

北京交通大学

硕士专业学位论文

基于 HIN 的图神经网络社会推荐算法研究
Research on Social Recommendation Algorithm of Graph Neural
Network Based on Heterogeneous Information Network

北京交通大学

2021 年 5 月

学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。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。

(保密的学位论文在解密后适用本授权说明)

学位论文作者签名: 

导师签名: 

签字日期: 2021 年 5 月 31 日

签字日期: 2021 年 5 月 31 日

学校代码：10004

密级：公开

北京交通大学

硕士专业学位论文

基于 HIN 的图神经网络社会推荐算法研究
Research on Social Recommendation Algorithm of Graph Neural
Network Based on Heterogeneous Information Network

作者姓名：张纯 学号：19125067

导师姓名：郭宇春 职称：教授

专业学位类别：电子与通信工程 学位级别：硕士

北京交通大学
2021 年 5 月

致谢

首先要感谢我的导师郭宇春教授，本文的研究工作是在她的耐心指导下完成的。从课题的选定到内容的完成，郭老师给予我耐心的指导，在我遇到问题的时候认真帮我解答。郭老师治学严谨、学识渊博、对工作精益求精，给我在科研的道路上树立了良好的榜样，在生活中，郭老师平易近人、温婉细致，给了我很多支持和帮助，在此由衷地感谢郭老师在两年的学习生涯中给我的悉心关怀和指导。

感谢陈一帅老师对我的鼓励和指导，在我遇到问题向他请求帮助时，陈老师总是能给我一些有效的建议，并提供大量的参考资料。陈老师饱满热情的工作态度和积极的生活态度对我影响深远，会以他为榜样努力工作、学习、生活。

感谢实验室的其他所有老师。感谢赵永祥、郑宏云、李纯喜、孙强、张立军、张梅等老师在研究生期间给予我的帮助。在此向各位老师致以真挚的感谢。

另外，在实验室学习和工作期间，尹姜谊师姐、孙欢师姐、戚余航师兄、刘翔师兄、李想师兄、杨晶晶师姐、王珍珠师姐、王冰同学和一起学习玩耍的同窗们给了我很多帮助，在此向他们表达诚挚的谢意。

最后，感谢我的家人一直以来对我的悉心付出和支持，在漫长的求学生涯中，是他们一直在身后默默地鼓励我，成为我坚实的后盾，是他们成就了今天的我。

摘要

得益于互联网的快速发展，人们可以便捷地获取大量信息，但也面临着信息过载问题。为了帮助用户在海量的数据中找到感兴趣的项目，个性化推荐系统应运而生。然而，大多数真实数据集中用户与推荐项目的交互相对稀少，导致模型无法获取足够信息以学习用户偏好，从而产生冷启动问题。

为此，已有不少研究工作在推荐中引入辅助信息，比如项目属性特征、用户社交信息等，以缓解冷启动问题。这些工作取得了不错的效果，但是仍存在以下不足：1) 所采用的建模方式不完善，没有充分挖掘出不同元路径形式内部信息；2) 信息融合方式单调简略，无法针对目标节点有效聚合元路径内部丰富的上下文信息。这两个问题都会导致用户和项目表征不完善。

针对上述问题，本文提出基于注意力机制的异质图网络推荐框架。基于图神经网络汇聚不同层次的信息，针对目标节点进行聚合操作，得到的用户项目嵌入表示与元路径上下文内容关系更密切，表征内容更丰富。最后在特征不同的两个真实数据集上进行实验验证和性能评估。本文贡献如下：

(1) 针对元路径信息利用不充分问题，加入连边信息，设计了基于关系旋转的用户项目所在元路径实例信息聚合方式。在保留元路径内部节点向量信息的同时，增加了对节点位置信息的利用。实验表明：相比基线模型 MCRec，准确率、召回率和归一化折损累计增益分别提升了 8.02%、5.05% 和 3.22%。

(2) 针对异质信息挖掘过程粗粒度问题，提出了基于注意力机制的用户项目所在不同元路径实例信息融合策略。采用多头注意力机制汇聚不同来源的元路径实例信息，能够克服异质图所带来的不稳定性。实验表明：相比基线模型 MCRec，准确率、召回率和归一化折损累计增益分别提升了 5.63%、2.72% 和 2.96%。

(3) 融合上述两点，在两个特征不同真实数据集上进行个性化推荐实验，实验表明：在数据分布较为平均的 LastFM 数据集上的准确率、召回率和归一化折损累计增益分别达到了 0.52、0.53 和 0.88，相比基线模型 MCRec 分别提升了 9.50%、6.09% 和 3.64%，在多数用户项目分布较为稀疏的 MovieLens 数据集上三个指标分别达到了 0.41、0.26 和 0.79，相比基线模型 MCRec 分别提升了 20.30%、21.17% 和 14.25%。冷启动对比实验也表明：随着数据稀疏性的增强，算法的提升效果也越明显，其中在最稀疏的数据集上三个指标分别提升了 24.94%、26.54% 和 19.82%。

本文工作有助于电商平台对用户精准推荐商品，具有一定实际意义。

图 14 幅，表 13 个，参考文献 42 篇。

关键词：个性化推荐；冷启动；图神经网络；异质信息网络；注意力机制

ABSTRACT

Thanks to the rapid development of the Internet, people can easily obtain a large amount of information, but they also face the problem of information overload. In order to help users find items of interest in massive amounts of data, personalized recommendation system came into existence. However, in most real data sets, the interaction between users and recommended items is relatively rare, which leads to the model dose not have enough information to learn user preferences, resulting in cold start problem.

To alleviate the cold start problem, existing research works have introduced auxiliary information in recommendation, such as item attribute characteristics, user social information, etc. These works have achieved good results, but the following shortcomings still exist: 1) The modeling method used is not perfect, and the internal information of different meta-path forms is not fully excavated; 2) The information fusion method is monotonous and simple, which cannot aggregate rich contextual information inside the meta-path effectively for the target node. Both of these problems make user and project characterization incomplete.

In response to the above problems, a heterogeneous graph network recommendation framework based on the attention mechanism is proposed. It gathers different levels of information based on the graph neural network, and aggregates information for the target node. By doing this, we can obtain richer user embedding and item embedding which have a closer relationship with the meta-path context content. Finally, experimental verification and performance evaluation are carried out on two real data sets with different characteristics. The contributions of this work are as follows:

(1) Aiming at the problem of insufficient utilization of meta-path information, information of edge is added, and an information aggregation method of meta-path instances where user projects are located based on relational rotation is designed. While retaining the node vector information within the meta-path, the utilization of node location information is increased. Experiments show that compared with the baseline model MCRec, the accuracy, recall, and normalized loss cumulative gain are increased by 8.02%, 5.05% and 3.22%, respectively.

(2) Aiming at the coarse-grained problem of heterogeneous information mining process, an attention mechanism-based information fusion strategy of different

meta-path instances where user items are located is proposed. Using a multi-head attention mechanism to gather meta-path instance information from different sources can overcome the instability caused by heterogeneous graphs. Experiments show that compared with the baseline model MCRec, the accuracy, recall and normalized loss cumulative gains are increased by 5.63%, 2.72% and 2.96%, respectively.

(3) Combining the above two points, a personalized recommendation experiment is carried out on two real data sets with different characteristics. The experiment shows that the accuracy, recall, and normalized loss cumulative gains on the LastFM data set with relatively even data distribution have reached 0.52, 0.53 and 0.88, which are 9.50%, 6.09% and 3.64% higher than the baseline model MCRec respectively. On the MovieLens dataset where most user projects are sparsely distributed, the three indicators reach 0.41, 0.26 and 0.79, respectively. Compared with the baseline Model MCRec increased by 20.30%, 21.17% and 14.25% respectively. The cold start comparative experiment also shows that with the increase of data sparsity, the improvement effect of the algorithm is more obvious. Among them, the three indicators on the sparsest data set have increased by 24.94%, 26.54% and 19.82% respectively.

The work of this article will help e-commerce platforms recommend more products which users are interested in, which has practical significance to a certain extent.

14 figures, 13 tables, and 42 references are contained in the dissertation.

KEYWORDS : Personalized Recommendation; Cold Start; Graph Neural Network; Heterogeneous Information network; Attention Mechanism

目录

摘要.....	iii
ABSTRACT.....	iv
1 引言.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 传统方法.....	2
1.2.2 结合深度学习的方法.....	3
1.3 研究主要内容及主要贡献.....	5
1.4 论文组织结构.....	6
2 技术背景.....	8
2.1 传统推荐算法.....	8
2.1.1 基于内容的推荐算法.....	8
2.1.2 基于协同过滤的推荐方法.....	8
2.2 异质信息网络相关概念.....	10
2.3 图嵌入技术.....	11
2.3.1 基于矩阵分解的图嵌入.....	12
2.3.2 基于随机游走的图嵌入.....	13
2.3.3 基于深度学习的图嵌入.....	15
2.4 图神经网络.....	16
2.4.1 图卷积神经网络.....	17
2.4.2 门控图神经网络.....	17
2.4.3 图注意力网络.....	18
2.5 开发平台.....	19
2.5.1 Python 编程语言.....	19
2.5.2 Anaconda 集成环境.....	19
2.5.3 Scikit-Learn 算法库.....	20
2.5.4 Pandas 数据分析库.....	20
2.5.5 Pytorch 框架.....	21

2.6 本章小结.....	21
3 整体设计与数据分析处理.....	22
3.1 任务场景.....	22
3.2 整体设计.....	22
3.3 数据集分析.....	24
3.3.1 数据集描述.....	24
3.3.2 数据集特征分析.....	25
3.3.3 数据预处理.....	28
3.4 本章小结.....	31
4 基于 HIN 的图神经网络社会推荐算法.....	33
4.1 基于 MAGNN 的个性化推荐算法.....	33
4.1.1 融合用户项目所在元路径实例信息.....	37
4.1.2 信息融合引入注意力机制.....	39
4.1.3 模型训练.....	41
4.2 评估指标.....	41
4.3 实验设置.....	43
4.4 实验结果分析.....	44
4.5 冷启动对比实验.....	47
4.6 本章小结.....	47
5 总结与展望.....	49
5.1 本文工作总结.....	49
5.2 未来工作展望.....	50
参考文献.....	51
作者简历及攻读硕士学位期间取得的研究成果.....	55
独创性声明.....	56
学位论文数据集.....	57

1 引言

1.1 研究背景及意义

在这个万物互联的时代，人们可以轻易获取大量信息，但随着数据量的增长扩大，用户也面临着信息过载的问题，在庞大的数据体系中获取自己想要的信息也是一个新的难题。为了解决此类难题，推荐系统应运而生，且随着技术不断发展，精准地针对用户进行个性化推荐也越来越普遍。

现在推荐系统中待处理的数据规模越来越大，相关记录动辄上百千万条，但是用户商品的实际交互行为却屈指可数，如此一来就会造成冷启动问题。Movielens 数据集因其具有丰富的数据特征经常被用来做推荐研究，其密度是 4.5%，Netflix 数据集是 1.2%，密度相对来说较大了，Delicious 数据集是 0.046%，在 DBLP 上得到的数据集在 0.02% 左右。而在实际的电商平台中，淘宝号称有 8 亿商品，平均每个用户浏览八百件的概率应该较低，那密度就在百万分之一甚至以下的量级了，可见在实际应用场景中由于数据稀疏造成的冷启动问题还是比较严重的。而这种稀疏性非常影响推荐性能，所以很有必要寻找一些解决方法利用有限的资源提供优质的服务。

为解决推荐系统中的冷启动问题，现有的研究方案大致可以分为两类。一类是通过引入附加信息丰富表征，比如用户的社交关系、项目的属性信息或者跨领域的公共知识等。在推荐系统的众多应用场景中，需要考虑的对象包括用户、项目以及相关属性等，并且这些对象通过各种复杂的关系相互影响，基于这种特殊的结构及特性可以将其建模为异质图网络。这种充分挖掘对象及其隐含关系的异质建模方式可以在一定程度上解决推荐系统所面临的冷启动问题^[1]。另一类是利用相关技术充分挖掘异质信息从而获得有效的用户和项目表征，如利用嵌入传播层网络、图卷积网络等。这些深度学习方法作为有利的辅助工具，可以帮助模型更好地利用附加信息学习用户偏好，从而丰富用户和项目的嵌入表示。

然而，现有的方案仍存在一些不足，在异质信息网络中，对异质信息的挖掘依赖于元路径，现有方案对用户项目所在不同形式的元路径信息利用不够充分。另一方面，对异质信息的融合策略较粗糙，未能有效利用元路径形式对用户和项目进行细粒度表征。因此，针对这些仍然存在的问题，需要考虑一种有效的学习框架来解决，进而提升推荐性能。

本文致力于研究基于异质信息的图神经网络社会推荐算法来改善冷启动问

题，从而实现个性化推荐性能提升。通过引入用户的社交信息以及项目的相关属性信息来丰富模型学习内容，并充分利用目标节点所在元路径的相关信息构建嵌入表示，同时利用图神经网络有效地学习用户和项目表征，旨在提升模型对用户偏好的学习能力，从而改善数据稀疏造成的冷启动问题。实现的任务场景是个性化列表推荐，通过对异质信息的细粒度学习提升了推荐性能，对冷启动问题有较好的改善能力。

1.2 国内外研究现状

为了解决推荐系统面临的数据稀疏问题，已经有大量工作致力于引入辅助信息构建异质信息网络，并利用相关策略挖掘用户项目之间的隐含语义关系从而提升推荐性能。下面将介绍一些传统解决方案以及结合深度学习的方法，并通过这些解决方案以及目前存在的问题来了解相关现状。

1.2.1 传统方法

由于异质信息网络的特殊结构特性，给推荐任务带来了不少挑战，已有不少学者为了解决这些挑战做出了大量研究，也提出了一些实用方法。

(1) 随机游走策略

在过去的十五年里，基于随机游走策略的推荐方法已经得到了广泛的研究，并且被大量应用于捕捉同构图以及异构图中节点之间的复杂关系^[2]。HERec^[3]基于异质信息网络中元路径的随机游生成节点序列以学习节点的相应嵌入表示，并且将其送入矩阵分解模型中最终完成商品推荐。Nandanwar 等人^[4]采用异质信息网络中顶点增强的随机游走策略，从而避免多个相邻且重要度较高的节点对推荐多样性的影响。HeteLearn^[5]是基于异质图中贝叶斯个性化排序的方法学习关系的权重，从而对用户的兴趣进行个性化建模。不过随机游走策略也有缺点，它需要在每个用户的每一步都生成所有候选项的排序分数，在大规模图上，效率低下的缺点就十分突出了。

(2) 基于分解的表示学习方法

针对不断扩大的网络规模以及数据的稀疏性所带来的问题，一种有效的解决方式是先将节点映射到低维向量空间，再应用于不同的数据挖掘任务中，即信息网络的表示学习^[6]。HERec^[3]利用元路径将异质图分割成多个同质网络，单独对每个同质网络进行学习，并根据需求采用 DeepWalk^[7]模型生成相应节点嵌入，最终再融合到一起。PTE^[8]利用文本信息构建异质信息网络，随后分割成 word-word、

word-document、word-label 这三个子网络，分别学习三个子网的节点嵌入表示。EOE^[9]将构建的学术异质信息网络分解成单词共现的网络和作者合作网络。在实际场景中，不同类型各个节点间的关系都可能隐含着丰富的语义信息，将原本完整异质图分割成若干同质图有可能造成信息损失。

异质信息网络中的节点以及节点之间含有丰富的语义信息，传统的学习方法无法很好捕捉其中的非线性关系，所以不能很好地抽取异质图中蕴含的复杂语义关系，无法充分挖掘异质信息。随着数据量的扩增，所构建的异质图规模也逐渐增大，这些浅层模型在大规模图上的处理能力是有限的，所产生的效率问题也会进一步影响推荐性能。

1.2.2 结合深度学习的方法

近几年深度学习技术有着蓬勃的发展趋势，引起了业界的广泛关注，已有大量研究学者将其应用到个性化推荐系统中，并且取得了不错的成效，因此将两者结合成为当前个性化推荐系统的研究热点之一。

(1) 异质信息结合深度神经网络方法

为了利用深度神经网络的优势，可以将不同节点及其特征信息构建为异质图嵌入表示。异构图嵌入旨在将异构图中的节点投影到低维向量空间中，针对这个问题已经有许多研究工作了。例如，metapath2vec^[10]生成的随机游走方式基于元路径，生成相应序列后送入到 skip-gram^[11]模型完成节点嵌入表示。ESim^[12]先根据用户定义相关元路径，再根据不同的正负样本实例随对应元路径来学习节点嵌入表示。HIN2vec^[13]可以对异质信息网络中节点和元路径的向量表示进行学习。模型PME^[14]通过将节点嵌入投影到相应的关系空间并优化投影节点之间的邻近度来学习节点嵌入。EPCDRec^[15]特点在于结合了两个领域的信息，一个是包含用户、电影以及电影属性的异构图，一个是包含用户、图书以及图书属性的异构图，通过公共属性连接两个领域信息构建包含用户、项目以及项目属性的异质图结构，根据不同属性设计不同的元路径形式。然后送入到嵌入传播层网络^[16]中，最终完成Top-N 的推荐任务。

MCRRec^[17]是近年被提出的基于异质信息网络的典型算法，利用 meta-path 提取 context 的信息，聚合用户、项目以及基于元路径的上下文信息进行推荐。为了丰富目标节点的嵌入表示，引入了项目的属性特征等作为辅助信息，采用随机游走挑选出合适的元路径实例，再采用卷积网络层对元路径实例内部信息进行聚合，接着采用最大池化操作生成该路径规则下的向量表示，接着采用注意力机制对不同的元路径信息进行聚合，得到 context 向量表示后与用户项目向量表示进行拼接，

再送入多层感知机得到相应的用户评分。

通过深度神经网络可以很好挖掘异质图中的相关信息，但是这些方案还存在一些需要解决的问题，主要集中以下几方面，一是节点本身的属性信息被忽略，当异质图中的节点内容特征较为丰富时不能有效处理；元路径信息利用不充分，在元路径内部，除了首尾节点以外的其他内部节点以及与之相连的边信息被丢弃，造成信息损失问题，从而影响推荐性能；信息融合策略粗糙，不同的元路径实例以及不同路径规则的元路径对目标节点的贡献值应该被区分，笼统地聚合不同来源的信息不利于全面刻画用户特征。

(2) 基于异质信息网络的社会推荐方法

通常可以利用社交信息进行两类推荐任务，一类是根据用户资料为其推荐可能感兴趣的新朋友，另一类是利用社交关系的潜在影响力为用户推荐感兴趣的项目。本文主要是针对第二类任务展开研究，这里介绍一些现有的研究方案。

HOSR^[18]致力于对社交网络中高阶邻居的影响力进行建模，进而完成项目推荐任务。考虑到不同用户拥有的朋友数量不均衡，对朋友数量较多的用户来说，与之直接相连的邻居信息已经足够，如果数量稀少就探索与之间接相连的高阶邻居信息。ESRF^[19]采用对抗学习框架为用户挑选出高质量的社交关系，并利用 motif 邻接矩阵对高阶社交信息进行表征。通过对候选邻居进行排序、降噪等得到较可靠的朋友关系，再利用这些信息学习用户偏好为其推荐项目。SAMN^[20]主要包括两个部分，基于注意力的记忆模块用来学习目标用户与朋友间的兴趣差异，基于朋友的注意力模块用来区分不同朋友的影响力，结合这两个模块完成项目推荐任务。GraphRec^[21]定义了用户社交图和用户项目交互图来刻画用户画像。其中用户社交图是同质图，社交影响力的传播会对推荐结果产生级联影响，了解用户的社会影响力在社交图上的传播有助于精准推荐。用户项目交互图是异质图，反映了用户对项目的购买或者评价关系。将用户作为桥梁连接两个图，利用图神经网络迭代聚合用户和项目的相关信息进行表征。

通过引入社交关系作为辅助信息有利于学习用户偏好，在一定程度上解决数据稀疏问题。但是现有的多数社会推荐算法受社交关系局限，利用的辅助信息较单一，忽略了项目本身的一些特征信息，而且当遇到没有社交关系的用户节点时，所构建的社交关系图就是不连通的，这样就无法充分挖掘用户项目间的隐藏信息。

(3) 异质信息结合图神经网络方法

伴随着图神经网络的兴起，不少研究学者将其与异质信息网络结合起来应用在推荐任务上，并且取得了不错的成效。HAN^[22]基于元路径表示，在原来异构图的基础上重构出同质图，同时采用图注意力机制汇聚邻近节点以及来自不同元路径的信息。PGCN^[23]认为用户在与项目序列交互过程中释放了丰富信息，从中构建

出用户-项目图、项目共现图以及用户-项目子序列图，在这三个子图上利用池化和卷积聚合邻居特征以学习用户和物品表示。**MEIRec**^[24]针对淘宝的实际业务需求，提出基于元路径的异质图神经网络，分别采用节点级别和语义级别的注意力机制学习意图推荐中的节点表征。**MCCF**^[25]初步探索用户购买动机，提出多成分图神经网络来基于不同动机分别聚合商品信息实现更细粒度的用户偏好建模，先利用分解器将用户商品交互拆分成多个可能导致购买行为的潜在组件，再利用合成器重新组合这些组件以获得用户和项目的嵌入表示，最终进行预测。以上方法主要是利用项目本身的一些属性信息以及用户项目之间的交互关系进行异质网络建模，没有考虑到其他有利于丰富用户项目表征的辅助信息，异质信息利用不充分。

上述基于异质信息网络的方法是借助附加信息来学习用户偏好，在一定程度上丰富了用户和项目表征。但是也存在一些仍需解决的问题，处在异质图中的节点包含丰富的内容及关系特征，对节点本身和所处元路径的信息挖掘不够充分，另一方面，针对目标节点，对异质信息的学习过程较为粗糙，因此探寻一种更有效的信息融合策略是有必要的。

1.3 研究主要内容及主要贡献

本文主要是针对个性化推荐系统中数据稀疏造成的冷启动问题，进行异质信息相关的图神经网络社会推荐算法研究，提出基于注意力机制的异质图网络推荐框架 **MAGNN-Rec**。为提升推荐性能，将相关数据建模为异质信息网络结构，同时利用图神经网络相关技术更好学习用户和项目表征。

本文的研究难点主要表现在以下几个方面：

(1) 汇聚高阶邻近节点信息。在将用户的社交信息以及项目相关属性信息建模为异质信息网络后，通常利用不同的元路径形式挖掘相关信息，针对目标节点，在一条元路径上与之直接或间接相连的邻近节点对其都有着一定的影响，需要充分利用这些邻近节点信息并将其汇聚到目标节点嵌入表示中。

(2) 细粒度学习信息融合过程。通常不同类型的节点对象会存在于不同的元路径中，针对每一类节点对象，每一种路径规则的元路径下可能有着多条具体的元路径实例。对目标节点来说，它可能存在于有着相同路径规则或者不同路径规则的元路径实例中，但凡存在，这些代表着不同具体含义的元路径信息对目标节点来说都是有意义的，所以需要寻求一种有效的信息融合策略来丰富用户和项目表征。

针对上述难点，本文的主要工作及贡献如下：

(1) 针对异质图中元路径信息利用不充分问题，加入边信息，设计了基于关

系旋转的用户项目所在元路径实例信息聚合方式。将通过元路径与目标节点直接或间接相连的每一个邻近节点信息和连边信息都融合到一起构成单一向量，同时保留了元路径内部节点的向量信息和位置信息，从而丰富节点嵌入表示。

(2) 针对不同的元路径实例引入注意力机制进行信息聚合。针对目标节点，每一种元路径下含有不同的元路径实例，且他们对目标节点的重要性是不同的，采用多头注意力机制汇聚不同来源的元路径实例信息，能够克服异质图所带来的不稳定性，同时有效聚合不同元路径内部实例信息。

(3) 在两个含有不同特征的数据集上进行 Top-N 列表推荐实验，相比基线模型，本文模型在各个评估指标上都有提升。其中，在数据分布较为平均的 LastFM 数据集上的准确率、召回率和归一化折损累计增益分别达到了 0.52、0.53 和 0.88，相比基线模型 MCRec 分别提升了 9.50%、6.09% 和 3.64%，在多数用户项目分布较为稀疏的 MovieLens 数据集上三个指标分别达到了 0.41、0.26 和 0.79，相比基线模型 MCRec 分别提升了 20.30%、21.17% 和 14.25%。并设计了冷启动对比实验，将数据集划分成不同稀疏度的子数据集，并与相关工作进行对比，实验结果表明，随着数据稀疏性的增强算法的提升效果越明显，其中在最稀疏的数据集上三个指标分别提升了 24.94%、26.54% 和 19.82%。

1.4 论文组织结构

本文的工作主要分为五章，相关内容安排如下：

第二章为技术背景，主要介绍了本文工作所涉及到的相关理论知识。首先通过相关传统推荐算法介绍了推荐的基本思路和关键计算方法。接着介绍了异质信息网络相关概念，通过举例展示了推荐系统中构建的异质图网络，并阐述了本文内容所涉及到的元路径以及元路径实例等概念。接着随图嵌入技术进行介绍，主要包括基于矩阵分解、基于深度学习和基于随机游走的图嵌入方法。随后阐述了图神经网络的相关知识，包括图卷积网络、门控神经网络以及图注意力网络。接着对研究工作中所使用的开发平台、框架以及一些技术工具进行介绍。这些内容为本文后续工作做了很好的铺垫。最后对本章内容进行总结。

第三章主要介绍了整体设计以及数据分析处理相关内容。首先介绍了本文的任务场景，为 Top-N 的个性化列表推荐。接着介绍了整体的设计框架，主要包括数据预处理、模型训练和实验验证这几个部分。随后对本文使用的数据集做了基本介绍，并分析了不同数据集的数据特征以及它们可能给实验结果带来的影响，随后阐述了数据集的预处理流程，主要包括构建邻接矩阵、生成元路径表示以及划分训练、验证、测试集这些过程。最后对本章内容进行总结。

第四章主要介绍了基于 HIN 的图神经网络社会推荐框架。先是介绍了现有推荐模型的基本情况，分析存在的问题，并给出了本文模型的整体架构。实现的内容主要包括两个方面，一个是对元路径内部实例信息进行编码，另一个是利用注意力机制进行相关信息融合。首先对异质节点信息进行线性转换将它们映射到同一个向量空间中去，接着采用关系旋转的元路径实例编码方式进行信息聚合，随后采用多头注意力机制聚合不同的元路径实例信息，并采用注意力机制融合不同的元路径信息。在完成对用户和项目的表征后设计优化目标函数训练模型，选取合适评估指标验证推荐性能，并设计相关实验观察模型的冷启动改善能力。最后对本章内容进行总结。

第五章概括总结了本文主要工作，并阐述了下一步研究方向。主要介绍了本文工作要解决的问题，并总结了本文的工作内容及主要成果，阐述本文工作仍存在的不足，同时展望未来可行的研究方向。

2 技术背景

本章主要介绍研究工作涉及到的相关技术知识，包括传统推荐算法、异质信息网络相关概念、图嵌入技术以及图神经网络相关知识，并介绍本文所使用的开发平台及环境。

2.1 传统推荐算法

经过不断地研究与探索，推荐系统表现出越来越好的性能，所使用的算法是在传统方法上面不断改进和完善的，所以本节先介绍传统的算法模式，并通过这些内容了解推荐的基本思路以及关键计算方法。传统的推荐方法主要包括基于内容的推荐算法和基于协同过滤的推荐算法。

2.1.1 基于内容的推荐算法

基于内容的推荐算法主要集中于项目本身的属性特征和用户的相关资料，寻找跟用户历史喜欢类似的项目作为推荐项^[26]。主要有三个步骤，首先提取项目的特征，然后根据用户的历史行为数据学习其偏好模型，接着将偏好模型和待选项目特征进行比较，从高到低对相似度排序，选出排名靠前的一组项目作为推荐项。这种方法的关键点在于计算不同项目特征向量之间的相似度，常用的度量方法为计算余弦相似度^[27]：

$$sim(i, j) = \frac{\sum_k w_{ik} w_{jk}}{\sqrt{\sum_k w_{ik}^2} \sqrt{\sum_k w_{jk}^2}} \quad (2-1)$$

其中 i 和 j 分别代表两种不同项目， k 代表项目 i 和 j 的共同属性 w_{ik} 和 w_{jk} 分别代表项目 i 和 j 在属性 k 上的取值，通过由高至低排列计算得到的相似度值，就得到了项目的推荐列表。

2.1.2 基于协同过滤的推荐方法

协同过滤推荐算法可以分为基于内存和基于模型的协同过滤推荐算法。

(1) 基于内存的协同过滤方法

基于内存的协同过滤算法按照实现角度的不同可以分为基于项目的协同过滤和基于用户的协同过滤这两种方法。基于项目的协同过滤方法主要思想是计算候

选项目与用户历史喜欢项目之间的相似度，选出较高的作为推荐项，通过对用户的历史行为进行分析来判断项目间的相似性，如果两个项目被较多的用户同时喜欢，则具有较高的相似度。同样基于用户的协同过滤方法是给目标用户推荐与其兴趣类似的用户所喜欢的项目，当不同用户对一些项目的评分多数较为接近时，可以判断这些用户的兴趣相似。基于项目和基于用户的方法计算过程类似，这里拿基于项目的协同过滤方法来举例，如果有历史评分记录，这里相似度常用皮尔逊相关系数来计算，假设 $r_{u,i}$ 为用户 u 对项目 i 的评分，对项目 i 和项目 j 都进行过评分的用户集合 $U_{ij} = U_i \cap U_j$ ，则两个项目之间的皮尔逊相关系数计算方法如下式：

$$\text{sim}(i, j) = \frac{\sum_{u \in U_{ij}} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U_{ij}} (r_{u,i} - \bar{r}_i)^2} \sum_{u \in U_{ij}} (r_{u,j} - \bar{r}_j)^2} \quad (2-2)$$

其中， $r_{u,i}$ ， \bar{r}_i 和 \bar{r}_j 所有用户对项目 i 和项目 j 的平均评分。

根据上式计算得到与待推荐项目相似度较高的邻居集合后，用户对目标项目的预测评分可以从这些集合的评分中得到，假设项目 t 的近邻集合为 S_t ，那么用户 u 对目标项目 t 的预测评分可以过下式计算得到：

$$r_{u,t} = \bar{r}_t + \frac{\sum_{i \in S_t} |\text{Sim}(t, i)| (r_{u,i} - \bar{r}_i)}{\sum_{i \in S_t} |\text{Sim}(t, i)|} \quad (2-3)$$

其中 \bar{r}_t 为所有用户对项目 t 的平均评分。

如果不存在用户项目之间评分记录而只有正反馈，也可以通过集合比较的方法来衡量项目间的相似度，如果两个项目拥有越多的共同交互用户，则说明他们之间的相似度越高。常用杰卡德相似度（Jaccard Similarity）和余弦相似度来计算：杰卡德相似度：

$$\text{Sim}(i, j) = \frac{|U(i) \cap U(j)|}{|U(i) \cup U(j)|} \quad (2-4)$$

余弦相似度：

$$\text{Sim}(i, j) = \frac{U(i) \cap U(j)}{\sqrt{|U(i)| \cdot |U(j)|}} \quad (2-5)$$

(2) 基于模型的协同过滤方法

基于模型的协同过滤算法重点在于对用户兴趣模型的建模，这种方法将用户和项目的相关资料作为训练数据送入模型训练参数，最终得到推荐结果。矩阵分解技术是比较经典的基于模型的协同过滤方法，它通过评分矩阵学习到用户和项目的低秩表示，并预测用户对某些项目的缺失评分。矩阵分解模型把用户和项目信息映射到一个维度为 K 的联合隐含语义空间中，每个物品 i 的对应向量为 q_i ，向

量中每个因子维度值代表该用户在该潜在因子上所占的权重大小，每个用户 u 对应向量为 p_u ，同样向量中每个值也代表用户的不同偏好程度。那么用户对项目的预测评分即可以通过向量间点积运算得到，即：

$$\hat{r}_{ui} = p_u^T \cdot q_i \quad (2-6)$$

假设用户对项目的真实评分为 r_{ui} ，则可以通过优化下式目标函数来学习模型中的参数值：

$$L(p_u, q_i) = \min_{p_u, q_i} \sum_{(u,i) \in \gamma} (\hat{r}_{ui} - r_{ui})^2 A = \pi r^2 \quad (2-7)$$

其中 (u,i) 表示用户项目对， γ 表示在评分矩阵中存在评分记录的用户项目对集合。如果评分矩阵中缺失值过多，可能引起推荐性能下降，则可以用下式改进损失函数作为优化目标：

$$L(p_u, q_i) = \min_{p_u, q_i} \sum_{(u,i) \in \gamma} (\hat{r}_{ui} - r_{ui})^2 + \lambda (\|p_u\|^2 + \|q_i\|^2) \quad (2-8)$$

其中， λ 为正则项系数。

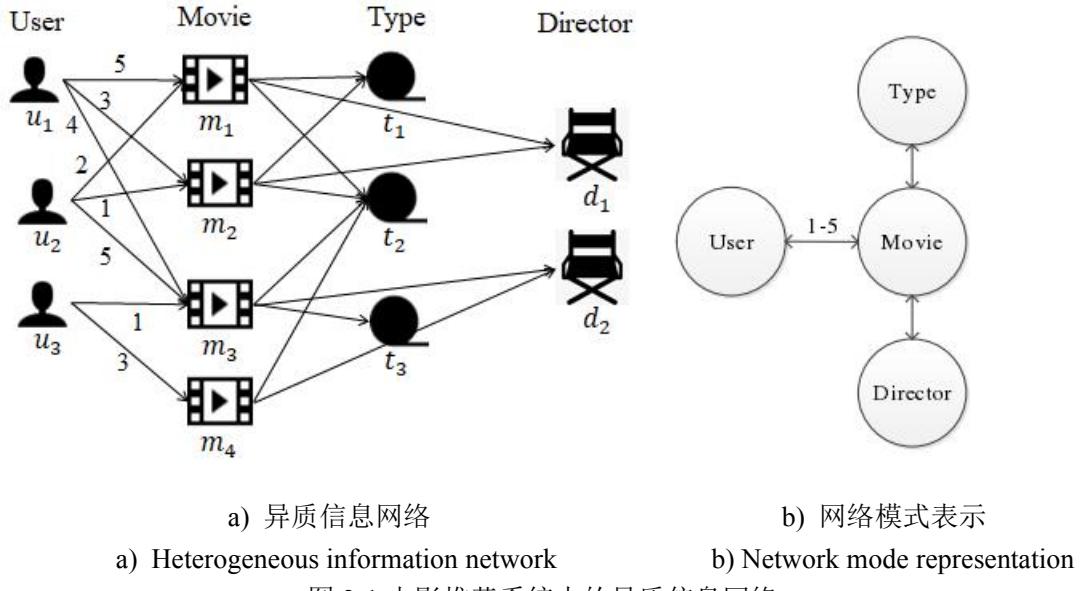
2.2 异质信息网络相关概念

在这个信息爆炸的大数据时代，海量数据中包含了许多不同种类的对象，对象之间也存在着形态各异的关联，要想充分利用这些信息可以将其建模为异质信息网络。放在推荐系统中，将用户、项目以及相关特征信息构建为异质信息网络可以更详细地刻画用户画像，从而实现更精准地个性化推荐。本节将介绍异质信息网络相关知识以及在推荐系统中的应用。

异质信息网络 (Heterogeneous Information Network) : 通常将信息网络定义为一个有向图 $G=(V, E, \varphi, \psi)$ ，其中，每个对象 $v \in V$ 属于对象集合类型 $A : \varphi(v) \in A$ 中的一个特定对象类型，每条链接 $e \in E$ 属于关系集合类型 $R : \varphi(e) \in R$ 中的一个特定关系类型。当其中的对象种类 $|A| > 1$ 或者关系种类 $|R| > 1$ 时，就称之为异质信息网络，如果不满足，就称之为同质网络。如图 2-1 a) 所示，在电影推荐系统中，包含有用户 (User)、电影 (Movie)、电影类型 (Type) 以及导演 (Director) 四种对象类型，同样的，不同种类的对象之间也对应着不同的关联类型。这种含有不同种类对象和关系的结构也可以抽象成网络模式表示，见图 2-1 b)。

元路径 (Meta-path) : 元路径是异质信息网络中比较重要的概念，以网络模式的方式定义，它能够表示不同类型网络结构中任意两个节点之间的不同路径类型。在网络模式 $T_G = (A, R)$ 中，元路径表示一条通过连边将对象关联起来的路径，记为 $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$ ，并将复合关系定义为 $R = R_1 \circ R_2 \circ \dots \circ R_l$ ，其中 \circ 表示关系合

成运算符, A_l 表示第 l 个对象类型, R_l 表示第 l 个关系类型。除了语义关系, 元路径也能够表示对象之间隐含的特征关系。仍以上述电影推荐系统为例, 元路径 $U \xrightarrow{\text{rate}} M \xrightarrow{\text{direct}} D \xrightarrow{\text{direct}^{-1}} M \xrightarrow{\text{rate}^{-1}} U$ 表示两个用户看过由相同导演执导的电影, 可以更全面地刻画用户特征^[36]。链接两个对象的不同元路径代表了不同的含义, 这样能得到包含丰富信息的特征表示。



基于元路径相关定义: 元路径实例即遵循某一种元路径模式的节点序列。如图 2-1 a)所示, $u_1 \rightarrow m_1 \rightarrow u_2$ 和 $u_2 \rightarrow m_2 \rightarrow u_3$ 就是遵循 $U \xrightarrow{\text{rate}} M \xrightarrow{\text{direct}^{-1}} U$ 这种路径规则的两个元路径实例。基于元路径的邻居表示通过某一种元路径模式与目标节点相连的邻居。比如, $u_1 \rightarrow m_2 \rightarrow u_2$ 和 $u_1 \rightarrow m_3 \rightarrow u_3$ 也遵循 $U \xrightarrow{\text{rate}} M \xrightarrow{\text{direct}^{-1}} U$ 的路径规则, 对目标节点 u_1 来说, u_2 和 u_3 就是基于这种模式所获得的两个邻居。基于元路径的图即由原始图上基于某一种元路径模式的邻居节点对组成。实质上是在原始元路径的基础上将中间节点去掉, 只留下首尾节点, 同时在新路径上用连边将两端节点关联起来, 通过在对象间建立起联系构成基于元路径的图。

2.3 图嵌入技术

图嵌入即将图中的节点、连边映射到低维的向量空间中, 实质上是把具有高维度、复杂、动态的海量数据映射到低维、统一、稠密的向量空间中, 旨在提供一种计算复杂度更低的方法, 从而便于应用到下游的节点分类、链接预测以及推荐任务中。

针对不同的应用场景，人们提出了不同的图嵌入方法和模型，主要可以分为基于矩阵分解的嵌入方法、基于随机游走的嵌入方法以及基于深度学习的嵌入方法。

2.3.1 基于矩阵分解的图嵌入

基于矩阵分解的嵌入方法主要思想是通过矩阵表示图中节点之间的关联信息，为每一个节点生成相应的向量表示，将图中结构映射到低维的向量空间中。通常用来表示图结构的有拉普拉斯矩阵和邻接矩阵，在实际应用中，根据不同矩阵的性质可以选择不同的矩阵分解方法，主要有基于拉普拉斯特特征映射以及基于相似矩阵这两种分解方法。

(1) 基于拉普拉斯特特征映射的分解方法

基于拉普拉斯特特征映射的方法主要思想是利用节点对之间的相似性来保持图的原本结构特性，如果相似的节点对被映射到特征空间中的距离较远，那么就给该节点对较大的惩罚值。经典的基于拉普拉斯特特征映射的学习方法有 LE^[28]、CGE^[29]（Cauchy Graph Embedding）、Isomap^[30]等。假设图结构用 $\mathcal{G}=(V, E)$ 来表示，其中 $V = (v_1, v_2 \dots v_n)$ 为图 G 中节点的集合， $E = \{e_{i,j} | \forall i, j \in [n], i \neq j\}$ 为图 G 中所有连边的集合， $W_{i,j} \geq 0$ （当节点 v_i 和节点 v_j 没有直接连边时， $W_{i,j} = 0$ ）表示 G 的邻接矩阵， Y_i 表示节点 v_i 的嵌入向量。基于上述思想，可以采用下式作为目标函数来完成低维映射，即：

$$Y = \sum_{i \neq j} (Y_i - Y_j)^2 W_{ij} = 2Y^T LY \quad (2-9)$$

其中， L 为图 G 的拉普拉斯矩阵， $L = D - W$ ， D 表示对角矩阵， $D = \sum_j W_{ij}$ 。

(2) 基于节点相似性的分解方法

基于节点相似性的分解方法主要是通过矩阵分解渐进地模拟原图中的相似节点，从而保持节点间的相似性。比如 LLE^[31]、GraRep^[32]（Graph Representation）以及 HOPE^[33]（High-Order Proximity Preserved Embedding），这些图嵌入方法都是基于节点相似性进行矩阵分解的。假设在图 $\mathcal{G}=(V, E)$ 中， Y 表示图中的原嵌入向量矩阵，要得到最终嵌入向量可以用公式(2-10)作为目标函数进行优化：

$$Y^* = \min(Y - Y_s Y_t^T) \quad (2-10)$$

其中 Y_s 和 Y_t 分别表示源空间和目标空间中节点的嵌入向量。

2.3.2 基于随机游走的图嵌入

基于随机游走的图嵌入方法是通过不同的游走策略获取图的局部或者全局结构，从而生成相应节点序列，然后通过 Skip-Gram 算法实现节点在低维空间的嵌入。这种方法是由自然语言处理中的 Word2vec^[1]模型演变而来的，能够很好捕捉图中各个节点之间的关系，比较适用于局部特征明显而整体结构不清晰的大规模图。

随机游走过程可以看成是一个游走者以一定的概率在图上做随机运动，游走者按照一定的概率选择下一个目标节点，再以这个节点作为起点选择下一个临近节点，由此循环往复，多次游走后所经过的节点构成随机游走序列。在推荐系统中，可以将用户和项目看作图中的节点，用户项目间的交互行为构成节点间的连边，假如存在 m 个用户和 n 个项目，用户游走到当前节点的概率由公式(2-11)表示：

$$P_{ij} = P(X_{u,d+1} = j | X_{u,d} = i) \quad (2-11)$$

其中， $P_{ij} \in \mathbf{P}$ ，那么 \mathbf{P} 表示 $m \times n$ 的转移概率矩阵。 $X_{u,d} = i$ 表示用户先进行 d 步随机游走，随后停留在项目 i 处的一个随机变量。当用户游走到某一个节点后，在当前节点的基础上继续游走的概率为 α ，回到初始节点从头开始随机游走的概率是 $1 - \alpha$ 。用户节点 u 和项目节点 i 之间的初始权重 $r_{u,i}$ 由用户项目的历史交互行为决定，将其归一化后得到 $\vec{r}_{u,i}$ ，用户对项目的初始概率分布由式(2-12)表示：

$$P(X_{u,0} = i) = \vec{r}_{u,i} \quad (2-12)$$

则用户节点 u 游走 d 步后到达项目 j 处的概率由式(2-13)表示：

$$P(X_{u,d} = j) = \alpha \sum_{i=1}^n P(X_{u,d-1} = i) P_{ij} = \alpha^d \sum_{i=1}^n \vec{r}_{u,i} P_{ij}^d = \alpha^d \vec{r}_{u,-} \vec{P}_{-j}^d \quad (2-13)$$

用户 u 在项目节点 j 处的总概率表示项目 j 对用户 u 的推荐值，如式(2-14)所示：

$$P(X_u = j) = \frac{\sum_{d=1}^{\infty} P(X_{u,d} = j)}{\sum_{d=1}^{\infty} \sum_{i=1}^n P(X_{u,d} = i)} = c \sum_{d=1}^{\infty} \alpha^d \vec{r}_{u,-} \vec{P}_{-j}^d \quad (2-14)$$

其中， c 对具体用户来说是常量。对于用户 u ，可以通过随机游走得到不同项目的推荐值，表示用户对它们的偏好程度，进行排序后可以选出前 N 个项目进行推荐。

由于同质图和异质图有着不同的结构，通常采用不同的基于随机游走的嵌入表示方法。

(1) 同构图中的随机游走

深度游走 (Deep Walk)^[7] 基于 Word2vec 延伸出的一种图嵌入方法，首先针对目标节点随机地生成邻居节点，得到一定长度的随机游走序列，然后通过 Skip-Gram 模型将生成的随机游走序列映射到低维空间中构成相应嵌入表示。它的

目标损失函数定义如下：

$$\max_f \sum_{u \in V} \log(Pr(N(u) | f(u))) \quad (2-15)$$

其中， $f(u)$ 就是当前节点， $N(u)$ 为邻居节点。这种方法自适应性比较强，当新增节点时不需要让已有的节点重新学习，而且属于同类型节点的向量表示是相似的。但是这种方法不适合有权重的图结构，而且有限长度的随机游走序列影响了全局特征的完整性，在大规模图上，不利于调整相关参数。

节点-向量（Node2vec）^[34]模型同样借鉴了Word2vec模型，有广度优先游走（Breadth-First Search, BFS）和深度优先游走（Depth-First Search, DFS）两种方式，通过调整这两种方法的参数来获得图的局部或者整体结构。如图 2-2 所示，节点序列 (u, s_4, s_5, s_6) 为深度优先游走的结果，节点序列 (u, s_1, s_2, s_3) 为广度优先游走的结果。其主要过程为先计算转移概率，接着利用 BFS 和 DFS 生成节点序列，然后利用 Skip-Gram 模型生成低维嵌入向量。

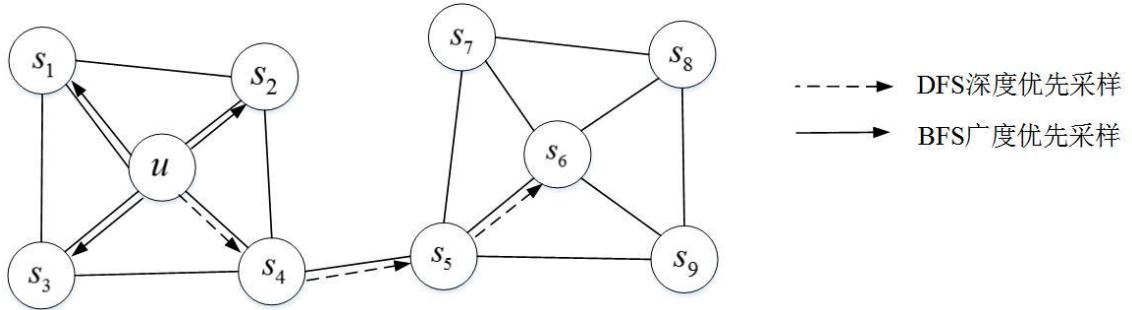


图 2-2 Node2vec 模型中两种采样方式
Figure 2-2 Two sampling methods in Node2vec model

与 Deep Walk 类似，其目标损失函数定义如公式(2-16)所示：

$$\max_f \sum_{u \in V} \log(Pr(N_s(u) | f(u))) \quad (2-16)$$

其中 $f(u)$ 代表当前节点， $N_s(u)$ 表示通过采样策略 s 获取的邻居节点。如果假设每个邻居被采样的概率是独立的，那么计算所有邻居被采样的概率只需要将采样每个邻居的概率相乘即可，如式(2-17)所示：

$$P_r(n_i | f(u)) = \prod_{n_i \in N_s(u)} Pr(n_i | f(u)) \quad (2-17)$$

如果再假设特性空间是对称的，那么对于空间中由一条边相连的节点 A 和节点 B 来说，它们对彼此的影响是相同的，基于此项假设可以得到下式：

$$P_r(n_i | f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))} \quad (2-18)$$

基于上述三个公式可以得到最终优化结果，即：

$$\max_f \sum_{u \in V} [-\log Z_u + \sum_{n_i \in N_s(u)} f(n_i) \cdot f(u)] \quad (2-19)$$

(2) 异构图中的随机游走

元路径-向量模型^[10] (Metapath2vec) 适用于异构图中，主要针对元路径进行相关随机游走。其主要过程是先将图中的节点通过一定的语义关联构建为元路径形式，接着根据元路径中的节点语义关系生成游走序列，然后将生成的序列映射成节点嵌入向量表示。比如一个 meta-path 的模式定义为 $V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} V_{l+1}$ ，具体含义已经在 2.2 节元路径中阐述过。那么给定一条元路径 P，第 i 步的转移概率就可以定义如下：

$$p(v^{i+1} | v_l^i, P) = \begin{cases} \frac{1}{N_{l+1}(v_l^i)} & (v^{i+1}, v_l^i) \in E, \phi(v^{i+1} = l+1) \\ 0 & (v^{i+1}, v_l^i) \in E, \phi(v^{i+1} \neq l+1) \\ 0 & (v^{i+1}, v_l^i) \notin E \end{cases} \quad (2-20)$$

其中， $v_l^i \in V_l$ ， $N_{l+1}(v_l^i)$ 表示节点 v_l^i 的 V_{l+1} 类型的邻居节点集合。第一行等式可以理解为两点之间有边，且下一个节点属于定义好的元路径上的下一类型集合，第二行等式可以理解为两点之间有边，但是下一个节点不属于定义好的元路径上的下一类型集合，第三行表示两个节点之间没有边。其目标损失函数定义如下：

$$\arg \max_{\theta} \sum_{v \in V} \sum_{l \in L_v} \sum_{c_l \in N_l(v)} \log p(c_l | v; \theta) \quad (2-21)$$

其中，通常将 $p(c_l | v; \theta)$ 定义为一个 softmax 函数，即：

$$p(c_l | v; \theta) = \frac{e^{X_{c_l} \cdot X_v}}{\sum_{u \in V} e^{X_u \cdot X_v}} \quad (2-22)$$

2.3.3 基于深度学习的图嵌入

随着对图嵌入技术的研究越来越深入，人们逐渐将深度学习相关知识结合起来以改进浅层模型，所构成的深层模型能够捕捉高度非线性的结构，从而实现最优解的生成。这种深层模型通常是以编码器-译码器作为框架结构的，编码器的功能是实现图中节点到低维嵌入向量的转换，译码器则是将低维向量重构成图结构，具体表示如下：

$$ENC : V \rightarrow R^d; DEC : R^d \times R^d \rightarrow R^+ \quad (2-23)$$

$$DEC(ENC(v_i), ENC(v_j)) = DEC(Y_i, Y_j) \approx s(v_i, v_j) \quad (2-24)$$

$$\mathcal{L} = \sum_{v_i, v_j \in V} \ell(DEC(Y_i, Y_j), s(v_i, v_j)) \quad (2-25)$$

其中，ENC 表示编码函数，将节点用嵌入向量表示，DEC 表示译码函数，重构节点之间的相似性， $s(v_i, v_j)$ 表示节点 v_i 和节点 v_j 之间相似性， ℓ 表示优化模型的损失函数，针对节点之间的预测相似度和真实值，目标在于尽可能最小化两者差的绝对值。目前典型的基于深度学习的图嵌入模型有下面几种。

结构化深度网络嵌入^[35]（Structural Deep Network Embedding, SDNE）是一种基于深度自动编码器的方法，它能够捕捉图中一阶乃至二阶的相似性，通过高度非线性的函数以及优化目标函数得到最终的嵌入向量表示。主要过程为先获取 K 层的隐嵌入向量表示 $y_i^{(k)}$ ，以及输出的数据表示 \hat{x}_i ，接着计算损失函数，然后通过随机梯度下降法来优化整个模型。

基于深度神经网络的图表示学习^[36]（Deep Neural Networks for Learning Graph Representation, DNGR）是一种结合了深度自动编码器以及随机漫游的方法，由于传统的方法大多为线性降维，这样就无法保持图结构的非线性，这种方法就能够很好解决这个问题。其主要过程为先利用随机漫游算法生成概率共现矩阵（Probability Co-Occurrence Matrix, PCO Matrix），再通过此矩阵计算正点互信息矩阵（Positive Point-wise Mutual Information Matrix, PPMI Matrix），然后通过栈式降噪编码器生成图嵌入表示，该向量表示包含了高阶相关性。这种方法能够捕捉复杂的图结构以及非线性结构特征，而且采用正点互信息矩阵能够避免嵌入大量的无效节点。

基于图采样和聚合的图嵌入^[37]（Graph Sample Aggregate, GraphSAGE）是一种利用节点本身特征信息的方法，对于新增的节点或者子图，采用改进的图卷积网络生成嵌入表示。这种方法能够实现增量节点的嵌入更新，同时保持图的结构和特征信息。其主要过程为先对图中每个节点的邻近节点进行采样，采样深度为 k，然后利用聚类函数聚合图中每个节点的邻近节点特征信息，再送入非线性激活函数中生成相应嵌入表示。

2.4 图神经网络

近年来，人们逐渐尝试将深度学习技术应用在图结构上，得益于卷积网络、循环网络以及深度自动编码器等相关技术的启发，研究人员定义和设计了一种特殊的神经网络结构来处理图数据—图神经网络（Graph Neural Networks, GNN）。图神经网络在分析和处理非欧几里德数据时有强大的能力，通常它的处理流程为先进行节点预表示，即利用图嵌入方法生成每一个节点的嵌入表示，然后进行图

节点采样，然后根据一定规律提取节点的临近节点构成子图，再提取这些子图的局部或者整体特征，最终生成图神经网络并进行训练。接下来介绍一些常见的图神经网络方法。

2.4.1 图卷积神经网络

图卷积网络算是图神经网络研究中比较成熟的方法了，它将原始的图结构数据 $\mathcal{G}=(V, E)$ 映射到新的特征空间 $f^G \rightarrow f^*$ 。以单层的向前传播图卷积网络为例，用 w_i 表示第 i 层神经网络的特征，在计算图中每一个节点时，第 $l+1$ 层神经网络的输出可以用下式表示：

$$H^{(l+1)} = f(H^l, A) \quad (2-26)$$

其中， $f(\cdot, \cdot)$ 表示非线性激活函数， A 表示特征的邻接矩阵。若以 $\text{ReLU} = \sigma(\cdot)$ 作为非线性激活函数，则得到的图卷积神经网络的分层传播规则可以用下式表示：

$$f(H^l, A) = \sigma(\hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} H^{(l)} W^{(l)}) \quad (2-27)$$

其中， $\hat{A} = A + I$ 表示图的邻接矩阵， I 表示单位矩阵， $\hat{D} = \sum_j \hat{A}_{ij}$ 表示矩阵 \hat{A} 的对角矩阵， $W^{(l)}$ 表示第 l 层神经网络的权重参数矩阵。

通过上述算法流程可以看出 GCN 实质是汇聚节点的邻近节点特征信息，通过对图上各个节点不断进行迭代训练，在特征向量空间中将相似性的节点聚合在一起。这种传播规则使得卷积神经网络的局部特性融入到了整体结构中，随着层数的增加每一个节点汇聚的特征信息也越来越丰富。

图卷积神经网络方法主要分为基于频域和基于空间两种模式。基于频域的图卷积神经网络可以用来处理无权重的无向图，将输入的图结构定义为可以进行特征分解的拉普拉斯矩阵。Bruna 等人^[38]最早提出了基于频域的图卷积神经网络，定义了卷积层函数，将图卷积神经网络定义为是图信号处理中的滤波器，从而进行特征分解。基于空间的图卷积网络主要思想是计算中心节点与邻近节点的卷积，通过卷积核的移动汇聚图上各个节点的特征信息。随机稳态图神经网络迭代算法^[39]是一种基于空间的图卷积算法，对于拥有不同邻近节点数量的节点采用了双向权重矩阵，从而在卷积迭代运算中对及节点特征信息进行不断更新。

2.4.2 门控图神经网络

门控图神经网络的特点在于在图神经网络的基础上采用了门控循环单元（Gated Recurrent Unit, GRU）。在进行不同轮次的迭代更新时，节点会有不同的

状态信息，采用 GRU 能够考虑到它们之间的关联。假设节点 i 的初始状态信息为 $h_i^{(l)} \in R^d$ ，节点 i 的邻居集合为 N_i ， A_{ij} 为传播矩阵，表示节点 i 到 j 的传播规则。在迭代轮次 t 上，节点 i 聚合了邻近节点信息得到了上下文交互信息 $m_i^{(t)} \in R^d$ ，那么节点 i 的隐藏向量表示与 $h_i^{(t)}$ 与前一轮次的向量表示 $h_i^{(t-1)}$ 之间的时序关系如下所示：

$$m_i^{(t)} = \sum_{j \in N_i} A_{ij} \cdot h_j^{(t-1)} \quad (2-28)$$

$$h_i^{(t)} = GRU(h_i^{(t-1)}, m_i^{(t)}) \quad (2-29)$$

目前也有不少基于门控机制的神经网络方法研究，比如门控图神经网络^[40]（Gated Graph Sequence Neural Networks, GGNN），这种方法是基于门控循环单元的，通过控制在网络传播过程中固定步数 T 的迭代循环来实现网络结构。首先汇聚节点 v 的邻近节点特征信息，再通过门控循环单元对每个节点隐藏状态进行迭代更新。每进行一次更新，每个节点都会收到邻近节点的信息，也会发送信息给邻近节点。

2.4.3 图注意力网络

图注意力网络的特点在于神经网络中的注意机制，这种方式可以看作是用注意力模块代替了常见卷积神经网络中的卷积激活器，也有不少研究将其与门控机制结合起来控制注意力权重参数，从而达到更好的效果。不同于卷积神经网络中的固化网络分层传播过程，图注意力网络利用邻近节点不断地累积求和过程实现信息聚合，不同的节点或者子图因有着不同的重要程度而享有不同的注意力权重。在图注意力网络中，可以用 $\alpha_{ij}^{(l)}$ 表示第 l 层的注意力权重，带注意力权重的节点可用式(2-30)表示：

$$z_i^{(l)} = W^{(l)} \bar{h}_i \quad (2-30)$$

其中， \bar{h}_i 表示原始的节点向量， $W^{(l)}$ 为可变的线性变换参数。这样根据不同的注意力权重，节点间的注意力分数会随着层数进行迭代更新。

近年来，人们也开始研究将注意力机制应用在异构图网络结构中。由于异质信息网络节点以及边的复杂特性，人们也面临着更大的挑战。MCRec (Leveraging Meta-path based Context for Top-N Recommendation with A Neural Co-Attention Model)^[17] 提出了一种在异构图上利用共同注意力机制来融合相关信息的方法，不仅考虑了用户和项目本身的内容特征，也考虑了连接两者的元路径上下文信息，最终将三者向量拼接到一起，但是针对不同的元路径实例信息以及不同路径规则

的元路径信息融合过程较为粗糙。

2.5 开发平台

本节主要介绍本文所使用的开发平台、框架以及所使用到的工具等相关内容。本文的算法以及数据预处理模块都用 Python 语言实现，为了能够使代码高效运行采用了 Anaconda 集成环境，可以在里面调用丰富的库并且自由切换不同环境，本文还用到了 Pytorch 深度学习框架，在进行数据处理时使用了相关科学计算库，这里就 Scikit-Learn 算法库以及 Pandas 数据分析库进行了介绍。

2.5.1 Python 编程语言

Python 是一种面向对象的可解释计算机编程语言，能够兼容 Window、Linux 以及 Mac 系统，不同平台上的编写差异不是很大，方便跨平台运行。其可解释性也很强，编写的代码易于理解，也方便开发人员分开协作。

此外，Python 语言在众多领域都有着普遍的应用，遍及大数据、人工智能、云计算、web 开发、系统运维以及游戏开发等。应用范围如此之广，得益于它庞大的标准库以及第三方库，能够处理各种领域的工作，包括且不限于正则表示、web 应用、数据库、图形用户界面、单元线程等。鉴于这些库的种类繁复且体系庞大，利用相关工具对其进行统一管理和维护使用起来就比较方便了。

2.5.2 Anaconda 集成环境

为了方便使用 Python 进行实验探究，可以使用 Anaconda 对相应库及环境进行统一管理。Anaconda 里面包含了常见的 Python 库及依赖包，针对环境依赖问题配置了 conda 管理系统，包管理的使用类似于 Python 中的 pip 命令，根据实际需求可以在不同环境中配置不同版本的 Python，可以对这些环境自由切换，并且无需配置环境变量。

Anaconda 有三个主要部件，Anaconda Prompt 为其自带的终端，可以使用命令配置相关变量，在自由切换不同环境的同时利用 pip 安装需要的特定包。Jupyter Notebook 是一种网页版的编辑器，能够实现在线运行调试，通常在进行数据预处理时会使用到，它能够实时显示输出。Spyder 是桌面版的编辑器，当代码量较大时可以使用它进行编译调试，跟 Pycharm 软件功能类似。

2.5.3 Scikit-Learn 算法库

Scikit-Learn 是针对 Python 语言的开源机器学习库。Scikit-Learn 含有丰富的机器学习算法，在相关数据集上能够进行强有力地信息挖掘。它支持分类、聚类、回归和降维这几类算法实现，主要能够实现的功能有数据处理、特征提取以及模型评估等。

Scikit-Learn 构建在 Numpy、Scipy、Matplotlib 这三大库之上。

其中，Numpy 库作为数据分析基础工具，是一个科学计算库，能够对数组以及矢量进行高效运算。Numpy 于 2005 年由 Travis Oliphant 创建，开源后被广泛使用，它本身多数是用 C 语言开发的，这就使得计算效率很高。通常会用它处理多维数组，提供了丰富的方法能对其进行排序、选择、统计等相关操作。数组对象表示为 ndarray，相比 Python 中自带的 List 数据结构，它的数据被连续存储在内存中，使得进程能够高效访问到，所以它的处理速度更快、效率更高。

Scipy 在 Numpy 库的基础上增加了许多科学计算库函数，例如线性代数、图像处理、常微分方程数值求解、信号处理、稀疏矩阵处理等。Scipy 所使用的数据结构是由 Numpy 提供的多维数组，覆盖了不同科学计算领域，比如常用的 scipy.sparse 用来处理大型稀疏矩阵，scipy.io 用来处理数据输入和输出等。

Matplotlib 是一个便捷的数据可视化工具，通常配合 Numpy 库使用，对运算结果进行可视化显示。可视化显示通常使用 plot 工具包，里面封装了绘图对象结构，类似于 Matlab 里面的绘图 API，调用相关函数，只需少量的代码就可以绘制高质量的直方图、柱状图、散点图或者二维、三维图像。里面提供丰富的函数，所以满足各种复杂图像的样式，也能够调整部分图像覆盖区域的颜色，同时也能够进行字体调整。

2.5.4 Pandas 数据分析库

Pandas 是基于 Numpy 的一种工具，主要是用来处理数据分析的一些任务。Pandas 在 2009 年开源出来被大家广泛使用，起初被用来分析相关金融数据，所以能够对含有时间属性的序列进行有效分析。

Pandas 是 Python 的数据分析包，内部含有丰富的库和相关数据模型，能够对大规模数据集进行高效处理。作为 Python 的一个库，它能够处理 Python 中的数据类型，但同时也有自己定义的一些数据结构，包括 Series、DataFrame、Panel 等。其中，Series 表示一维数组结构，与一维结构的 Array、Python 中的 List 数据类型都比较相似，但不同点在于 List 中可以存放不同的数据类型，而 Series 和 Array

中存储的数据类型需一致，存放单一类型的数据有利于进行内存管理，提高运算效率。DataFrame 定义为二维表格数据结构，可以存放 Series 类型数据。Panel 表示三维数组，可以存放 DataFrame 类型数据。

2.5.5 Pytorch 框架

Pytorch 是一款基于 Python 的框架，具有简洁高效的特点，在深度学习领域应用广泛。Pytorch 于 2017 年被首次开源，自此该框架被越来越多的人所熟知。追溯其诞生来源，Torch 于 2002 年使用 Lua 语言作为接口，由于该语言并没有被广泛使用，对 Lua 不太熟悉的人要想使用该框架还要现学这门语言，由此一来使用 Torch 的门槛就提高了。为了提高 Torch 的普遍适用性，开发人员决定使用比较受欢迎的 Python 作为接口，于是在 2017 年推出了 Pytorch。

Pytorch 受到越来越多的追捧，是因为其具有众多的优点。在具有较高灵活性的同时，它的速度也毫不逊色，可以满足高度复杂网络中的计算要求。同时，在官网中对于框架的使用有完整的文档，方便查询理解。它支持三种安装方式，分别为 conda、pip 和 source，本文是结合 Anaconda 环境安装的。

2.6 本章小结

本章主要是阐述了本文工作涉及到的相关知识。

首先介绍了传统推荐系统算法，主要包括基于内容的推荐算法和基于协同过滤的推荐算法，通过这些算法介绍阐述了推荐系统的一般流程以及关键计算步骤。接着介绍了异质信息网络相关知识，阐述了异质网络、元路径的基本定义以及本文要涉及到的基于元路径的相关概念。

在建立异构图的基础上，关键的一步是将图中节点的相关信息转换成嵌入表示，所以接着介绍了图嵌入技术。其中包括基于矩阵分解的嵌入方法、基于随机游走的嵌入方法以及基于深度学习的嵌入方法，并对不同方法所涉及到的关键步骤作了说明。

本文的研究工作涉及到图神经网络，介绍了图神经网络相关概念及常见的扩展结构，包括图卷积网络、图、门控图神经网络以及注意力网络。最后对本文实验所使用的编程语言、开发平台、框架以及相关数据分析库等做了相关介绍。

3 整体设计与数据分析处理

本章介绍要实现的任务场景以及构建基于注意力的异质图网络推荐算法的主要流程，并对所使用数据集的基本情况以及主要预处理过程进行介绍。在进行整体框架的描述时，首先介绍需要考虑的问题以及解决方案，然后给出整体流程，并对每个阶段做相关说明。在数据分析处理模块，先分析了不同数据集特征，然后对预处理过程进行描述。

3.1 任务场景

推荐的主要任务是根据用户的历史行为以及相关信息预测他们的偏好，从而进行商品推荐。在推荐系统中有两大任务场景，为评分预测和 Top-N 列表推荐。评分预测通常需要用户对商品的一些显示反馈数据，比如说用户对电影的具体评分值，通过模型训练，使得模型对项目的预测评分尽可能接近真实值。而 Top-N 列表推荐只需要一些隐式反馈数据就好了，比如用户的浏览、购买记录等，通过训练模型学习用户偏好，针对每个用户生成长度为 N 的个性化推荐列表。

由于在实际场景中，多数情况下无法拿到用户对项目的显示反馈数据，但是像浏览记录这样的隐式反馈数据比较容易获得，那么相比之下 Top-N 列表推荐就更有实际意义。因此，本文将针对 Top-N 列表推荐任务场景进行研究。

Top-N 列表推荐通常用于购物网站或者音乐推荐这样的实际场景中，通过用户的购买记录、对歌曲的收听记录等来学习用户偏好，为他们推荐可能感兴趣的项目列表。简单情况下的用户项目邻接矩阵中元素值为 1 或 0，分别表示已有过历史交互的用户项目对和没有交互过的用户项目对，通常将其中的项目称作正向反馈和负向反馈项目。通过最大化对正向反馈和负向反馈项目的预测评分之差，对模型进行训练优化，得到项目排序列表。

3.2 整体设计

实现推荐框架的关键步骤在于充分挖掘跟用户项目相关的潜在信息，并将这些隐藏信息融入到用户和项目的嵌入向量表示中，再对模型进行训练不断优化目标函数。因此，在设计基于异质信息的图神经网络推荐算法架构时需要考虑以下几个问题：

- (1) 充分利用除了交互信息以外的用户和项目相关信息。在现实生活中，除

了常见的用户对项目的评分记录或者消费记录，项目的其他属性信息也很丰富，比如电影的类型、执导导演、参与的演员等等，由这些属性关联起来的电影确实更有可能被用户青睐。已有大量研究表明，用户的朋友对用户在选择某些商品时也有很大的影响力。所以首先需要将用户、项目、相关属性对象以及它们之间不同的关系类型建模为异质信息网络，并将这些含有不同特征属性的节点信息统一映射到相同隐含语义空间。

(2) 完成节点在异质图上的嵌入表示。构建好的异质信息网络上，节点会通过不同类型的关系与其他节点产生关联，将这些作为辅助信息融入到目标节点的嵌入表示中是至关重要的。首先可以预设具有一定意义的元路径形式，然后利用图中的注意力机制融合元路径内邻近节点对目标节点的贡献信息，再采用语义注意力区分目标节点所在不同元路径带来的不同贡献值，从而得到节点的嵌入表示既包含所在元路径所有邻近节点信息又综合了每一条所在元路径的贡献信息。

(3) 设置优化目标函数完成模型训练。在得到用户和项目节点的嵌入向量表示后，采用内积运算得到预测评分。在推荐中，对某一用户而言，存在着一些与该用户有过历史交互行为的项目，也有着大量没有明显交互的项目，在这里将前者称为正向反馈项目，后者称为负向反馈项目。这里设置的优化目标函数旨在最大化用户对正向反馈项目以及负向反馈项目的预测排序分值之差。

(4) 选取合适的实验评估方法评测推荐性能。针对要实现的 Top-N 个性化列表推荐任务，将采用该任务常用的 Recall、Precision 以及 NDCG 指标来对推荐性能进行评估，并设计相关实验对模型的冷启动改善能力进行评估。

本文基于注意力的异质图网络社会推荐方法整体设计流程如下图：

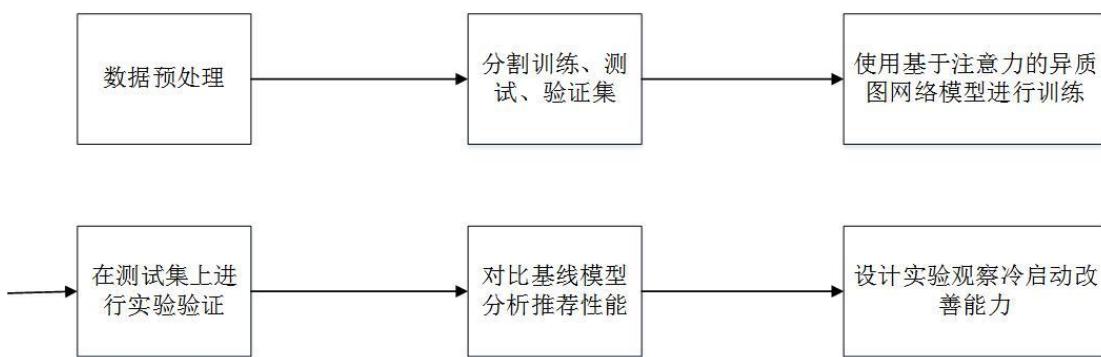


图 3-1 基于 HIN 的图神经网络推荐方法流程示意图

Figure 3-1 The process of the HIN-based graph neural network recommendation method

从上述流程示意图中可以看出，本文的整体设计主要包括数据预处理、模型训练、实验验证这三个阶段。

在数据预处理阶段，首先观察数据集特征，进行数据清洗，然后将用户、项

目以及相关属性用具有一定排列顺序的数字表示，并将整个数据集按照 7:1:2 的比例划分为训练集、验证集以及测试集。

在模型训练阶段，采用基于元路径内部以及元路径之间的聚合方式定义节点的嵌入表示，并通过优化目标函数训练相关参数。

在实验评估阶段，除了与基线模型对比不同 Top-N 情况下的推荐性能，也设计了相关实验评估模型冷启动改善能力。将数据集分别以 20%、40%、60%、80% 的比例分割出来，代表不同程度的稀疏度，从而与基线模型对比实验效果。

3.3 数据集分析

本文构建的是基于异质信息网络的算法模型，希望能充分利用用户和项目的附加信息，所以在选取数据集时考虑到需包含相关属性信息以及用户之间的关联信息。另外，数据集的相关特征也是需要考虑的因素。本节将介绍数据集的基本情况，并且对相关数据特征进行分析。

3.3.1 数据集描述

本文用到了两个数据集，一个是 MovieLens 数据集，另一个是 LastFM 数据集。其中，MovieLens 包含了用户对电影的评分信息以及电影的相关属性信息。LastFM 数据集是关于音乐推荐的数据集，包含了用户比较喜爱的艺术家列表以及对应歌曲播放次数，同时也包含了艺术家相关属性信息。本文所使用的这两个数据集字段描述如表 3-1 所示：

表 3-1 MovieLens 和 LastFM 数据集字段描述
Table 3-1 Detailed description of datasets

数据集	数据集字段	含义
LastFM	User	听歌用户
	Artist	用户喜爱的艺术家
	Tag	艺术家所带标签
MovieLens	User	观影用户
	Movie	观看的电影
	Genre	电影所属类型

为了更好对异质图中的相关推荐算法进行实验验证，有研究学者利用皮尔逊相关系数计算用户之间的相关性，在这里可以理解为相似度更高的用户具有更类

似的品味，可以看作用户的社交信息，由此用户与用户之间也有了直接关联从而能够丰富元路径的表示意义，类似地，通过相似性计算，电影之间也有了直接关联的依据。

在 LastFM 数据集中，艺术家所带的标签是指该艺术家创作的音乐风格，比如摇滚中的金属乐等，在 MovieLens 数据集中，不同的电影类型包括动作、冒险、喜剧等。本文所使用的这两个数据集具体情况如下表 3-2 所示：

表 3-2 MovieLens 和 LastFM 数据集的数据统计

Table 3-2 Data statistics of datasets

Datasets	Relation(A-B)	Number(A)	Number(B)	Number(A-B)
LastFM	User-Artist	1,892	17,632	92,834
	User-User	1,892	1,892	18,602
	Artist-Artist	17,632	17,632	153,399
	Artist-Tag	17,632	11,945	184,941
MovieLens	User-Movie	943	1,682	50,126
	User-User	943	943	4,568
	Movie-Movie	1,682	1,682	82,798
	Movie -Genre	1,682	18	2,891

如表 3-2 所示，对两个数据集所包含的不同类型对象以及对象之间不同类型的连边进行了数量统计。

3.3.2 数据集特征分析

在上述两个数据集中，都包含了用户和项目的交互记录、用户与用户之间的关联信息以及项目的属性信息，用户和项目的直接交互记录是用户对项目偏好的直接凭据，而其他的信息则可以看作是用来推测用户偏好的附加信息。两者数据的分布特征有所不同，所以这里先分析一下两个数据集中已有用户项目对在不同区间的分布情况，对数据特征的提取步骤如表 3-3 所示：

表 3-3 数据特征提取步骤

Table 3-3 Data feature extraction steps

输入：user_movie.dat, user_artist.dat 文件

输出：以字典形式存放每个用户交互过的项目数量

- 利用 pandas.read_csv 函数读取相应文件并分别存放在 um、ua 两变量中

3-3 续表（数据特征提取步骤）

2. 分别建立空字典 ua_dic、um_dic

3. 遍历每一行，将用户 id 作为键，交互过的项目数量作为值存入字典

for line in lines:

```
line = line.strip('\n').split('\t')
```

```
if line[0] in ua_dic:
```

```
    ua_dic[line[0]]+=1
```

```
else:
```

```
    ua_dic[line[0]]=1
```

4. 输出 ua_dic, um_dic

进行上述处理后得到包含用户以及其观看电影数量的键值对，其中键表示用户 id，由于篇幅原因，这里展示 MovieLens 数据集中部分用户观看的电影数量，如图 3-2 所示：

```
{'186': 44, '244': 107, '298': 67, '253': 48, '6': 111, '286': 136, '210': 70, '303': 246, '194': 147, '234': 236, '167': 31, '291': 151, '95': 150, '102': 104, '160': 55, '301': 121, '290': 79, '157': 24, '278': 13, '7': 212, '284': 24, '276': 269, '246': 92, '249': 78, '178': 128, '81': 29, '25': 41, '72': 74, '292': 62, '20': 27, '13': 338, '138': 27, '60': 102, '223': 54, '243': 44, '241': 15, '254': 77, '127': 11, '299': 138, '222': 195, '42': 99, '162': 22, '279': 218, '119': 82, '62': 113, '135': 32, '90': 166, '293': 173, '166': 1, '0': 271, '136': 265, '27': 110, '63': 237, '19': 128, '92': 44, '74': 22, '82': 97, '250': 66, '87': 11, '3': 305, '124': 43, '104': 57, '54': 269, '173': 99, '64': 259, '24': 121, '39': 49, '118': 155, '9': 172, '1': 5, '268': 193, '80': 15, '66': 22, '200': 94, '213': 52, '94': 218, '38': 61, '130': 184, '256': 107, '92': 1, '88': 251, '42': 42, '181': 253, '161': 29, '125': 92, '58': 78, '151': 159, '91': 51, '294': 78, '36': 11, '63': 4, '4': 14, '46': 46, '83': 87, '192': 21, '307': 56, '297': 110, '85': 154, '59': 193, '193': 54, '275': 52, '219': 1, '7': 123, '28': 158, '89': 302, '9': 201, '195': 1, '1': 139, '23': 75, '296': 67, '33': 8, '16': 64, '154': 30, '270': 75, '187': 27, '170': 12, '184': 120, '218': 22, '215': 43, '104': 59, '267': 87, '240': 11, '18': 15, '2': 308, '198': 198, '191': 10, '61': 9, '272': 22, '207': 106, '203': 25, '197': 62, '134': 13, '109': 115, '5': 8, '0': 174, '101': 101, '216': 63, '65': 31, '111': 19, '189': 78, '214': 66, '285': 14, '96': 26, '115': 47, '116': 7, '5': 70, '235': 44, '164': 30, '224': 66, '129': 15, '64': 101, '45': 26, '21': 88, '264': 73, '131': 17, '230': 68, '231': 10, '288': 35, '56': 95, '79': 26, '239': 83, '69': 42, '22': 72, '282': 10, '118': 42, '17': 1, '16': 16, '10': 88, '107': 14, '226': 29, '306': 18, '280': 127, '173': 23, '185': 22, '188': 57, '311': 164, '165': 21, '263': 57, '274': 33, '295': 96, '93': 6, '77': 41, '236': 62, '156': 19, '168': 34, '141': 50, '2': 21, '81': 81, '195': 44, '108': 19, '37': 29, '113': 25, '255': 39, '89': 30, '140': 10, '11': 93, '150': 17, '6': 8, '14': 14, '180': 35, '145': 157, '84': 38, '17': 18, '148': 26, '233': 48, '217': 38, '198': 98, '283': 23, '5': 3, '14': 24, '24': 31, '75': 43, '313': 133, '149': 22, '106': 31, '78': 14, '196': 20, '190': 31, '220': 10, '27': 22, '262': 77, '206': 33, '76': 46, '314': 122, '248': 24, '179': 20, '122': 33, '304': 12, '3': 29, '2':
```

图 3-2 MovieLens 中部分用户观看电影数量

Figure 3-2 Parts of user-movie in MovieLens

(1) LastFM 数据集

通过对 LastFM 数据集中用户与艺术家的交互数据进行分析，具体情况如表 3-4 所示，从中可以看出，听过 50 个艺术家歌曲的用户群占了绝大部分，其他占据少数，这也就说明每个用户的项目大多相对持平，只有极少数用户的交互记录较少。整体而言，用户的交互数据分配地比较平均。

表 3-4 LastFM 数据集中用户与项目交互数据分布
Table 3-4 Distribution of the user-project interaction in LastFM

用户听过的艺术家数量	用户数量
[1,20)	32
[20,40)	18
[40,50)	13
50	1829
总计	1892

(2) MovieLens 数据集

通过对该数据集中用户电影观看记录的数据进行分析，得到分布情况如下图所示：

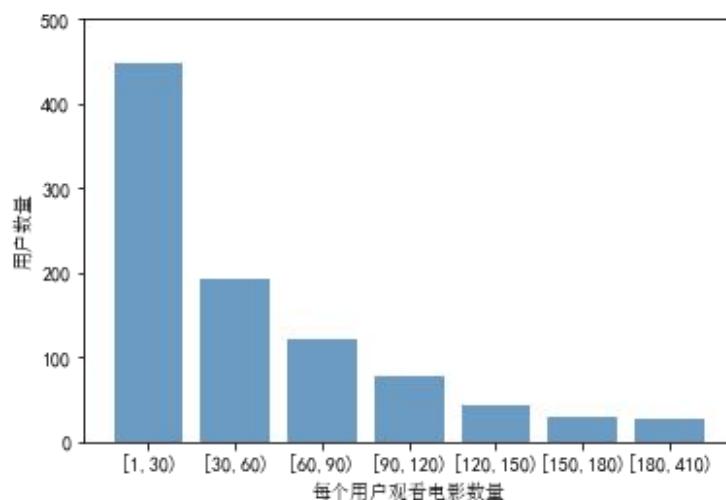


图 3-3 MovieLens 数据集中用户观影数量分布
Figure 3-3 Distribution of the user-movie interaction in MovieLens

由图 3-3 可以看出，观看电影数量在 30 以下的用户占据最多比重，随着观看电影数量的增加，用户群的数量急剧递减，这说明用户项目交互行为稀疏的用户占据绝大多数，由此利用附加信息改善推荐性能的发挥空间就更大。

通过对两个数据集的用户项目交互行为数据进行分析，可以看出 LastFM 数据集中每个用户交互过的项目数量大都一致，只有极少数用户含有较少的交互数据，但是 MovieLens 数据集中大多数用户交互过的项目数量是稀少的，这也就意味着附加信息的作用可能就越大。通过利用这两种具有不同数据特征的数据集在模型上进行实验，可以更好地看出模型的改善能力。

3.3.3 数据预处理

为了高效利用模型完成实验，在送入到模型之前需要对原始数据进行一些处理。数据预处理主要包括构建邻接矩阵、生成元路径表示，再划分训练、验证、测试集这几个过程。

(1) 构建邻接矩阵。原始的数据是由相关数据格式文件组成的，首先需要把这些数据构建成邻接矩阵。以 MovieLens 数据集为例，里面包含了 user_movie、movie_movie、user_user 以及 movie_genre 这样的文件，文件里面含有类似 user、movie 这样的对象并且是以数字 ID 形式呈现，为方便后续数据处理先对 ID 重新编号。通过 pandas 的相关函数读取这些文件，仍以电影数据集为例，使用了 user、movie 以及 genre 这几种对象，通过上面重新编号的 ID 可得到每种对象的数量。构建邻接矩阵的关键步骤如表 3-5 所示，先设置维度为这三种对象数量之和的全 0 矩阵，再对读取的每一个文件遍历重置矩阵中的值，比如某一个用户电影对有交互记录，重置后该坐标所对应的矩阵值就是评分值，如此依次遍历相关文件重置矩阵值得到有意义的邻接矩阵。

表 3-5 构建邻接矩阵核心步骤
Table 3-5 Core steps for constructing adjacency matrix

输入：user_movie.dat、user_user.dat、movie_genre.dat 文件

输出：邻接矩阵 adjM

- 利用 pandas.read_csv 函数读取文件内容并分别存放在 user_movie、user_user、movie_genre 变量中

- 构建二维全 0 矩阵

```
dim = num_user + num_movie + num_genre
adjM = np.zeros((dim, dim), dtype = int)
```

- 遍历步骤 1 中生成的三个变量，对应位置数值重置(下例为遍历 user_movie)

```
for _, row in user_movie.iterrows():
    uid = row['userID'] - 1
    mid = num_user + row['movieID'] - 1
    adjM[uid, mid] = max(1, row['rating'])
    adjM[mid, uid] = max(1, row['rating'])

return adjM
```

(2) 生成元路径表示。通过上述建立的邻接矩阵可以将对象间的关联信息以数组形式表示，然后定义有意义的元路径形式，得到不同元路径对应的数组信息，

比如 `user_artist_list` 表示所有用户交互过的项目数组表示。两个数据集定义的元路径信息以及含义描述如下表所示：

表 3-6 元路径设计及描述
Table 3-6 Description of meta-path in MAGNN-Rec

数据集	路径规则	关联关系
LastFM	U-U	喜好高度相似的用户对
	U-A-U	听过同一个艺术家的用户对
	U-A-T-A-U	听过带有同一种标签两个艺术家的用户对
	A-U-A	被同一个用户听过的项目对
	A-T-A	带有同一种标签的项目对
	A-U-U-A	分别被两个爱好相似用户听过的项目对
Movielens	U-U	喜好高度相似的用户对
	U-M-U	看过同一个电影的用户对
	U-M-G-M-U	看过同一种类型的两部电影的用户对
	M-U-M	被同一个用户看过的项目对
	M-G-M	属于同一种类型的项目对
	M-U-U-M	分别被两个爱好相似用户看过的项目对

在设计元路径时着重考虑了项目的类型等属性信息，并且把用户的社交信息融合进来，这样构建的异质图不仅包含了用户和项目的直接交互信息，也囊括了用户的社交信息和项目的相关属性信息，为预测用户偏好提供更多有意义的关联信息。

由表 3-6 可以看出，本文针对每一个数据集设计了六种不同的元路径形式，这种设计方式可以帮助我们找到没有直接相连却有潜在关系的节点对。以 Movielens 为例，数据集中路径规则为 U-M-U 的元路径为例，两个用户跟同一个项目有了交互，也就意味着两个用户的品味有着一定程度的相似性，根据该潜在信息有利于推测用户偏好。如表 3-7 所示，在实际数据处理过程中，通过对邻接矩阵进行处理得到了 `user_movie_list` 这样的数组表示，要得到符合 U-M-U 路径形式的所有数组，先定义 `u_m_u` 的空数组，然后遍历 `user_movie_list` 提取跟同一个 movie 有关联的 user 对，转换成 `[user,movie,user]` 的形式添加到预设的 `u_m_u` 数组当中。通过这样的方法提取不同形式元路径的节点序列，以一定的形式保存起来，完成元路径的数据表示。

表 3-7 生成元路径表示核心步骤

Table 3-7 The core steps of generating meta-path representation

输入：邻接矩阵 adjM

输出：用户、项目对应的元路径文件，邻接矩阵文件

1. 遍历邻接矩阵 adjM 得到不同类型的字典列表(下例为生成 user_movie_list)

```
for i in range(num_user)
    t = adjM[i,num_user:num_user+num_movie].nonzero()[0]
    user_movie_list = {i:t}
```

2. 定义对应六个不同元路径形式的空数组，遍历字典列表生成不同元路径数组（下例为生成 u_m_u）

```
u_m_u = []
for m, u_list in movie_user_list.items():
    u_m_u.extend([(u1, m, u2) for u1 in u_list for u2 in u_list])
u_m_u = np.array(u_m_u)
u_m_u[:, 1] += num_user
sorted_index = sorted(list(range(len(u_m_u))), key=lambda i : u_m_u[i, [0, 2, 1]].tolist())
u_m_u = u_m_u[sorted_index]
```

3. 赋予元路径数组对应键并写入变量

```
metapath_indices_mapping = {(0, 1, 0): u_a_u,
                             (0, 1, 2, 1, 0): u_a_t_a_u,
                             (0, 0): u_u,
                             (1, 0, 1): a_u_a,
                             (1, 2, 1): a_t_a,
                             (1, 0, 0, 1): a_u_u_a}
```

4. 将元路径数据写入元路径.pkl、元路径.adjlist 文件，邻接矩阵写入 adjM.npz 文件
-

```
idx00
[[ 0  944  0]
 [ 0  946  0]
 [ 0  948  0]
 ...
 [ 942 1168  0]
 [ 942 1170  0]
 [ 942 1181  0]]
```

图 3-4 U-M-U 元路径实例表示

Figure 3-4 Meta-path instances in U-M-U

如图 3-4 所示，描述对象是编号为 0 的用户节点，图中展示的是该节点通过元路径 U-M-U 与其他节点产生关联的所有元路径实例表示。

(3) 划分训练、验证、测试集。初始数据中已经包含了用户项目交互记录，接下来要得到没有交互过的用户项目对。由于在开始已经对用户、项目等重新进行编号，这里先利用 pandas 相关函数读取文件并将数据保存在变量中，然后通过两层遍历循环并设置条件就能得到没有直接交互的用户项目对。将原始数据中的跟用户交互过的项目看作正向反馈，这里看作 positive 节点对，没有历史交互记录的项目看作负向反馈，这里看作 negative 节点对。在两类中对所有的用户项目对以 7:1:2 的比例进行划分，分别当做训练、验证以及测试集，在两个数据集中划分后具体 UI (user-item) 对的数量统计如表 3-8 所示：

表 3-8 数据集划分详情
Table 3-8 Details of data set division

	type	train	validation	test	total
LastFM	positive	6,4984	9283	1,8567	9,2834
	negative	2335,1822	333,5974	667,1948	3335,9744
Movielens	positive	3,5136	4587	1,0403	5,0126
	negative	107,5253	15,3600	30,7147	153,6000

3.4 本章小结

本章首先介绍了本文实现的 Top-N 列表推荐任务场景，然后对框架整体设计以及数据集分析处理进行介绍。

在进行框架介绍时先阐述了设计流程时需要考虑的问题，并给出了解决方案。需要考虑将用户、项目以及相关属性信息充分利用到，可以将这些信息构建到异质信息网络中去。还需要完成异质图上节点的嵌入表示，可以定义有意义的元路径形式并通过不同的注意力机制融合不同语义的异质信息，从而更好挖掘用户偏好信息。在对模型进行训练时需定义优化目标函数，这里可以设置为最大化对正、负向反馈项目的预测评分之差。在对模型进行评估时需要选择合适的评估指标，这里针对 Top-N 应用场景可以采用 Precision、Recall 等常用指标进行评估。

在进行数据集分析时，展示了数据相关统计信息，并对数据集进行特征分析，随后介绍了预处理过程。先对数据集进行了介绍，利用图表展示了所用数据集的基本情况。随后对两个数据集的用户项目对分布情况进行分析，发现两者具有不同特征，使用这两个数据集有利于对比模型改善能力。随后介绍了预处理流程，

主要包括构建邻接矩阵、生成元路径表示以及划分训练、验证、测试集这几个步骤。

4 基于 HIN 的图神经网络社会推荐算法

本章针对 Top-N 个性化列表推荐任务场景，提出基于注意力机制的异质图网络推荐模型 MAGNN-Rec，设计相关实验进行性能评估。在模型结构方面，与现有的 MCRec^[17]模型相比，设计了基于关系旋转的用户项目所在元路径实例信息聚合方式，在原有模型的基础上增加了对边信息的利用，以解决之前位置信息缺失的问题；在元路径实例信息聚合层面，引入注意力机制区分不同元路径实例的重要程度，以更好地挖掘异质信息对用户和项目进行表征。通过实验验证对模型进行性能评估，然后选取合适数据集设计实验观察冷启动改善能力。

4.1 基于 MAGNN 的个性化推荐算法

为了给用户推荐更多他们感兴趣的的商品，不少研究学者引入了一些辅助信息，比如项目的属性信息、用户的社交信息等，将其建模到异质信息网络中，通过元路径表示学习它们之间的潜在关系特征。

MCRec^[17]是学者们于 2018 年提出来的基于异质信息网络的推荐模型，其主要思想是考虑不同元路径所提供的上下文信息，构成<user,meta-path,item>的三元组表示。将用户的最终嵌入表示记为 \tilde{x}_u ，项目的最终嵌入表示记为 \tilde{y}_i ，假设连接该用户项目对的有路径规则为 $U \xrightarrow{\text{rate}} M \xrightarrow{\text{genre}} G \xrightarrow{\text{genre}^{-1}} M$ 和 $U \xrightarrow{\text{rate}} M \xrightarrow{\text{rate}^{-1}} U \xrightarrow{\text{rate}} M$ 的两条元路径形式，模型中使用了注意力机制将这两个元路径所包含的信息以不同的权重聚合起来，记为 $c_{u \rightarrow i}$ ，将这三种信息融合起来构成整体表示，如式(4-1)所示：

$$\tilde{x}_{u,i} = \tilde{x}_u \oplus c_{u \rightarrow i} \oplus \tilde{y}_i \quad (4-1)$$

其中 \oplus 表示聚合操作。该算法主要关注三个方面的信息，用户、项目以及 UI (user-item) 对之间基于元路径的上下文信息，用户、项目使用 one-hot 向量表示，在对 $c_{u \rightarrow i}$ 表示时，首先对 UI 对所在的元路径实例根据其整体相似性进行筛选，保留相似度较高的 K 条实例参与后面的信息聚合，接着将元路径实例内部的节点向量信息送入卷积层进行编码，得到元路径实例的向量信息后进行最大池化操作得到元路径向量表示，再进行加权聚合拼接等操作得到最后的向量表征。该方法在构建异质信息网络时考虑了用户的社交信息、项目的相关属性信息，并挖掘了基于元路径的上下文信息，有利于学习用户偏好。但是也存在一些不足，一方面是针对某一特定路径规则下的元路径实例，如图 4-1 所示，聚合的过程只考虑到节点信息，相关的位置信息没有被利用到，信息利用不充分。另一方面是针对不同元路径实例信息的聚合，如图 4-2 所示，只是采用最大池化操作生成该路径规则下的

元路径的向量表示，融合策略较为粗糙。

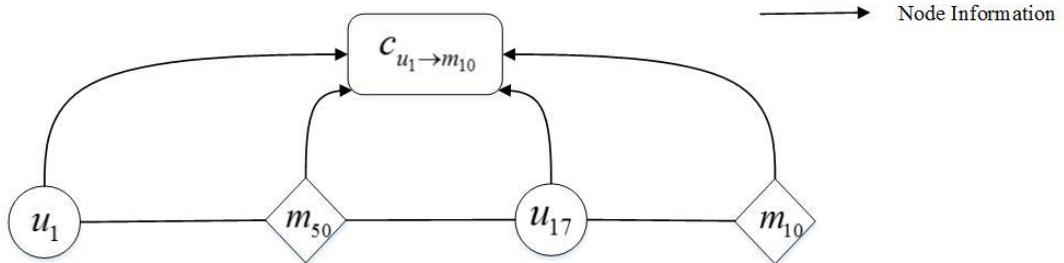


图 4-1 MCRec 中元路径实例编码示意图

Figure 4-1 Schematic diagram of meta path instance coding in MCRec

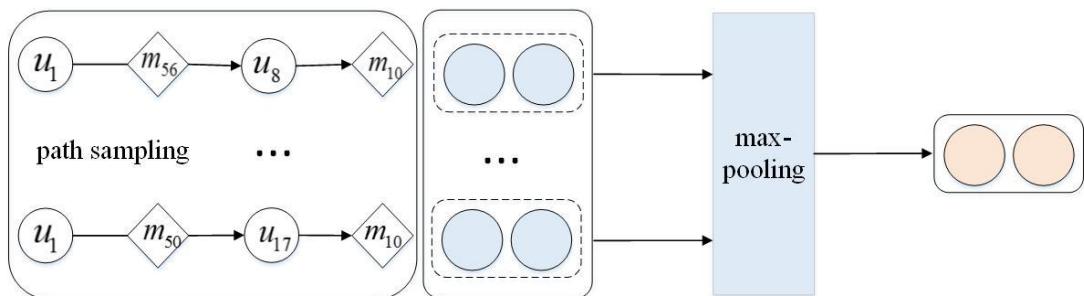


图 4-2 MCRec 中聚合不同元路径实例过程

Figure 4-2 Process of aggregating different meta-path instances in MCRec

针对上述模型中存在问题，本文参考了学者们于 2020 年提出来的 MAGNN^[41] 算法模型，该模型被用来解决大规模图中的节点分类问题。MAGNN 主要思想是利用元路径充分挖掘异质图上的节点信息，利用图神经网络能够很好地学习节点在图上的低维向量表示。以 DBLP 数据集为例，目标节点类型为作者，属性有发布的论文、所属会议以及论文中的关键字，通过建立不同的关联构建异质信息网络，挖掘异质信息对作者节点进行向量表征，判断其所属标签类型。应用在推荐任务上，为了缓解数据稀疏问题，通常利用社交、项目属性等附加信息构建异质信息网络，该思想有利于充分挖掘用户项目特征图中的异质信息，从而得到用户和项目丰富的嵌入表示，因此本文提出 MAGNN-Rec 模型进行 Top-N 个性化列表推荐。与节点分类任务不同的是，在推荐系统存在用户和项目这两种类型的目标对象，也就是需要针对用户和项目分别建立不同的元路径表示体系，从而分别对用户和项目进行表征学习，同时用户、项目等节点没有所谓的标签，需要采用无监督方式训练。

将推荐系统中的用户、项目以及属性等对象看作是异质图中的节点，提升推荐性能的关键在于如何有效地完成用户和项目表征，建模到异质图网络中也就是如何学习节点的图嵌入表示。之前的 MCRec 推荐模型为了在稀疏数据集上充分挖掘有效信息引入了项目的属性信息以及用户的社交信息，构成异质信息网络后设

计了不同的元路径形式，除了对用户和项目进行嵌入表示，将连接两者的元路径信息也进行了表征。但是 MCRec 模型针对每一条元路径实例进行信息聚合时，只考虑了与目标节点直接或间接相连的节点信息，忽略了它们之间边的特征，因此本文提出了基于关系旋转的用户项目所在元路径实例信息聚合方式，来充分对用户和项目进行表征，同时，每一种元路径类型内部不同元路径实例对目标节点的贡献值应该是不同的，所以还引入了注意力机制学习它们不同的权重，最终得到的嵌入表示为这些向量的加权聚合。本文提出的 MAGNN-Rec 模型整体流程结构如下图所示：

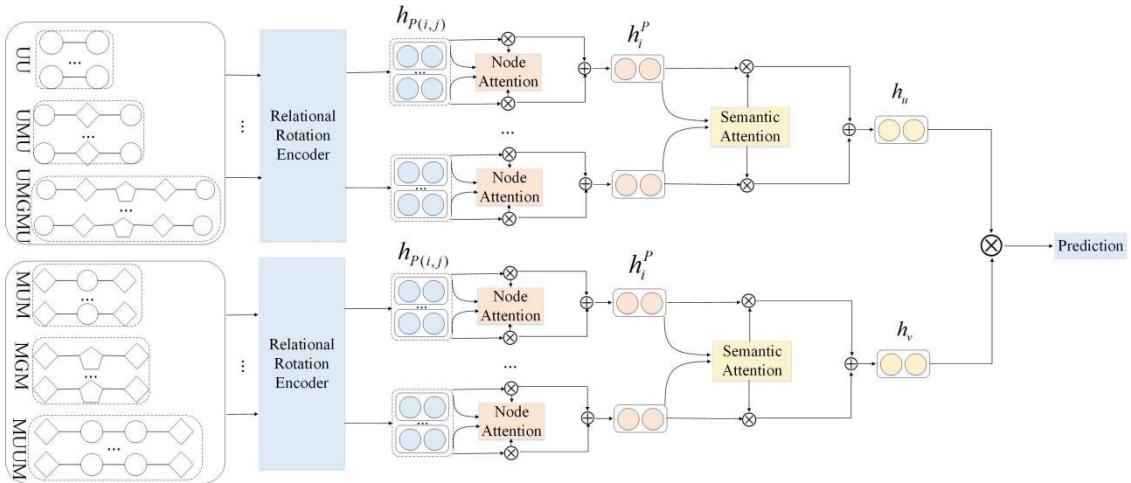


图 4-3 MAGNN-Rec 模型整体结构
Figure 4-3 The overall structure of MAGNN-Rec

如图 4-3 所示，最左边分别代表了用户和项目在异质图中的部分结构，为了符合本文所使用的数据集以及元路径结构形式，每一个目标节点分别用三条元路径类型作为样例，同时每一种元路径可能包括一个或多个元路径实例。元路径实例具体含义已经在 2.2 节中介绍过，这里不再赘述。不同形状代表了不同的对象，以图 4-3 中用户节点的图表示为例，目标节点三条元路径分别表示 $U \xrightarrow{\text{friend}} U$, $U \xrightarrow{\text{rate}} M \xrightarrow{\text{rate}^{-1}} U$, $U \xrightarrow{\text{rate}} M \xrightarrow{\text{genre}} G \xrightarrow{\text{genre}^{-1}} M \xrightarrow{\text{rate}^{-1}} U$ ，每一种元路径有不同数量的元路径实例。针对每一条元路径实例，与目标节点直接或间接相连的每一个邻近节点信息都会汇聚到目标节点上，从而将每一个元路径实例聚合成一个向量表示，这样在每一个目标节点上能充分挖掘到元路径内部的上下文信息，从而使表征可解释性更强。随后采用多头注意力机制将这些不同的元路径实例再以不同的权重将信息聚合到目标节点上，一方面能够有效聚合元路径实例信息，另一方面采用多头注意力机制能够缓解异质图所带来的高方差。在完成节点内部信息聚合以后，再对不同元路径进行语义级别的信息聚合，最终得到用户和项目丰富的嵌入表示，再进行内

积操作，完成个性化列表推荐任务。

在对用户和项目节点进行向量表征时，关键的算法伪代码如下表所示：

表 4-1 节点向量表征过程

Table 4-1 Node vector representation process

输入：异质图 $\mathcal{G}=(V, E)$,

节点类型 $\mathcal{A} = \{A_1, A_2, \dots, A_{|\mathcal{A}|}\}$,

元路径 $\mathcal{P} = \{P_1, P_2, \dots, P_{|\mathcal{P}|}\}$,

节点特征 $\{x_i, \forall i \in \mathcal{V}\}$,

注意力头数 K,

层数 L

输出：用户项目节点嵌入 $\{z_i, \forall i \in \mathcal{V}\}$

1. for node type $A \in \mathcal{A}$ do
2. 节点特征转换 $W_A \bullet x_i \rightarrow h_i^0, \forall i \in \mathcal{V}_A$
3. end
4. for $l = 1 \dots L$ do
5. for node type $A \in \mathcal{A}$ do
6. for meta-path $P \in \mathcal{P}_A$ do
7. for $i \in \mathcal{V}_A$ do
8. $h_{P(i,j)} = f_\theta(P(i,j)) = f_\theta(h_i^l, h_j^l | h_t^l, \forall t \in \{m^{P(i,j)}\})$
9. $\sigma(\sum_{j \in \mathcal{N}_i^P} [\alpha_{ij}^P]_k \bullet h_{P(i,j)}^l) \rightarrow [h_i^P]^l$
10. end
11. end
12. $\sum_{P \in \mathcal{P}_A} \beta_P \bullet [h_i^P]^l \rightarrow [h_i^{\mathcal{P}_A}]^l, \forall i \in \mathcal{V}_A$
13. end
14. $h_i^l = \sigma(W_o^l \bullet [h_i^P]^l), \forall i \in \mathcal{V}_A, \forall A \in \mathcal{A}$
15. end
16. $h_i^L \rightarrow z_i, \forall i \in \mathcal{V}$

如表 4-1 所示，在进行前向传播过程时，首先对不同节点类型的节点特征映射到同一向量空间中，针对每一层网络，分别对用户和项目节点进行向量表征，主

要有两个步骤，先汇聚目标节点所在不同元路径实例信息，接着对不同元路径信息加权聚合。针对基线模型存在的问题，主要对元路径实例信息聚合过程做了改进：1) 利用元路径实例中的序列结构信息进行编码；2) 引入注意力机制聚合不同元路径实例信息。

4.1.1 融合用户项目所在元路径实例信息

在对用户和项目进行表征前需要先进行节点特征转换。对有着多种类型节点的异质图来说，不同节点类型的节点可能对应着不同维度的特征向量，比如说数据集中的用户节点和项目属性节点，这两种类型对象的向量维度大概率是不同的。即使不同类型的节点恰好有着同样维度的特征向量，它们也处在不同的特征空间中。比如说有个维度为 n_1 的文本词向量，另外有个维度为 n_2 的图像直方图向量，即使 $n_1 = n_2$ ，两者也不能直接聚合成一起。

在同一个框架里去处理不同维度的特征向量是比较麻烦的，所以在进行节点信息聚合之前需要将不同类型的特征向量映射到同一特征空间中去。在这里通过特定类型线性转换，将用户、项目、属性这些不同类型的节点特征向量转换到同一个隐藏向量空间中。针对属于类型 $A \in \mathcal{A}$ 的节点 $i \in \mathcal{V}_A$ ，转换过程如式(4-2)所示：

$$\vec{h}_i = W_A \cdot \vec{x}_i^A \quad (4-2)$$

其中， $\vec{x}_i^A \in \mathbb{R}^{d_A}$ 表示节点 i 的原始特征向量， W_A 表示节点类型为 A 的参数权重矩阵， \vec{h}_i 表示经过转换后节点 i 的隐藏向量表示。通过该线性转换操作解决了图中含有不同内容特征节点的异质问题，异质图中所有节点的隐藏向量享有相同的维度，方便后续信息聚合等步骤。

转换后的节点信息可以进行元路径内部信息聚合。在路径规则为 P 的元路径形式下，编码器能够对元路径内部不同实例进行信息聚合，除了目标节点本身的信息，还可以学习到与其相连的邻近节点信息，以及它们之间隐含的语义信息和结构信息。针对元路径 P ，这里将连接目标节点 i 与邻近节点 j 的元路径实例记为 $P(i, j)$ ，通过元路径 P 与目标节点 i 相连的所有邻近节点集合记为 \mathcal{N}_i^P ，定义 $P(i, j)$ 的内部节点为 $\{m^{P(i,j)}\} = P(i, j) \setminus \{j, i\}$ 。在元路径实例 $P(i, j)$ 中，通过元路径实例编码器汇聚每一个节点的特征向量，最终生成一个单一的嵌入表示，如公式(4-3)所示：

$$\vec{h}_{P(i,j)} = f_\theta(P(i, j)) = f_\theta(\vec{h}_i, \vec{h}_j \{ \vec{h}_t, \forall t \in \{m^{P(i,j)}\} \}) \quad (4-3)$$

其中 $\vec{h}_{P(i,j)} \in \mathbb{R}^{d'}$ 为聚合后的维度为 d' 的单一向量， f_θ 表示实例编码运算， \vec{h}_i 和 \vec{h}_j 分别表示目标节点和邻近节点进行线性转换后的隐藏向量。在目标节点 i 和邻近节点

j 之间会存在一条或者多条元路径实例，例如有两个元路径实例都遵循路径规则为 $U \xrightarrow{\text{rate}} M \xrightarrow{\text{rate}^{-1}} U$ 的形式，分别为 $u_1 \rightarrow m_1 \rightarrow u_2$ 和 $u_1 \rightarrow m_2 \rightarrow u_2$ ，这样用户节点 u_1 和 u_2 之间就存在多条元路径实例。

针对元路径实例，需要一种方式将目标节点与元路径实例内部的邻近节点信息聚合起来。RotatE^[42]提出一种关系旋转的建模方式，是用在知识图谱嵌入学习中的，类比一下对这里的元路径实例信息聚合也很适用。假设元路径实例表示如下：

$$P(i, j) = (t_0 \xrightarrow{R_1} t_1 \dots \xrightarrow{R_l} t_n) \quad (4-4)$$

其中 t_n 就表示目标节点 i ， t_0 表示通过元路径实例相连的另一端节点 j ， t_1, \dots, t_{n-1} 表示通过元路径实例与目标节点 i 相连的中间节点， R_l 表示节点 t_{l-1} 和 t_l 之间的关系。那么目标节点与它邻近节点之间的聚合方式如下所示：

$$o_0 = h_{t_0}^i = h_j^i \quad (4-5)$$

$$o_l = h_{t_l}^i + o_{l-1} \odot r_l \quad (4-6)$$

$$h_{P(i,j)} = \frac{o_n}{n+1} \quad (4-7)$$

其中， r_l 为关系 R_l 的向量表示， \odot 表示两个向量中对应元素逐一进行相乘运算。公式 4-5 表示初始化，首先以节点 j 的向量作为初始向量，接着通过式(4-6)进行迭代聚合，得到 o_n 的向量表示后通过式(4-7)进行归一化得到最终聚合向量表示。

如图 4-4 所示，以用户节点所处的 $U \xrightarrow{\text{rate}} M \xrightarrow{\text{rate}^{-1}} U$ 路径形式为例，假设目标节点 u_1 处在 $u_1 \rightarrow m_2 \rightarrow u_2$ 这样一条元路径实例中，迭代运算中包含了每一个邻近节点本身的信息以及关联的边信息。这种信息聚合方式不仅让每一个向量都参与运算，还保留了元路径的序列结构，有利于对目标节点进行准确表征。

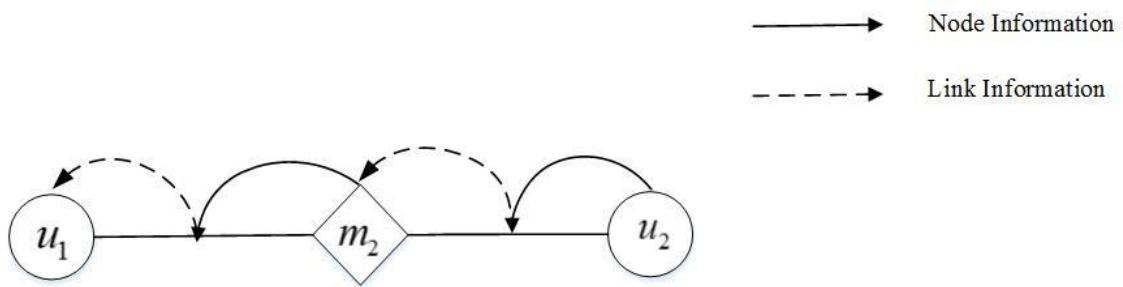


图 4-4 元路径实例内部信息聚合示意图

Figure 4-4 Internal information aggregation of meta-path instances

由于异质图中存在不同类型的节点，它们所对应的特征向量维度以及特征空

间是不同的，所以在进行节点的向量表征前需要通过线性转换将它们映射到同一个特征空间。有了节点初始向量，针对特定元路径中的不同元路径实例，目标节点通过不同关联类型与一个或多个节点相连，这里采用关系旋转的编码方式，将每一条元路径实例中包括目标节点以及与之相连的邻近节点信息聚合成一个向量，从而丰富目标节点的嵌入表示。

4.1.2 信息融合引入注意力机制

在得到每一条元路径实例的向量表示后，需要针对目标节点 i 将这些向量表示聚合起来，考虑到不同元路径实例对目标节点应该对应不同的权重，这里引入注意力机制对它们进行加权聚合，具体流程如下图所示：

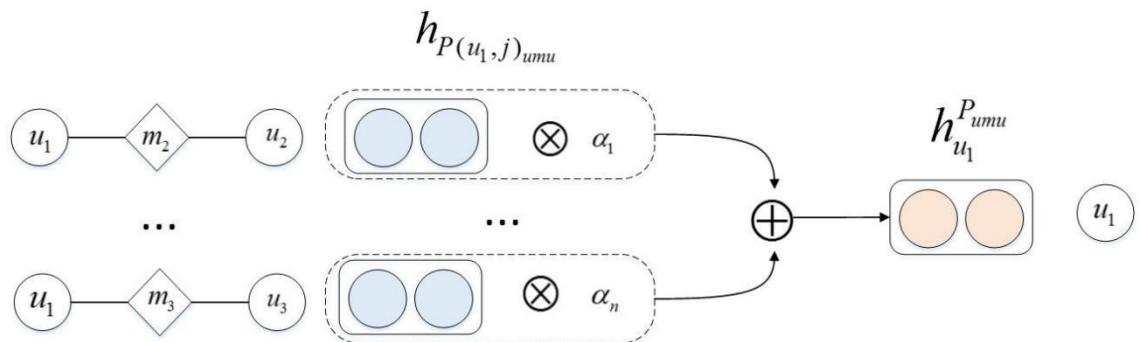


图 4-5 不同元路径实例信息融合示意图

Figure 4-5 Schematic diagram of different meta-path instances fusion

如图 4-5 所示，这里以用户节点类型举例，以路径规则为 $U \xrightarrow{\text{rate}} M \xrightarrow{\text{rate}^{-1}} U$ 的元路径为例，针对目标节点 u_1 ，包含了 $u_1 \rightarrow m_2 \rightarrow u_2$ 和 $u_1 \rightarrow m_3 \rightarrow u_3$ 以及其他相关元路径实例，每一条元路径实例对目标节点 u_1 的贡献值应该是不同的，这里利用权重参数将它们区分开来。

给定元路径 P ，针对每一个符合该路径规则的元路径实例，可以学习一个归一化权重参数 α_{ij}^P ，其具体计算公式如下所示：

$$e_{ij}^P = \text{LeakyReLU}(\alpha_p^T \cdot [h_i^T \| h_{P(i,j)}]) \quad (4-8)$$

$$\alpha_{ij}^P = \frac{\exp(e_{ij}^P)}{\sum_{s \in \mathcal{N}_i^P} \exp(e_{is}^P)} \quad (4-9)$$

其中， $\alpha_p^T \in \mathbb{R}^{2d'}$ 是元路径 P 的参数注意力向量， $\|$ 表示向量间的拼接操作。常用 ReLU 函数表示线性修正单元，特点是出现负值就设为零，而这里 LeakyReLU 函数是出

现负值就赋予它非零斜率，这样即使出现负值导数也不至于为零。

由公式(4-8)可以获得元路径实例 $P(i, j)$ 对目标节点 i 的重要程度 e_{ij}^P ，经过公式(4-9)运算获得了所有节点 $j \in \mathcal{N}_i^P$ 的归一化权重 α_{ij}^P ，接着就可以采用注意力机制将不同的元路径实例进行加权聚合了。将元路径实例向量进行加权聚合后，再通过一个激活函数得到输出。这里采用了多头注意力机制，有利于降低图网络的异质性所带来的高方差，即使用 K 个独立的注意力机制，将最后的输出拼接起来。将针对目标节点 i 对元路径 P 中所有元路径实例进行聚合后的向量表示记为 h_i^P ，其计算方法如下所示：

$$h_i^P = \left\| \sigma \left(\sum_{k=1}^K [\alpha_{ij}^P]_k \cdot h_{P(i,j)} \right) \right\| \quad (4-10)$$

其中 $[\alpha_{ij}^P]_k$ 表示元路径实例 $P(i, j)$ 在第 k 个注意力头上对目标节点 i 的归一化权重。

这里将多头注意力网络中头的个数设置为 8。

在得到目标节点 i 在元路径 P 上的向量表示后，需要聚合不同元路径的信息，同时也应该区分不同元路径对目标节点的重要性。如图 4-6 所示，同样以用户节点类型举例，针对目标节点 u_1 ，列举了路径规则分别为 $U \xrightarrow{\text{friend}} U$ ， $U \xrightarrow{\text{rate}} M \xrightarrow{\text{rate}^{-1}} U$ 以及 $U \xrightarrow{\text{rate}} M \xrightarrow{\text{genre}} G \xrightarrow{\text{genre}^{-1}} M \xrightarrow{\text{rate}^{-1}} U$ 的元路径，三者将以不同的权重聚合成一个向量表示。

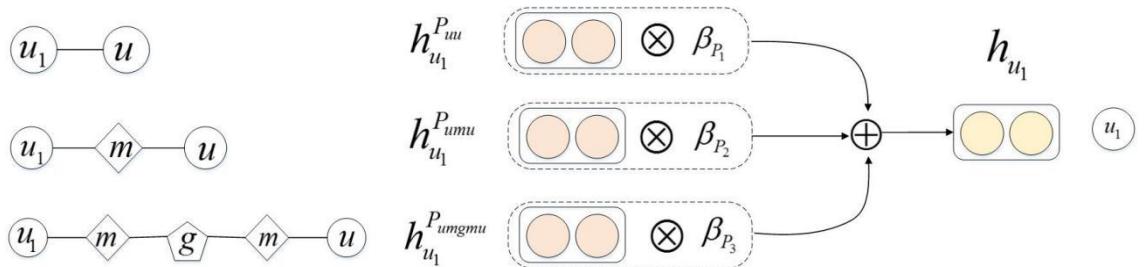


图 4-6 不同元路径信息融合示意图

Figure 4-6 Information fusion of different meta-paths

针对某一类型的目标节点 $i \in \mathcal{V}_A$ ，通过元路径实例信息聚合可以得到对应的向量表示集合： $\{h_i^{P_1}, h_i^{P_2}, \dots, h_i^{P_m}\}$ ，其中 m 为节点类型为 A 的元路径种类数量。鉴于不同的元路径在异质图中重要性是不同的，这里同样采用注意力机制进行信息聚合。首先针对每条元路径 $P_m \in \mathcal{P}_A$ ，对所有类型为 A 的节点在特定元路径下的节点向量进行转换，如式(4-11)所示：

$$s_{P_m} = \frac{1}{|\mathcal{V}_A|} \sum_{i \in \mathcal{V}_A} \tanh(M_A \cdot h_i^{P_m} + b_A) \quad (4-11)$$

其中 $M_A \in \mathbb{R}^{d_m \times d'}$ 和 $b_A \in \mathbb{R}^{d_m}$ 是学习参数。然后使用注意力机制聚合特定元路径下

的节点 i 的特征向量，计算方式如下所示：

$$e_{P_m} = q_A^T \cdot s_{P_m} \quad (4-12)$$

$$\beta_{P_m} = q_A^T \cdot s_{P_m} \quad (4-13)$$

$$h_i = \sum_{p \in \mathcal{P}_A} \beta_p \cdot h_i^p \quad (4-14)$$

其中 $q_A \in \mathbb{R}^{d_m}$ 表示针对 A 类型节点所赋予的参数注意力向量， β_{P_m} 用来衡量元路径 P_m 对 A 类型节点的影响力，使用该注意力系数对节点 i 的所有特定类型向量进行加权求和，最终得到特定类型的节点嵌入表示。

针对特定元路径 P，特定类型节点会存在多个元路径实例，在经过元路径实例函数编码后得到每条原路径实例的向量表示，接着采用多头注意力机制聚合元路径 P 中的不同元路径实例 $P(i, j)$ ，能够得到目标节点 i 在元路径 P 上的向量表示 h_i^P ，随后采用注意力机制对不同元路径的信息进行聚合，最终得到融合了元路径内部实例信息以及不同元路径信息的嵌入表示。

4.1.3 模型训练

通过上述信息聚合，可以得到用户和项目的表征，分别记为 h_u 和 h_v ，模型预测排序分数 \hat{y}_{uv} 计算过程如式(4-15)所示：

$$\hat{y}_{uv} = h_u^T \cdot h_v \quad (4-15)$$

这里得到的结果并不是用户对项目的具体预测评分，而是一个排序分数，用来生成个性化排序列表。

数据集中，存在与用户直接交互过的项目，通过邻接矩阵得到还没有交互记录的项目，将两者分别称作正向反馈项目和负向反馈项目，这里模型采用最大化两者排序分数之差来优化参数，优化目标函数如式(4-16)所示：

$$\mathcal{L} = \sum_{(u, v, v')} -\log \sigma(\hat{y}_{uv} - \hat{y}_{uv'}) \quad (4-16)$$

其中， \hat{y}_{uv} 和 $\hat{y}_{uv'}$ 分别表示模型计算出的用户对正向反馈和负向反馈项目的预测排序分数，得到两者之差再经过激活函数和对数运算，取其相反数得到 loss 值，模型通过最小化该损失值来优化参数。

4.2 评估指标

在 Top-N 列表推荐任务场景中，模型目的是识别用户感兴趣的项目，当在模

型给出的推荐列表中出现越多用户喜欢的项目时，说明模型效果越好，其实可以看作是对项目的标签分类问题。一般采用 Precision、Recall、NDCG 等作为评估指标，在认识这些评估指标前介绍下混淆矩阵，具体示意图如下：

		预测类别	
		1	0
真实类别	1	True Positive (TP)	False Negative (FN)
	0	False Positive (FP)	True Negative (TN)

图 4-7 混淆矩阵
Figure 4-7 Confusion matrix

如图 4-7 所示，其中 1 表示正例或阳性，0 表示负例或阴性，TP 表示预测类别为正例真实也为正例预测正确，TN 表示预测类别为负例真实也为负例预测正确，FP 表示预测类别为正例真实却是负例预测错误，FN 表示预测类别为负例真实却是正例预测错误。

放在列表推荐任务场景中，测试集中针对每个用户以一定比例取出了用户实际交互过的项目，这部分看作用户真正喜欢的项目集合 I_u ，而模型会根据计算推荐一串它认为用户喜欢的项目序列，这里记为 \hat{I}_u 。

(1) Precision

准确率表示预测为正例的样本集合中，真正类别为正例所占的比重，其计算方法如下式所示：

$$Precision = \frac{TP}{TP + FP} \quad (4-17)$$

放在列表推荐任务场景中，该指标描述的是用户真正喜欢的项目所占模型推荐整个项目序列的比重，可用式(4-18)描述：

$$Precision_u = \frac{|I_u \cap \hat{I}_u|}{|\hat{I}_u|} \quad (4-18)$$

(2) Recall

召回率表示在所有真实类别为正样本的集合中，准确预测出正例所占的比重，其计算方式如下：

$$Recall = \frac{TP}{TP + FN} \quad (4-19)$$

在列表推荐场景中，也就是模型预测出用户真正喜欢的项目占用户实际所有喜欢项目集合的比重，可用式(4-20)描述：

$$Recall_u = \frac{|I_u \cap \hat{I}_u|}{|I_u|} \quad (4-20)$$

(3) NDCG

NDCG (Normalized Discounted Cumulative Gain) 表示归一化折损累计增益，可以用来衡量排序的优劣。其中 CG (Cumulative Gain) 表示累计增益，表示排序结果中对应位置得分的总和，计算方式如下式所示：

$$CG_p = \sum_{i=1}^p rel_i \quad (4-21)$$

其中， rel_i 表示第 i 个位置上的得分值， p 表示列表长度。如果某一个排序结果为 3、2、1、3、2，则 $CG = 3 + 2 + 1 + 3 + 2 = 11$ 。这种方式没有考虑到每个分值的位置关系，在实际情况中比较希望得分值高的排在前面，所以将得分值按位置折算一下，这就是折损累计增益 (DCG) 的思想，计算过程如式(4-22)所示：

$$DCG_p = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2 i} \quad (4-22)$$

符号含义跟上面一样，就是按照位置给分值打了对应折扣，同样是上述排序的例子，折损累计增益 $DCG = 3 + 2 + 0.63 + 1.5 + 0.86 = 7.99$ 。NDCG 是归一化的折扣累计增益，计算方式如下式所示：

$$NDCG_p = \frac{DCG_p}{IDCG_p} \quad (4-23)$$

其中， $IDCG$ 表示 ideal DCG，即理想的 DCG，上述例子的理想排序结果应该是 3、3、2、2、1，还是按 DCG 的方式计算，得到 $IDCG = 3 + 3 + 1.26 + 1 + 0.43 = 8.69$ ，然后就可以得到 $NDCG = 7.99 / 8.69 = 0.91$ 。

4.3 实验设置

在本文实验中，相关实验环境信息如下所示：

编程语言：Python 3.5

神经网络框架：Pytorch 1.2.0

图工具：DGL 0.3.1

硬件配置：显存为 8G 的特斯拉 P4GPU，系统为 Ubuntu16.04，部分实验在搭载 8G 内存 CPU 的 windows7 系统和 Google Colab 上完成的。

科学运算库: numpy, pandas, pickle, scipy, argparse 等。

主要参数设置如表 4-2 所示:

表 4-2 主要参数设置
Table4-2 Main parameter settings

参数名称	参数值
num_ntype	3
dropout_rate	0.5
learn_rate	0.005
num_heads	8
attn-vec-dim	128
etypes_lists (边类型)	<pre>[[[0, 1], [0, 2, 3, 1], [None]], [[1, 0], [2, 3], [1, None, 0]]]</pre>

4.4 实验结果分析

为了观察模型的推荐性能, 本节将与相关工作进行实验对比。这里选取的是 MCRec 模型, 该模型是学者们提出的基于异质信息网络的个性化推荐算法。为解决数据集的稀疏问题, MCRec 引入了用户的社交关系以及项目的相关属性信息, 并将其建模为异质信息网络, 同时采用元路径进行信息挖掘, 但是它在对用户和项目进行表征时先是单独将用户和项目的嵌入向量表示出来, 在处理元路径信息时利用了随机采样策略选取不同的元路径实例, 经过卷积池化等操作后得到向量表示, 最后将三者嵌入表示拼接到一起。

MCRec 模型跟本文工作存在一些差异。MCRec 模型在聚合元路径实例信息时只考虑了节点的特征信息, 而本文在考虑节点特征信息的同时还考虑了边的特征信息, 使得节点的内容和位置信息都得以保留。而且前者没有区分元路径内部不同实例的贡献信息, 而本文采用了多头注意力机制对元路径内部实例信息进行聚合, 同样也对不同路径规则的元路径信息做了加权聚合。前者在进行向量表征时分别得到用户、项目以及基于元路径的上下文语义向量后将它们简单拼接, 再送入多层次感知机。而本文基于图神经网络将所有节点特征向量映射到统一特征空间中, 针对目标节点, 首先将与之相连的邻近节点信息和边信息汇聚到一起, 随后进行不同元路径实例以及不同元路径信息的聚合, 最终得到该节点的嵌入表示, 从而得到的用户和项目表征信息更丰富, 与上下文信息相关性更强。

(1) 整体对比实验

本文分别在 MovieLens 和 LastFM 数据集上进行了实验，实验结果如下所示：

表 4-3 相关工作在 MovieLens 上的实验对比
Table 4-3 Performance of MAGNN-Rec and baseline on MovieLens

Metrics		MCRec	MAGNN-Rec _{att}	MAGNN-Rec _{rot}	MAGNN-Rec
Precision	Top@10	0.3443	0.3861	0.4098	0.4142
	Top@20	0.3425	0.3938	0.3943	0.3950
Recall	Top@10	0.2224	0.2473	0.2658	0.2695
	Top@20	0.2769	0.3097	0.3128	0.3136
NDCG	Top@10	0.6932	0.7795	0.7913	0.7920
	Top@20	0.6773	0.7605	0.7617	0.7624

表 4-4 相关工作在 LastFM 上的实验对比
Table 4-4 Performance of MAGNN-Rec and baseline on LastFM

Metrics (Top@10)	MCRec	MAGNN-Rec _{att}	MAGNN-Rec _{rot}	MAGNN-Rec
Precision	0.4807	0.5078	0.5193	0.5264
Recall	0.5068	0.5206	0.5324	0.5377
NDCG	0.8526	0.8779	0.8801	0.8837

从表 4-3 和表 4-4 中实验结果中可以看出，相比 MCRec，MAGNN-Rec 在两个数据集上的各个推荐指标都有所提升。其中，在 K 取 10 的情况下，相比基线模型，在 MovieLens 数据集上 Precision、Recall、NDCG 分别提升了 20.30%、21.17% 和 14.25%，在 LastFM 数据集上三个指标分别提升了 9.50%、6.09% 和 3.64%。

另外，为了探寻不同因素对实验结果的影响，设置了两组对比实验。其中 MAGNN-Rec_{att} 表示只针对不同元路径实例做了带注意力机制的加权聚合操作，并没有对每一条元路径实例采用关系旋转的方式编码。MAGNN-Rec_{rot} 表示采用了关系旋转的编码方式，但没有对多条元路径实例进行加权聚合操作。通过实验结果可以看出，相比引入的注意力机制操作，所使用的关系旋转编码方式对提升模型性能帮助更大一些。

再对两个数据集的实验结果进行比较，可以看出模型在 MovieLens 数据集上各个指标上的改善性能要好一些，两个数据集的特征在 3.3 节中已经分析过，MovieLens 数据集中有 47.5% 的用户观影数量在 30 以下，有 68% 的用户观影数量在 60 以下，而在 LastFM 数据集中有 96.7% 的用户每个人至少有 50 个交互项目。这也就意味着 LastFM 数据集中每个用户的项目分配数量并不算稀少而且大都一致，

而 MovieLens 数据集中大多数用户的交互数据是很稀疏的，这就使得引入的属性以及社交信息还有对用户项目的表征方式发挥更大的作用，也证明了本文所使用的信息融合方式能够更有效地完成用户和项目表征。

(2) 局部信息利用对比实验

由于在 MCRec 中对元路径实例进行处理时，是通过随机游走精心挑选了 K 个元路径实例参与后面的信息聚合。在每一个元路径实例内部，首先计算连续节点对之间的相似度，对这些相似度取均值，根据该结果对不同元路径实例进行排序，最后选取平均相似度较高的 Top-K 条元路径实例。而本文在处理这部分信息时考虑了全部的元路径实例，并对他们进行加权聚合。为了探寻不同信息利用率对模型性能的影响，设计了这样一组对比实验，首先将 MovieLens 数据集以 2:1:7 的比例分割分别当作训练、测试、验证集，在两个模型上进行实验，接着再将以 7:1:2 比例分割得到的数据集在 MCRec 上单独进行实验。实验结果如下表所示：

表 4-5 局部信息利用对比试验
Table 4-5 Partial information utilization comparative experiment

Metrics (Top@10)	MCRec	MCRec _{all}	MAGNN-Rec
Precision	0.2974	0.3443	0.3716
Recall	0.2004	0.2224	0.2536
NDCG	0.6415	0.6932	0.7687

在表 4-5 中，MCRec 和 MAGNN-Rec 两栏表示两者都只利用了整个数据集的 20% 对模型进行训练得到的实验结果，而且 MCRec 模型在 20% 数据利用的基础上又进行了 Top-K 元路径实例信息的筛选，总体使用的信息量要少一些，MCRec_{all} 一栏是模型使用了整体数据集的 70% 进行训练得到的实验结果，在此基础上也进行了 K 条实例信息的筛选。通过对 MCRec 和 MAGNN-Rec 两栏实验结果，在使用整体占比相同数据量对模型进行训练时，MAGNN-Rec 有更好的效果，原因可能有两个：1) 由于没有进行实例信息筛选，相比下使用了更多的信息对模型进行训练；2) 使用了特殊的元路径实例编码方式和基于注意力的信息融合策略。再对比 MCRec_{all} 和 MAGNN-Rec 这组实验，经统计 MCRec_{all} 实际用到的元路径实例数量约为 667 万条，而在 MAGNN-Rec 模型中实际参与训练的元路径实例约为 168 万条，在使用更少数据对模型训练的情况下，MAGNN-Rec 仍然表现出更好的效果，说明在性能提升方面本文改进的两个点发挥了主要的作用。在信息取舍方面，MCRec_{all} 利用了随机游走相关方法对信息进行优化筛选，而在这组对比实验中 MAGNN-Rec 只是随机挑选了部分数据，从这个角度来考虑实验条件对本文模型更不利，但也更能说明本文算法工作改进的有效性。

4.5 冷启动对比实验

本文致力于利用用户的社交信息等异质信息和有效的信息融合方法对用户项目进行表征，从而改善个性化推荐中的冷启动问题，因此本节将设计相关实验观察模型的冷启动改善能力。

通过分析可知，LastFM 数据分布较为平均，而 MovieLens 数据集大部分用户的数据交互都比较稀疏，所以 MovieLens 适用于做冷启动改善实验，实验结果如图 4-8 所示。

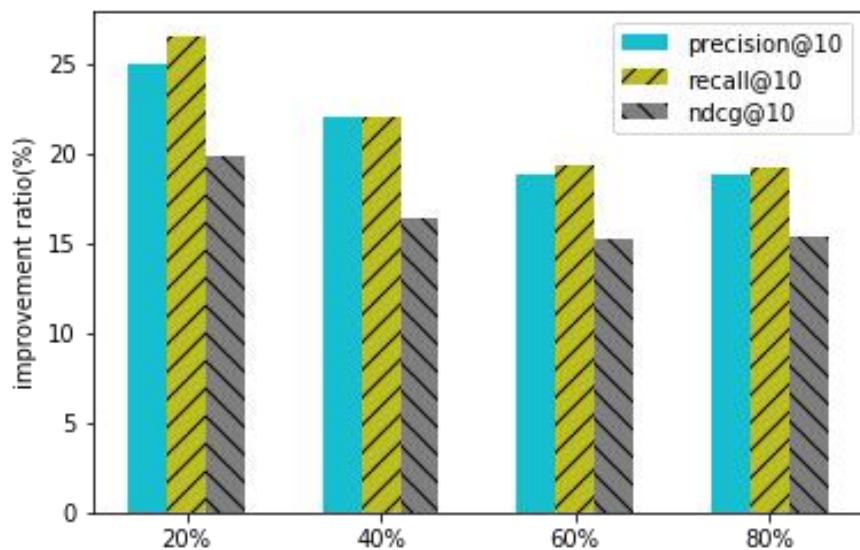


图 4-8 冷启动改善结果

Figure 4-8 Improvement of cold start problem

这里首先将整个数据集分为五等份，取出其中一份作为测试集，剩下的分别划分为占原始数据集 20%、40%、60%、80% 的子数据集，将这四个数据集分别放在 MCRec 和 MAGNN-Rec 模型上进行实验，将 Top-N 设置为 Top@10。如图 4-8 所示，相比基线模型，MAGNN-Rec 在各个稀疏度的子数据集上性能都有所提升，且随着数据稀疏性的增强，模型的改善能力也越来越好，可见模型有较好的冷启动改善能力。

4.6 本章小结

本章主要介绍了基于注意力机制的异质图网络推荐算法 MAGNN-Rec，相比现有基于异质信息网络的 MCRec 模型，本章实现了对每一条元路径实例所包含节点的内容和位置信息提取，并引入了注意力机制融合不同的元路径实例信息，从而更好地对用户和项目进行表征。

首先介绍了模型的整体结构，包括节点信息转换、元路径内部实例信息聚合、

不同元路径信息聚合等部分。鉴于异质图中存在多种不同类型的对象节点，通常它们含有不同的维度或存在于不同的特征空间，所以需要先对它们进行线性转换映射到同一隐藏向量空间。接着对目标节点进行元路径实例信息聚合，采用了关系旋转的编码方式将在同一条元路径实例上与目标节点相连的邻近节点信息聚合为一个向量，接着采用多头注意力机制聚合不同的元路径实例信息，随后采用注意力机制聚合不同元路径信息，从而分别得到用户和项目的表征，再进行相关运算得到预测排序评分，通过最大化用户对正向反馈和负向反馈项目的排序评分之差优化模型中相关参数，进行模型训练。

接着对模型进行实验验证。针对 MovieLens 数据集和 LastFM 数据集，采用列表推荐任务中常用的 Precision、Recall、NDCG 作为评估指进行实验验证。其中，在数据分布较为平均的 LastFM 数据集上的准确率、召回率和归一化折损累计增益分别达到了 0.52、0.53 和 0.88，相比基线模型 MCRec 分别提升了 9.50%、6.09% 和 3.64%，在多数用户项目分布较为稀疏的 MovieLens 数据集上三个指标分别达到了 0.41、0.26 和 0.79，相比基线模型 MCRec 分别提升了 20.30%、21.17% 和 14.25%。接着设计实验观察模型对冷启动问题的改善能力，实验结果表明随着数据集稀疏性的增强，模型表现出的改善能力越强，其中在最稀疏的数据集上三个指标分别提升了 24.94%、26.54% 和 19.82%。

5 总结与展望

本章主要总结了本文的工作内容以及主要的贡献成果，同时对未来的工作进行了展望。在进行工作总结时，先阐述了目前个性化推荐领域还存在的问题，还有针对这些问题所采用的解决方法，接着对本文主要工作内容和关键性成果进行了总结。同时分析了本文工作还存在的问题，并阐述了针对该问题可以考虑的改进方向，还有针对不同的现实场景需求推荐系统所面临的问题，以此为依据对未来的发展方向进行展望。

5.1 本文工作总结

本文主要针对个性化推荐系统中的数据稀疏以及冷启动问题设计了一种基于异质信息的图神经网络推荐框架。为了提高推荐性能，需要解决两个关键问题，一个是如何利用稀疏的数据集充分挖掘相关信息，还有一个是如何更好的对用户和项目进行表征。针对数据稀疏性问题，本文引入用户的社交信息以及项目的相关属性信息并将其建模为异质信息网络，充分利用有利于推测用户偏好的附加信息。针对用户和项目表征问题，设计不同的元路径形式，针对目标节点，基于图神经网络聚合不同层次的信息。在用户项目所在不同路径规则下的元路径实例内部，考虑到节点的内容和位置信息，同时还考虑到不同元路径实例对目标节点不同的重要性，因此采用了多头注意力机制汇聚每一种元路径中不同的元路径实例信息，随后再次采用注意力机制聚合不同元路径的信息。通过上述过程能够分别得到用户和项目的表征，注意到所有的信息聚合过程都是针对目标节点完成的，以此种方式得到的用户项目嵌入表示与元路径上下文内容关系更密切。本文的主要工作及贡献如下：

(1) 针对异质图中元路径信息利用不充分问题，加入边信息，设计了基于关系旋转的用户项目所在元路径实例信息聚合方式，来充分对用户和项目进行表征。将通过元路径与目标节点直接或间接相连的每一个邻近节点信息和连边信息都融合到一起构成单一向量，同时保留了元路径内部节点的向量信息和位置信息，从而丰富用户和项目的嵌入表示。

(2) 针对不同的元路径实例引入注意力机制进行信息聚合。针对目标节点，每一种元路径下含有不同的元路径实例，且他们对目标节点的重要性是不同的，采用多头注意力机制汇聚不同来源的元路径实例信息，能够克服异质图所带来的不稳定性，同时有效聚合不同元路径内部实例信息。

(3) 在两个含有不同特征的数据集上进行 Top-N 列表推荐实验，相比基线模型，本文模型在各个评估指标上都有提升。其中，在数据分布较为平均的 LastFM 数据集上的准确率、召回率和归一化折损累计增益分别达到了 0.52、0.53 和 0.88，相比基线模型 MCRec 分别提升了 9.50%、6.09% 和 3.64%，在多数用户项目分布较为稀疏的 MovieLens 数据集上三个指标分别达到了 0.41、0.26 和 0.79，相比基线模型 MCRec 分别提升了 20.30%、21.17% 和 14.25%。并设计了冷启动对比实验，将数据集划分成不同稀疏度的子数据集，并与相关工作进行对比，实验结果表明，随着数据稀疏性的增强算法的提升效果越明显，其中在最稀疏的数据集上三个指标分别提升了 24.94%、26.54% 和 19.82%。

5.2 未来工作展望

为了缓解个性化推荐领域中由于数据稀疏造成的冷启动问题，本文主要提出了基于注意力机制的异质图网络推荐模型。在现实生活中的推荐场景中，用户的实时交互记录对提升推荐性能是很有价值的，比如说大型购物网站中用户每天都有可能对商品进行浏览和消费，用户的喜好也可能受周围环境的影响而改变。而且在异质信息网络中，人为定义的元路径形式不一定就是最有效的。因此，未来希望在以下两个方面进行探索研究：一方面是，每一个新的事件都有可能影响图中其他节点信息，面对用户项目动态的实时交互行为，如何在大规模异质图上进行动态的图构造是需要考虑的问题。另一方面，抛开人工定义不同的元路径方式，模型如果能够自动学习更多有意义的元路径形式，推荐性能应该会有提升。

参考文献

- [1] Han J. Mining heterogeneous information networks by exploring the power of links[C]// Mining heterogeneous information networks by exploring the power of links. International Conference on Discovery Science. Springer: 13-30.
- [2] Baluja S, Seth R, Sivakumar D, et al. Video suggestion and discovery for youtube: taking random walks through the view graph[C]// Video suggestion and discovery for youtube: taking random walks through the view graph. Proceedings of the 17th international conference on World Wide Web. 895-904.
- [3] Shi C, Hu B, Zhao W X, et al. Heterogeneous information network embedding for recommendation[J]. 2018, 31(2): 357-370.
- [4] Nandanwar S, Moroney A, Murty M N. Fusing diversity in recommendations in heterogeneous information networks[C]// Fusing diversity in recommendations in heterogeneous information networks. Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. 414-422.
- [5] Jiang Z, Liu H, Fu B, et al. Recommendation in heterogeneous information networks based on generalized random walk model and bayesian personalized ranking[C]// Recommendation in heterogeneous information networks based on generalized random walk model and bayesian personalized ranking. Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. 288-296.
- [6] Zhang Y, Ai Q, Chen X, et al. Joint representation learning for top-n recommendation with heterogeneous information sources[C]// Joint representation learning for top-n recommendation with heterogeneous information sources. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 1449-1458.
- [7] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations[C]// Deepwalk: Online learning of social representations. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 701-710.
- [8] Tang J, Qu M, Mei Q. Pte: Predictive text embedding through large-scale heterogeneous text networks[C]// Pte: Predictive text embedding through large-scale heterogeneous text networks. Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. 1165-1174.
- [9] Xu L, Wei X, Cao J, et al. Embedding of embedding (EOE) joint embedding for coupled heterogeneous networks[C]// Embedding of embedding (EOE) joint embedding for coupled heterogeneous networks. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. 741-749.
- [10] Dong Y, Chawla N V, Swami A. metapath2vec: Scalable representation learning for heterogeneous networks[C]// metapath2vec: Scalable representation learning for heterogeneous networks. Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 135-144.
- [11] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:13013781, 2013.
- [12] Shang J, Qu M, Liu J, et al. Meta-path guided embedding for similarity search in large-scale

- heterogeneous information networks[J]. arXiv preprint arXiv:161009769, 2016.
- [13] Fu T-y, Lee W-C, Lei Z. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning[C]// Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 1797-1806.
- [14] Chen H, Yin H, Wang W, et al. PME: projected metric embedding on heterogeneous networks for link prediction[C]// PME: projected metric embedding on heterogeneous networks for link prediction. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1177-1186.
- [15] Yin J, Guo Y, Chen Y. Heterogenous Information Network Embedding Based Cross-Domain Recommendation System[C]// Heterogenous Information Network Embedding Based Cross-Domain Recommendation System. 2019 International Conference on Data Mining Workshops (ICDMW). IEEE Computer Society: 362-369.
- [16] He X, Deng K, Wang X, et al. Lightgcn: Simplifying and powering graph convolution network for recommendation[C]// Lightgcn: Simplifying and powering graph convolution network for recommendation. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 639-648.
- [17] Hu B, Shi C, Zhao W X, et al. Leveraging meta-path based context for top-n recommendation with a neural co-attention model[C]// Leveraging meta-path based context for top-n recommendation with a neural co-attention model. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1531-1540.
- [18] Liu Y, Liang C, He X, et al. Modelling High-Order Social Relations for Item Recommendation[J]. 2020.
- [19] Yu J, Yin H, Li J, et al. Enhance social recommendation with adversarial graph convolutional networks[J]. 2020.
- [20] Chen C, Zhang M, Liu Y, et al. Social attentional memory network: Modeling aspect-and friend-level differences in recommendation[C]// Social attentional memory network: Modeling aspect-and friend-level differences in recommendation. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. 177-185.
- [21] Fan W, Ma Y, Li Q, et al. Graph neural networks for social recommendation[C]// Graph neural networks for social recommendation. The World Wide Web Conference. 417-426.
- [22] Wang X, Ji H, Shi C, et al. Heterogeneous graph attention network[C]// Heterogeneous graph attention network. The World Wide Web Conference. 2022-2032.
- [23] Xu Y, Zhu Y, Shen Y, et al. Learning shared vertex representation in heterogeneous graphs with convolutional networks for recommendation[C]// Learning shared vertex representation in heterogeneous graphs with convolutional networks for recommendation. IJCAI. 4620-4626.
- [24] Fan S, Zhu J, Han X, et al. Metapath-guided heterogeneous graph neural network for intent recommendation[C]// Metapath-guided heterogeneous graph neural network for intent recommendation. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2478-2486.
- [25] Wang X, Wang R, Shi C, et al. Multi-component graph convolutional collaborative filtering[C]// Multi-component graph convolutional collaborative filtering. Proceedings of the

- AAAI Conference on Artificial Intelligence.34: 6267-6274.
- [26] Van Meteren R, Van Someren M. Using content-based filtering for recommendation[C]// Using content-based filtering for recommendation. Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop.30: 47-56.
- [27] Baeza-Yates R, Ribeiro-Neto B. Modern information retrieval[M]. City: ACM press New York, 1999.
- [28] Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering[C]// Laplacian eigenmaps and spectral techniques for embedding and clustering. Nips.14: 585-591.
- [29] Luo D, Ding C H, Nie F, et al. Cauchy graph embedding[C]// Cauchy graph embedding. ICML.
- [30] Tenenbaum J B, De Silva V, Langford J C J s.A global geometric framework for nonlinear dimensionality reduction[J].2000,290(5500):2319-2323.
- [31] Roweis S T, Saul L K J s.Nonlinear dimensionality reduction by locally linear embedding[J].2000,290(5500):2323-2326.
- [32] Cao S, Lu W, Xu Q. Grarep: Learning graph representations with global structural information[C]// Grarep: Learning graph representations with global structural information. Proceedings of the 24th ACM international on conference on information and knowledge management. 891-900.
- [33] Ou M, Cui P, Pei J, et al. Asymmetric transitivity preserving graph embedding[C]// Asymmetric transitivity preserving graph embedding. Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 1105-1114.
- [34] Grover A, Leskovec J. node2vec: Scalable feature learning for networks[C]// node2vec: Scalable feature learning for networks. Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 855-864.
- [35] Wang D, Cui P, Zhu W. Structural deep network embedding[C]// Structural deep network embedding. Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 1225-1234.
- [36] Cao S, Lu W, Xu Q. Deep neural networks for learning graph representations[C]// Deep neural networks for learning graph representations. Proceedings of the AAAI Conference on Artificial Intelligence.30.
- [37] Hamilton W L, Ying R, Leskovec J J a p a.Inductive representation learning on large graphs[J]. arXiv preprint arXiv:170602216, 2017.
- [38] Bruna J, Zaremba W, Szlam A, et al.Spectral networks and locally connected networks on graphs[J]. arXiv preprint arXiv:13126203, 2013.
- [39] Dai H, Kozareva Z, Dai B, et al. Learning steady-states of iterative algorithms over graphs[C]// Learning steady-states of iterative algorithms over graphs. International conference on machine learning. PMLR: 1106-1114.
- [40] Li Y, Tarlow D, Brockschmidt M, et al.Gated graph sequence neural networks[J]. arXiv preprint arXiv:151105493, 2015.
- [41] Fu X, Zhang J, Meng Z, et al. MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding[C]// MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. Proceedings of The Web Conference 2020. 2331-2341.

- [42] Sun Z, Deng Z-H, Nie J-Y, et al. Rotate: Knowledge graph embedding by relational rotation in complex space[J]. arXiv preprint arXiv:190210197, 2019.

作者简历及攻读硕士学位期间取得的研究成果

一、作者简历

张纯，女，1997年7月生。2015年9月至2019年6月就读于河南理工大学物理与电子信息学院电子信息工程专业，取得工学学士学位。2019年9月至2021年6月就读于北京交通大学电子与信息工程学院电子与通信工程专业，研究方向是信息网络，取得工学硕士学位。攻读硕士学位期间，主要从事个性化推荐方面的研究工作。

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：

签字日期： 年 月 日

学位论文数据集

表 1.1： 数据集页

关键词*	密级*	中图分类号	UDC	论文资助
个性化推荐；冷启动；图神经网络；异质信息网络；注意力机制	公开			
学位授予单位名称*	学位授予单位代码*	学位类别*	学位级别*	
北京交通大学	10004	工学	硕士	
论文题名*	并列题名		论文语种*	
基于 HIN 的图神经网络社会推荐算法研究			中文	
作者姓名* 张纯		学号* 19125067		
培养单位名称*	培养单位代码* *	培养单位地址	邮编	
北京交通大学	10004	北京市海淀区西直门外上园村 3 号	100044	
学科专业* 电子与通信工程	研究方向* 信息网络	学制* 2	学位授予年* 2021	
论文提交日期 * 2021				
导师姓名* 郭宇春		职称* 教授		
评阅人 答辩委员会主席*		答辩委员会成员		
电子版论文提交格式 文本() 图像() 视频() 音频() 多媒体() 其他() 推荐格式: application/word; application/pdf				
电子版论文出版(发布)者	电子版论文出版(发布)地	权限声明		
论文总页数* 54 页				
共 33 项，其中带*为必填数据，为 21 项。				