下来对文章整体工作进行总结。

本文首先在第一章介绍了推荐模型的必要性以及目前仍然存在的一些问题，其中包括训练信号稀疏问题，以及基于 GNN 推荐模型聚合后样本分布不均产生的长尾问题等。针对这些问题，提出了本文的两种对应解决办法的模型。在第二章对提出的模型所用到的相关理论以及技术进行了简述。

本文的核心工作为第三章以及第四章。针对基于 GNN 的推荐模型存在的训练信号稀疏问题，第三章提出了多视图对比学习的知识感知推荐模型 (MVCK)。首先在子图构建模块，对用户-项目以及项目-实体两个视图采用不同的丢边、丢点策略生成不同的子图。然后在消息聚合模块学习不同子图中用户、项目的向量特征。接着对学习到的用户、向量特征采用多负例对比学习充分挖掘数据潜在信息。最后在 Movielens, Last.FM 和 Book-Crossing 三个基准数据集上进行了充分的实验。实验结果表明 MVCK 模型具有优秀的性能，且能够有效缓解训练信号稀疏问题。

针对基于 GNN 推荐模型信息聚合后嵌入空间中样本分布不均匀产生的长尾问题。第四章提出了多视图噪声增强图对比学习的知识感知推荐模型 (MNCL)。在第三章中，证明了多视图能够从多视角更全面丰富的挖掘数据中的潜在信息。第四章保留了 MVCK 模型中的部分模块，并针对所提出的问题加入了用户-项目-实体视图模块以及加入噪声增强的 Light-GCN。在子图构建模块，与 MVCK 模型不同，MNCL 模型除了用户-项目以及项目-实体两个视图还加入了用户-项目-实体用户-项目-实体视图，继而对三个视图构建相应的子图。然后在消息聚合模块，MVCK 模型的用户-项目子图会进入噪声增强后的 Light-GCN 层进行学习。接着对三个子图学习到的用户以及项目的特征向量进行充分的对比学习。最后，在 Movielens, Last.FM 和 Book-Crossing 三个基准数据集上进行了充分的实验，实验结果表明 MNCL 模型性能优于对比的所有模型，验证了 MNCL 模型各个模块的有效性。

5.2 未来工作展望

本文根据现有基于 GNN 的推荐模型研究基础和现实情况进行分析，探索了对比学习在知识感知推荐模型中的应用，并针对存在的部分问题，提出了基于多视图对比学习的知识感知推荐模型（MVCK）以及多视图噪声增强图对比学习的知识感知推荐模型（MNCL）。虽然提出了模型都取得了一些成果，并且通过大量实验验证了模型的可解释性和泛化性，但本文的工作仍然存在不足。基于目前工作中存在的不足之处提出以下展望：

（1）MVCK 模型构造了多个子图。虽然能够在一定程度上提升模型的性能，模型的工作量巨大。在之后的工作可能需要探讨更加高效的子图构建方法。同时多子图可能在消息聚合中引入更多的噪音，因此导致模型的准确性下降，可能引入注意力机制能够缓解噪音带来的干扰。

（2）本文提出的 MNCL 模型在加入全局模块以及噪声增强后的 Light-GCN 模型后在性能上有着极大地提升。但是神经网络是动态的并且复杂的，无法准确预测其中的具体工作，噪声增强的适应性应该要结合现实考虑。同时 MNCL 模型需要构造不同的视图，计算成本比较大，而真实的推荐场景中，时间也是衡量模型的标准之一。为了增加模型的适用性以及鲁棒性，未来的工作应该考虑对模型的结构进行优化，同时要结合实际情况设计更加合适的对比学习方法。

（3）知识图谱具有丰富的信息，其中还包括关系信息。本文的两种模型虽然基于关系构建了项目-实体的视图，但是并没有充分利用关系这一信息。不同于其他图结构，知识图谱的关系具有方向性，也就是说一个节点的出度以及入度是可知的，这是知识图谱中重要的组成部分。未来可以考虑构建关系的向量，充分利用知识图谱中的信息。

参考文献

- [1] 黄立威, 江碧涛, 吕守业等. 基于深度学习的推荐模型研究综述[J]. 计算机学报, 2018, 41(7): 1619-1647.
- [2] 许海玲;吴潇;李晓东;阎保平.互联网推荐模型比较研究[J].软件学报,2009,20(02):350-362.
- [3] Su X , Khoshgoftaar T M .A Survey of Collaborative Filtering Techniques[J].Advances in Artificial Intelligence, 2009, 2009(12).
- [4] Liu Q , Ma H , Chen E ,et al.A SURVEY OF CONTEXT-AWARE MOBILE RECOMMENDATIONS[J].International Journal of Information Technology & Decision Making, 2013, 12(01):139-172.
- [5] Koren Y .Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]//Knowledge Discovery and Data Mining.ACM, 2008.
- [6] Zheng L, Noroozi V, Yu P. S. Joint deep modeling of users and items using reviews for recommendation[J]. ACM Computing Surveys, 2018: 425-434.
- [7] Zhang S , Yao L , Sun A ,et al.Deep Learning Based Recommender System: A Survey and New Perspectives[J].ACM Computing Surveys, 2017.
- [8] He X , Liao L , Zhang H ,et al.Neural Collaborative Filtering[J].International World Wide Web Conferences Steering Committee, 2017.
- [9] 周飞燕, 金林鹏, 董军. 卷积神经网络研究综述[J]. 计算机学报, 2017, 40(6): 1229-1251.
- [10] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[J]. arXiv preprint arXiv:1205.2618, 2012.
- [11] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions[J]. IEEE transactions on knowledge and data engineering, 2005, 17(6): 734–749.
- [12] Baeza-Yates, R., B. Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley, 1999.
- [13] Belkin, N. and B. Croft. Information filtering and information retrieval. Communications of the ACM, 35(12):29-37, 1992.
- [14] Jia Z, Yang Y, Gao W, et al. User-based collaborative filtering for tourist attraction recommendations[C]//2015 IEEE international conference on computational intelligence & communication technology. IEEE, 2015: 22-25.
- [15] Xue F, He X, Wang X, et al. Deep item-based collaborative filtering for top-n recommendation[J]. ACM Transactions on Information Systems (TOIS), 2019, 37(3): 1-25.
- [16] Covington P, Adams J, Sargin E. Deep neural networks for youtube recommendations [C]//Proceedings of the 10th ACM conference on recommender systems.2016: 191-198.

- [17] Liu H, Wen J, Jing L, et al. Deep generative ranking for personalized recommendation[C]//Proceedings of the 13th ACM Conference on Recommender Systems. 2019: 34-42.
- [18] Abdi H. Singular value decomposition (SVD) and generalized singular value decomposition[J]. Encyclopedia of measurement and statistics, 2007, 907: 912.
- [19] Rendle S. Factorization machines[C]//2010 IEEE International conference on data mining. IEEE, 2010: 995-1000.
- [20] He X, Liao L, Zhang H, et al. Neural collaborative filtering[C]//Proceedings of the 26th international conference on world wide web. 2017: 173-182.
- [21] He X, Liao L, Zhang H, et al. Neural collaborative filtering[C]//Proceedings of the 26th international conference on world wide web. 2017: 173-182.
- [22] Zhang A, Fawaz N, Ioannidis S, et al. Guess who rated this movie: Identifying users through subspace clustering[J]. arXiv preprint arXiv:1208.1544, 2012.
- [23] Bengio Y , Courville A , Vincent P. Representation Learning: A Review and New Perspectives[J]. 2012.
- [24] Cheng H. T., Koc L, Harmsen J, Shaked T, Chandra T, Aradhye H, Ghemawat S. Wide & deep learning for recommender systems. In Proceedings of the 1st workshop on deep learning for recommender systems, 2016, 7-10.
- [25] Xiaoxue L, Xuesong B, Longhe W, et al. Review and trend analysis of knowledge graphs for crop pest and diseases[J]. IEEE Access, 2019, 7: 62251-62264.
- [26] WANG Q, MAO Z D, WANG B, et al. Knowledge graph embedding: a survey of approaches and applications[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(12): 2724-2743.
- [27] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[J]. Advances in neural information processing systems, 2013, 26.
- [28] Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes[C]//Proceedings of the AAAI conference on artificial intelligence. 2014, 28(1).
- [29] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion[C]//Proceedings of the AAAI conference on artificial intelligence. 2015, 29(1).
- [30] Ji G, He S, Xu L, et al. Knowledge graph embedding via dynamic mapping matrix[C]//Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers). 2015: 687-696.

-
- [31] Yang B, Yih W, He X, et al. Embedding entities and relations for learning and inference in knowledge bases[J]. arXiv preprint arXiv:1412.6575, 2014.
 - [32] Yu X, Ren X, Sun Y, et al. Personalized entity recommendation: A heterogeneous information network approach[C]//Proceedings of the 7th ACM international conference on Web search and data mining. 2014: 283-292.
 - [33] Zhao H, Yao Q, Li J, et al. Meta-graph based recommendation fusion over heterogeneous information networks[C]//Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017: 635-644.
 - [34] Sun Z, Yang J, Zhang J, et al. Recurrent knowledge graph embedding for effective recommendation[C]//Proceedings of the 12th ACM conference on recommender systems. 2018: 297-305.
 - [35] Wang H, Zhang F, Wang J, et al. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems[C]//Proceedings of the 27th ACM international conference on information and knowledge management. 2018: 417-426.
 - [36] Wang H, Zhao M, Xie X, et al. Knowledge graph convolutional networks for recommender systems[C]//The world wide web conference. 2019: 3307-3313.
 - [37] Wang X, He X, Cao Y, et al. Kgat: Knowledge graph attention network for recommendation[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 950-958.
 - [38] Wu J, Wang X, Feng F, et al. Self-supervised graph learning for recommendation[C]//Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. 2021: 726-735.
 - [39] Wang H, Zhang F, Xie X, et al. DKN: Deep knowledge-aware network for news recommendation[C]//Proceedings of the 2018 world wide web conference. 2018: 1835-1844.
 - [40] Nickel M, Tresc V, Krieger H P. A three-way model for collective learning on multi-relational data[C]//Icml. 2011, 11(10.5555): 3104482.3104584.
 - [41] Nickel M, Rosasco L, Poggio T. Holographic embeddings of knowledge graphs[C]//Proceedings of the AAAI conference on artificial intelligence. 2016, 30(1).
 - [42] Sun Y, Han J. Mining heterogeneous information networks: principles and methodologies[J]. Synthesis Lectures on Data Mining and Knowledge Discovery, 2012, 3(2): 1-159.
 - [43] Yu X, Ren X, Gu Q, et al. Collaborative filtering with entity similarity regularization in heterogeneous information networks[J]. IJCAI HINA, 2013, 27.

- [44] Sun Z, Yang J, Zhang J, et al. Recurrent knowledge graph embedding for effective recommendation[C]//Proceedings of the 12th ACM conference on recommender systems. 2018: 297-305.
- [45] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[J]. Advances in neural information processing systems, 2017, 30.
- [46] Wu J, Wang X, Feng F, et al. Self-supervised graph learning for recommendation[C]//Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. 2021: 726-735.
- [47] Chen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations[C]//International conference on machine learning. PMLR, 2020: 1597-1607.
- [48] Gutmann M, Hyvärinen A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models[C]//Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2010: 297-304.
- [49] Cao Y, Wang X, He X, et al. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences[C]//The world wide web conference. 2019: 151-161.
- [50] Liu X, Zhang F, Hou Z, et al. Self-supervised learning: Generative or contrastive[J]. IEEE transactions on knowledge and data engineering, 2021, 35(1): 857-876.
- [51] Jie Chen, Haw-ren Fang, and Yousef Saad. 2009. Fast Approximate kNN Graph Construction for High Dimensional Data via Recursive Lanczos Bisection. Journal of Machine Learning Research (2009).
- [52] Yu Chen, Lingfei Wu, and Mohammed Zaki. 2020. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. Advances in Neural Information Processing Systems (2020).
- [53] He X, Deng K, Wang X, et al. Light-GCN: Simplifying and powering graph convolution network for recommendation[C]//Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 2020: 639-648.
- [54] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[J]. arXiv preprint arXiv:1205.2618, 2012.
- [55] Rendle S. Factorization machines with libfm[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2012, 3(3): 1-22.

- [56] Yu X, Ren X, Sun Y, et al. Personalized entity recommendation: A heterogeneous information network approach[C]//Proceedings of the 7th ACM international conference on Web search and data mining. 2014: 283-292.
- [57] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In SIGKDD. 353–362.
- [58] Wang H, Zhang F, Zhang M, et al. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 968-977.
- [59] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning Intents behind Interactions with Knowledge Graph for Recommendation. In WWW. 878–887.
- [60] Zou D, Wei W, Wang Z, et al. Improving knowledge-aware recommendation with multi-level interactive contrastive learning[C]//Proceedings of the 31st ACM international conference on information & knowledge management. 2022: 2817-2826.
- [61] Gao J, He D, Tan X, et al. Representation degeneration problem in training natural language generation models[J]. arXiv preprint arXiv:1907.12009, 2019.
- [62] Scarselli F, Gori M, Tsoi A C, et al. The graph neural network model[J]. IEEE transactions on neural networks, 2008, 20(1): 61-80.
- [63] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.
- [64] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]//Proceedings of the 2008 ACM SIGMOD international conference on Management of data. 2008: 1247-1250.
- [65] Velickovic P, Cucurull G, Casanova A, et al. Graph attention networks[J]. stat, 2017, 1050(20): 10.48550.
- [66] Lin Z, Tian C, Hou Y, et al. Improving graph collaborative filtering with neighborhood-enriched contrastive learning[C]//Proceedings of the ACM web conference 2022. 2022: 2320-2329.
- [67] Yu J, Yin H, Gao M, et al. Socially-aware self-supervised tri-training for recommendation[C]//Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021: 2084-2092.

- [68] Zou D, Wei W, Mao X L, et al. Multi-level cross-view contrastive learning for knowledge-aware recommender system[C]//Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022: 1358-1368.
- [69] Wang H, Xu Y, Yang C, et al. Knowledge-adaptive contrastive learning for recommendation[C]//Proceedings of the sixteenth ACM international conference on web search and data mining. 2023: 535-543.
- [70] Yang Y, Huang C, Xia L, et al. Knowledge graph contrastive learning for recommendation[C]//Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. 2022: 1434-1443.
- [71] Yu J, Xia X, Chen T, et al. XSimGCL: Towards extremely simple graph contrastive learning for recommendation[J]. IEEE Transactions on Knowledge and Data Engineering, 2023.
- [72] Wang J, Shi Y, Yu H, et al. Mixed-curvature manifolds interaction learning for knowledge graph-aware recommendation[C]//Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023: 372-382.

致 谢

时间飞逝，转眼间已经在武汉大学过去了四年。还记得当初下定决心跨专业考研时候的一腔热血，把握最后一次机会尽自己最大努力考入这所梦校。与武大的缘分可以追溯到高中，作为一名湖北人一直向往着这所全国最美大学，庆幸最终成功了。这四年与我而言无疑是最难忘的，非科班出身的我在这所国内顶尖学府有一些不知所措，然而自己也并没有那么努力去证明自己，因此一度陷入自我怀疑的沼泽。这种感觉一直伴随我的研究生四年。但是，我并没有因此而抱怨，我一直信奉的是自己的做的选择坚决不要后悔，幸运的是一切的结局都是好的。在这四年里，我收获得不止是学业上的进步。

兜兜转转，终于临近毕业了。在我即将完成硕士学业的时刻，也是结束我二十年学业生涯的时刻，我想对硕士期间的四年时间里帮助过我的人表达我最真挚的感激之情。

首先，我想感谢我的导师王健教授。因为跨专业的缘故，我在专业技能上面了解的并不是很多，在我众多的邮件对象里面，是王老师热情的回应了我的消息，谢谢王老师给我这次机会。在今后的日子里，是王老师耐心的检查我的每周的任务，并以专业知识指导我，改进我的错误，引领我走进了科研的大门。

其次，我想感谢帮助过我的课题组的师兄弟们。感谢李川师兄帮助我完成人生中的第一篇论文。感谢王靖雄师弟耐心的跟我分享专业知识，从他的身上我学习到了真正的科研态度以及奉献精神。还要感谢课题组其他师兄弟们对我的科研上面的问题提出的建议。同时，还要感谢我研究生时间的舍友们，是他们才能让我快速的融入这个新的环境。

最后，我最要感谢的是一直支持着我的家人。研究生四年，父母一直是最关心我毕业的人，因为怕我的压力大，所以他们并没有一直追问，每次提问的时候也总是关心着我的情绪。是他们让我给了我无限的动力，让我相信自己一定能够成功，他们是我面对一切困难最坚强的后盾。

硕士期间研究成果