

分类号

密级

中国地质大学（北京）
专业硕士学位论文

基于自监督学习的图神经网络协同过滤
推荐算法研究

研究生 万双玮 学 号 2104210035

专业学位类别 电子信息 专 业 计算机技术

校内导师 牛云云 产业导师 张永虹

学习方式 全日制

2024 年 5 月

**A Dissertation Submitted to
China University of Geosciences for Master of Professional
Degree**

**Research on collaborative filtering recommendation algorithm
based on self-supervised learning for graph neural networks**

Master Candidate: Shuangwei Wan

Professional Degree: Master of Engineering

Dissertation Supervisor: Prof. Yunyun Niu

Associate Supervisor: Yonghong Zhang

China University of Geosciences (Beijing)

声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得中国地质大学或其它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。

签 名：_____日 期：_____

关于论文使用授权的说明

本人完全了解中国地质大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。**延期公开的论文在到期公开后应遵守此规定。**

签 名：_____导师签名：_____日 期：_____

摘要

随着信息技术的迅猛发展和互联网的大量普及，人们接收并产生着大量信息数据，不再受限信息匮乏。然而，当今世界正在加速进入一个信息爆炸的大数据时代，造成了信息过载的问题，推荐系统随之而来。传统推荐系统在缓解信息过载问题的同时，也存在着不可忽视的局限性：一是数据集稀疏问题；二是冷启动问题；三是无法捕捉高阶信息问题。图神经网络推荐算法作为一种深度学习推荐算法，由传统推荐算法发展而来，在处理图结构数据和捕捉高阶信息方面具有巨大优势，从而实现更加精准的个性化推荐。由于推荐系统中的大部分信息本质上都具有图结构，图神经网络推荐算法在推荐系统领域广泛使用，成为热门的研究方向。然而，图神经网络推荐算法也面临着一些挑战，如网络层过度平滑、模型结构过于复杂以及噪声长尾问题。针对以上所述问题，提出以下两种算法：

(1) 基于自监督学习的简化图神经网络协同过滤算法。该算法将自监督学习算法与简化的图神经网络协同过滤算法相结合，提高了算法的鲁棒性，缓解了噪声长尾问题。通过去除图神经网络层之间的非线性并将结果函数折叠成单个线性变换来降低模型过度复杂问题，使模型易于训练，可扩展至大型数据集。通过计算压缩图上的网络嵌入来获得图中每个节点的嵌入的初始化，减轻稀疏交互数据的图卷积聚合操作中的过度平滑问题，能够更好地处理用户和项目之间的复杂交互关系，提供更加精确的个性化推荐。

(2) 基于自监督学习的邻居感知注意力图神经网络协同过滤算法。在过去的研究中，邻居之间隐含的相关性经常被忽略。该算法是在以往算法的基础上做出了改进，强调了用户-项目交互图上每个节点的邻居之间隐含相关性的重要性。通过捕获节点邻居之间的隐含相关性，用于推荐任务。具体来说，在基于自监督学习的简化图神经网络协同过滤算法的基础上，引入了一个邻居感知图注意力层来计算用户的邻居项目和项目的邻居用户的邻居感知注意力系数。通过对邻居之间的每个成对注意力函数的结果进行平均来计算注意力系数。使用邻居嵌入和注意力系数的线性组合作为每个节点的下一层嵌入，从而进一步堆叠更多的邻居感知图注意力层，来收集来自高阶邻居的有影响力的信号。有效地缓解了算法冷启动问题，从而提高了推荐算法的性能。

关键词：推荐算法，深度学习，图神经网络，协同过滤，自监督学习

Abstract

With the rapid development of information technology and the massive popularization of the Internet, people are receiving and generating large amounts of information data and are no longer limited by the lack of information. However, today's world is accelerating into an information explosion of big data era, causing the problem of information overload, recommender system follows. While alleviating the problem of information overload, traditional recommender systems also have limitations that cannot be ignored: first, the problem of sparse dataset; second, the problem of cold-start; and third, the problem of not being able to capture higher-order information. Graph neural network recommendation algorithm, as a deep learning recommendation algorithm, developed from traditional recommendation algorithms, has a huge advantage in processing graph-structured data and capturing higher-order information, so as to achieve more accurate personalized recommendation. Since most of the information in recommender systems essentially has a graph structure, graph neural network recommendation algorithms are widely used in the field of recommender systems and have become a popular research direction. However, graph neural network recommendation algorithms also face some challenges, such as over-smoothing of network layers, over-complex model structure, and noisy long-tail problem. The following two algorithms are proposed to address the above mentioned problems:

(1) Simplified graph neural network collaborative filtering algorithm based on self-supervised learning. The algorithm combines a self-supervised learning algorithm with a simplified graph neural network collaborative filtering algorithm, which improves the robustness of the algorithm and alleviates the noisy long-tail problem. The problem of model over-complexity is reduced by removing nonlinearities between graph neural network layers and collapsing the resultant function into a single linear transformation, making the model easy to train and scalable to large datasets. Initialization of the embedding of each node in the graph is obtained by computing the network embedding on the compressed graph to mitigate the over-smoothing problem in the graph convolutional aggregation operation for sparse interaction data, which can better handle the complex interactions between the user and the item, and provide more accurate personalized recommendations.

(2) Self-supervised learning based collaborative filtering algorithm for neighbor-aware attention graph neural networks. In past research, the implicit correlation between neighbors is often ignored. This algorithm is an improvement on previous algorithms, emphasizing the importance of the implicit correlation between neighbors of each node on the user-item interaction graph. By capturing the implicit correlation between node neighbors, it is used for recommending tasks. Specifically, based on a simplified graph neural network collaborative filtering algorithm based on self-supervised learning, a neighbor-aware graph attention layer is introduced to compute the neighbor-aware attention coefficients of the user's neighboring items and the item's neighboring users. The attention coefficients are computed by averaging the results of each pairwise attention function between neighbors. Use the linear combination of neighbor embeddings and attention coefficients as the next layer of embeddings for each node, thus further stacking more neighbor-aware graph attention layers to collect influential signals from higher-order neighbors. The algorithm cold-start problem is effectively mitigated, thus improving the performance of the recommendation algorithm.

Keywords: Recommendation algorithms, Deep learning, Graph neural networks, Collaborative filtering, Self-supervised learning

目 录

摘要	1
1 引言	1
1.1 研究背景及意义	1
1.2 国内外研究现状	3
1.2.1 协同过滤推荐算法研究现状	4
1.2.2 基于图神经网络推荐算法研究现状	5
1.3 当前研究不足	6
1.4 研究内容及论文结构	8
1.4.1 研究内容	8
1.4.2 论文结构	8
2 相关理论及其介绍	10
2.1 推荐系统相关理论及介绍	10
2.1.1 推荐系统概述	10
2.1.2 协同过滤推荐算法	10
2.1.3 推荐算法的评价指标	11
2.2 图神经网络相关理论和技术	12
2.2.1 图神经网络基本理论	12
2.2.2 基于图神经网络的协同过滤推荐模型	14
2.3 自监督学习	17
2.3.1 监督学习、半监督学习和无监督学习	17
2.3.2 自监督学习	17
2.3.3 自监督图学习过程	18
2.4 注意力机制	18
2.4.1 注意力机制计算过程	19
2.5 本章小结	20
3 基于自监督学习的简化图神经网络协同过滤算法	21
3.1 模型设计与描述	21
3.1.1 输入模块设计	23

3.1.2 数据增强模块设计	24
3.1.3 图神经网络编码器设计	25
3.1.4 目标优化模块设计	27
3.2 实验设计	28
3.2.1 实验设置	28
3.2.2 对比算法	29
3.2.3 参数设置	30
3.3 实验结果与分析	30
3.3.1 SSGCF 模型训练 Loss 曲线	30
3.3.2 SSGCF 的 λ_1 参数实验	31
3.3.3 SSGCF 与 LightGCN 在不同传播层的性能比较	31
3.3.4 SSGCF 噪声实验	32
3.3.5 对比实验	33
3.4 本章小结	34
4 基于自监督学习的邻居感知注意力图神经网络协同过滤算法	35
4.1 模型设计与描述	36
4.1.1 图神经网络编码器设计	37
4.1.2 目标优化模块设计	40
4.2 实验设计	41
4.2.1 实验设置	41
4.2.2 对比算法	41
4.2.3 参数设置	42
4.3 实验结果与分析	42
4.3.1 模型训练 Loss 曲线与 Recall 曲线	42
4.3.2 NASGCF 的 λ_1 参数实验	43
4.3.3 NASGCF 邻居数实验	44
4.3.4 NASGCF 不同变体的消融实验	45
4.3.5 对比实验	46
4.4 本章小结	47

5 结论与展望	49
5.1 论文结论	49
5.2 未来展望	50
参考文献	51
致谢	57
附录	59

1 引言

1.1 研究背景及意义

随着信息技术和互联网的迅猛发展和大量普及，尤其是移动互联网的兴起，信息数据为人们的工作学习、生产生活以及社交娱乐提供了极大的便利。人们每天都徜徉在信息世界的海洋中，通过电脑、移动设备和机器传感器接收并产生着大量数据，这些海量的数据为人们提供着方便，让人们不再受限于信息匮乏的问题中。然而，当今世界正在加速进入一个信息爆炸的大数据时代，面临数量庞大，纷繁复杂的数据时，人们接受、处理和利用信息的能力是有限的，难以寻找到有价值的信息，这就造成了信息过载^{[1][2]}的问题。

信息过载问题的出现意味着人们在面对海量数据时，难以有效地筛选和处理这些信息，导致有用信息被淹没在大量冗余数据中。无论是在工作中需要迅速找到关键数据，还是在日常生活中寻找所需的信息，信息过载都会带来困扰和压力。这种情况不仅影响了个体的决策效率和生活质量，也对企业和社会的整体生产力和创新能力产生了负面影响。为了应对信息过载的难题，谷歌、必应、雅虎等主流互联网搜索引擎工具诞生，在某种程度上缓解了这个问题。但是，搜索引擎的弊端也是显而易见的。当用户存在疑问却不知如何搜索，就不能找到想要的信息。在这种背景下，推荐系统应运而生。

上世纪 90 年代，首个推荐系统 Tapestry^[3]诞生。推荐系统的产生源于对用户需求不明确的情况下，通过技术手段从大量信息中找到用户感兴趣的内容，从而提高信息利用效率。推荐系统综合用户信息、项目信息以及用户对项目的历史操作交互行为，利用推荐算法构建个性化的用户兴趣模型，从而为用户提供准确的个性化推荐。推荐系统已经深刻影响着人们的日常生活，比如，淘宝等电商平台会根据你的浏览点击商品记录和与你有相似购物习惯的用户，为你推荐你感兴趣的商品；抖音等短视频平台会根据你的观看时长，点赞评论收藏和搜索记录，为你推荐你感兴趣的短视频；腾讯 QQ 等社交网络平台会根据与你拥有共同好友的人或者附近的人，为你推荐可能认识的人；网易云音乐等音乐平台会对歌曲贴上音乐类型和音乐国别等标签，通过分析用户听过的歌，为用户推送每日歌曲推荐等等。这些推荐系统取得了令人印象深刻的效果，有效地缓解了信息过载问题，为互联网公司带来巨大的收益，为人们的生活提供了极大的便利。

推荐算法是推荐系统的核心内容，传统的协同过滤^[4]作为推荐算法中的经典算法，仍然是当今最流行的推荐算法，研究人员对协同过滤算法的研究产生了丰厚的成果^[5]，在推

荐系统中应用得极其广泛和成功。然而，传统的协同过滤算法也存在着局限性：（1）数据集稀疏问题：用户对项目的评价和反馈数据通常呈现出一种稀疏性，即大多数用户只与少数项目进行了交互，而且很少对其他项目进行评估或反馈。这种数据的稀疏性使得推荐算法所能利用的有效信息减少，导致了算法性能的下降。稀疏数据不仅限制了推荐系统的准确性，还增加了推荐结果的不确定性，降低了用户对推荐结果的满意度。（2）冷启动问题：在面对新用户或新项目时，传统的推荐算法面临着冷启动问题。对于新用户来说，由于缺乏历史行为数据，推荐系统很难准确地了解他们的兴趣和偏好。同样，对于新项目而言，由于缺乏被用户评价的历史数据，推荐系统无法准确地评估其特征和与用户兴趣的匹配程度。因此，在缺乏足够信息的情况下，传统推荐算法往往无法提供令用户满意的推荐结果，从而影响了推荐系统的用户体验和效果。（3）传统的推荐算法通常只能捕捉用户和项目之间的简单关系，难以捕捉复杂的高阶关系和交互模式。例如，传统的协同过滤算法主要依赖于用户或项目之间的相似性来进行推荐，而忽略了更深层次的交互模式和隐含的语义信息。这种局限性导致了在处理大规模、复杂数据集时性能的限制，无法满足用户对个性化推荐的需求。因此，如何有效地捕捉用户和项目之间的高阶关系和复杂交互模式，成为推荐系统研究和应用中的重要挑战。这些局限性推动了研究者不断探索新的推荐算法，特别是基于深度学习的推荐算法，来克服传统推荐算法的局限性并提高推荐系统的性能。深度学习^[6]近些年来，在人脸识别^[7]、语音识别^[8]、自然语言处理^[9]，自动驾驶^[10]等领域广泛应用并且取得了显著的成绩的同时，在推荐系统领域的应用也如雨后春笋般崭露头角^[11]。

图是一种常见的数据结构，是节点以及这些节点之间关系的集合。人们日常生活中的数据可以通过图数据结构进行表示，图的强大之处在于对数据的理解能力和认知能力，考虑节点之间的关系以及通用性，可以很自然的表达现实世界中的实体及其关联关系，广泛应用于机器学习和数据挖掘等领域。包括社会科学：如社交网络^[12]、社区发现^{[13][14]}、自然科学中的物理系统^{[15][16]}和蛋白质相互作用网络^[17]，以及知识图谱^[18]和许多其他研究领域^[19]。图神经网络是一种神经模型，通过在图节点之间传递消息来获取图的依赖关系，将神经网络方法拓展到了图数据计算领域，采用嵌入传播以迭代地聚合邻域嵌入，通过堆叠传播层，每个节点都可以访问高阶邻居的信息。根据 Wu 等人^[20]的研究，图神经网络可以分为四类：循环图神经网络、图卷积网络、图自编码器和时空图神经网络。循环图神经网络 (Recurrent graph neural networks, RecGNN)^[21]是图神经网络的前驱，运用了类似循环神经网络的思想，采用了循环神经网络结构来学习节点表示，假设图中的节点会不断与其邻居交换信息，直

到达到稳定状态，为后续图卷积神经网络的研究提供了启示。图卷积网络 (Convolutional graph neural networks, GCN)^[22]是图神经网络的开山之作，将图像处理中的卷积操作简单的用到图结构数据处理中来。图自编码器 (Graph Autoencoders)^[23]是无监督学习框架，将节点或图编码到潜在向量空间并从编码信息中重建图数据，用于学习网络嵌入和图生成分布，解决了 GCN 不能处理大图和训练时必须要知道整个图结构信息的问题。时空图神经网络 (Spatial-temporal graph neural networks, STGNNs)^[24]旨在从时空图中学习隐藏模式，关键思想是同时考虑空间依赖和时间依赖。图神经网络在各种应用情景中扮演越来越重要的角色，例如作为交通速度预测^[25]，驾驶员机动预期^[26]和人类动作识别^[27]。这些图神经网络算法在许多深度学习任务中展示了突破性的表现，取得令人信服的结果。

图神经网络在推荐系统中的应用得到了越来越多的关注和应用，主要因为推荐系统中的数据往往具有图结构的特点。相比传统的推荐算法，图神经网络能够更好地捕捉数据之间的复杂关系和拓扑结构，从而提高了推荐的准确性和个性化程度。在处理复杂的用户关系或数据时，传统模型往往面临着限制，而图神经网络通过学习图结构中的信息，能够更好地解决这些问题。此外，传统的神经网络模型往往无法充分编码数据中的高阶信息，而图神经网络通过多层次的信息传递和聚合，能够更好地捕捉数据的高阶特征，提高了模型的性能和泛化能力。因此，图神经网络算法在推荐系统领域被广泛认为是一种先进的推荐算法，为推荐系统的发展带来了新的思路和方法。当前研究中的图神经网络推荐算法在解决复杂推荐问题的同时，也面临着一些挑战和问题。其中，冷启动问题是指对于新用户或新项目，由于缺乏历史数据，传统的图神经网络模型难以提供有效的推荐。数据集稀疏问题则指用户对项目的评价和反馈数据通常是稀疏的，这导致推荐算法能够使用的有效数据较少，影响了算法的性能。另外，过度平滑问题是指当使用多层图神经网络进行学习时，节点的隐层表示会趋向于收敛到同一个值，导致模型性能急剧下降。此外，由于图神经网络模型的复杂性，训练和推断的计算成本较高，模型过于复杂也成为一个问题。最后，数据中存在的噪声干扰也会影响模型的性能和稳定性。针对这些问题，需要对图神经网络协同过滤推荐算法进行深入探索和研究，并提出相应的解决方案进行改进。。

1.2 国内外研究现状

随着互联网的发展，国内外研究人员对于推荐系统的研究在学术界和工业界取得了瞩目的成绩。推荐系统广泛应用于生产生活中，提高了人们的效率，产生了巨大的经济价值。本节主要介绍协同过滤算法和基于图神经网络的协同过滤推荐算法研究现状。

1.2.1 协同过滤推荐算法研究现状

协同过滤推荐算法，是一种通过收集不同兴趣爱好相似的用户信息来对某一用户的兴趣进行自动预测的方法，在推荐系统领域仍然占据重要的地位。协同过滤的定义分类方式十分宽泛，按照一种被大多数人认可的分类方式，协同过滤算法可以分为：传统的协同过滤方法和基于深度学习的协同过滤算法。基于邻域的方法、隐语义方法以及基于图的随机游走方法被定义为传统的协同过滤方法。在基于邻域的方法中，Herlocker 等人^[28]根据实证研究的结果，提供了一组建议来指导基于邻域的预测系统的设计，将基于邻域的预测方法分为相似性计算、邻居选择和评级组合三个部分。随后，Robert 等人^[29]从数据中去除了“全局效应”，计算插值权重作为一个优化问题的全局解决方案，从而增强了基于邻域的方法，显著提高了预测精度。Koren 等人提出的隐语义模型^[30]，引入了基于 SVD 的潜在因素模型的扩展，通过将隐式反馈集成到模型中来提高精度。此外，推导出一个将邻域模型和潜在因素模型结合起来的综合模型，提高了模型性能。Yildirim 等人^[31]提出的基于图的随机游走方法，能够捕捉图中的隐含信息，具有较好的自适应性，适合用于多种任务，有效地解决了数据稀疏的问题。

近些年来，随着深度学习在推荐算法中广泛应用，基于深度学习的协同过滤算法取得了巨大的成功。基于深度学习的协同过滤算法可以分为基于表示学习的算法和基于匹配方法学习的算法。在基于表示学习的算法中，用户和项目分别通过神经网络生成各自的嵌入向量，即表示向量。将其作为中间产物，然后再通过内积等交互函数得到匹配分数，进行排序推荐。Wang 等人^[32]提出了一个用于用户表示学习的时间感知注意网络，该模型考虑了预测项目、近期历史交互项和近期历史交互项的交互时间来估计不同历史交互项的权重以精准估计用户偏好，同时，使用动态用户偏好表示、潜在项目特征表示以及这两种表示之间的线性交互作为其输入，赋予了该模型更强大的特征交互学习能力。Sedhain 等人^[33]提出一种结合了自动编码器和协同过滤思想的推荐算法。算法核心思想是使用自动编码器来学习用户和项目的嵌入向量，从而能够更好地捕捉它们之间的关系。通过将这些嵌入向量结合在一起，可以生成个性化的推荐。在基于匹配方法学习的算法中，模型通过学习参数化的函数或嵌入向量，能够捕捉输入数据中的匹配关系，最大程度地理解输入数据中的交互模式，来提高模型在相关任务上的性能。He 等人^[34]提出一种端到端的模型，使用嵌入层来学习用户和项目的低维向量表示，使用多层神经网络代替了矩阵分解模型中的内积操作，提高了特征的交叉能力。He 等人^[35]提出在嵌入层之上使用外积，从而得到一个更

具表现力和语义合理性的二维交互图。在通过外积获得的交互图之上，使用卷积神经网络来学习嵌入维度之间的高阶相关性，为个性化推荐提供更强大的建模能力。

1.2.2 基于图神经网络推荐算法研究现状

图神经网络最早由 Marco Gori^[36]、Franco Scarselli^[37]等人提出，该模型具有能够整合节点属性信息和拓扑结构信息的巨大潜力，传统推荐算法通常只考虑用户-项目之间的交互信息，而图神经网络能够更全面地利用图结构信息，包括用户-用户、项目-项目之间的关系，以及更高阶的邻居关系（如图 1-1 所示），从而更准确地捕捉用户和项目之间的关联，因此在许多领域广泛应用。Ying 等人^[38]提出了一种名为 PinSage 的图卷积神经网络模型，该模型能够高效地处理大规模图数据，结合了基于随机游走的图采样技术和卷积神经网络的特性，能够在计算效率和模型性能之间取得平衡。Zhang 等人^[39]提出了 Star-GCN，这是一种堆叠和重构的图卷积网络模型。Star-GCN 采用了一种堆叠和重构的方法，分别从不同层次聚合节点信息，并通过重构模块有效地利用深层信息。Xu 等人^[40]提出了一个图上下文自注意力网络模型 GCSAN，用于会话推荐。该模型结合了图结构信息和自注意力机制，以更好地捕捉用户行为序列中的上下文信息。通过构建会话图，将会话中的项目表示为图中的节点，并利用图神经网络（GNN）进行信息传播，从而捕获项目之间的关系。引入了自注意力机制，以更好地捕捉会话中的长程依赖关系。Ma 等人^[41]提出了一个记忆增强图神经网络模型 MA-GNN，用于序列推荐任务。该模型结合了图神经网络和外部记忆模块，以更好地捕捉用户行为序列中的复杂依赖关系。通过构建用户行为序列图，将序列中的项目表示为图中的节点，并利用图神经网络进行信息传播，从而捕获项目之间的关系。引入外部记忆模块，记录并提取用户的长期行为信息，以增强模型的表示能力。由于推荐系统中的大部分信息本质上都具有图结构，因此图神经网络在推荐系统领域中蓬勃发展。

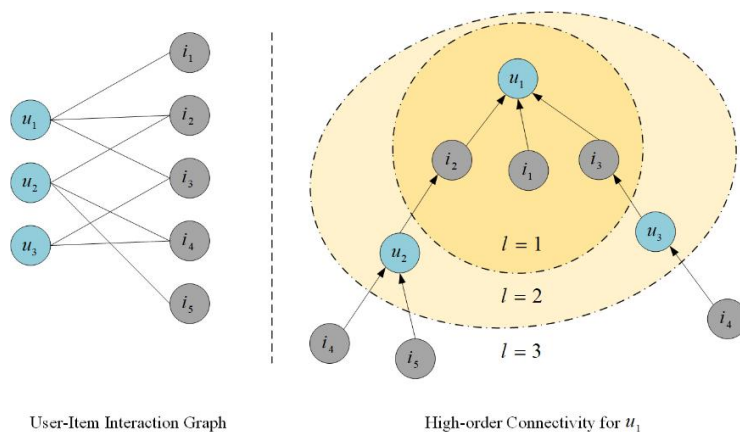


图 1-1 图神经网络与传统推荐算法的交互对比

为了加深对高阶邻居子图结构的使用, Wang 等人^[42]提出了基于图神经网络的协同过滤推荐算法, 灵感来源于图卷积网络 (GCN)^{[22][43]}, 遵循特征变换, 邻域聚合和非线性激活的传播规则来细化嵌入, 解决了用户和项目交互信息的传播信号无法在嵌入层中表现这一挑战。He 等人^[44]提出了一种简化的图神经网络协同过滤算法 LightGCN, 放弃了图卷积神经网络的特征转换和非线性激活, 因为他们对协同过滤并没有实质性帮助, 反而增加了模型训练难度并且降低了推荐性能。LightGCN 通过省略特征转换和非线性激活来简化协同过滤的图卷积神经网络, 降低了模型的复杂程度, 提高了模型训练效率。Mao 等人^[45]观察到消息传递在很大程度上减慢了图卷积神经网络在训练期间的收敛速度, 特别是对于大规模推荐系统, 这阻碍了它们的广泛采用。因此, 进一步提出了一种超简化的图卷积神经网络 UltraGCN。UltraGCN 跳过了无限层的信息传递以实现高效推荐, 允许更合适的边缘权重分配和相对重要性的灵活调整在不同类型的关系之间, 能够易于实现和高效训练。Wang 等人^[46]意识到用户-项目关系的重要作用和先前嵌入函数的局限性, 特别关注了用户-项目关系在更精细的用户意图级别上的关系。因此设计了新的模型, 解纠缠图协同过滤 (DGCF)。具体来说, 通过对每个人的意图分布建模用户-项目交互, 迭代地细化意图感知交互图和表示, 鼓励不同意图的独立性。因此, DGCF 模型可以有效地提取与每个意图相关的信息。Wang 等人^[47]认为考虑购买动机之间的差异可以捕捉更多复杂的交互特性, 全面反映用户偏好。提出了一种新颖的多分量图卷积协同过滤 (MCCF) 方法, 将图卷积网络应用于多组件, 使模型更适应多样化的用户和项目特征, 为个性化推荐提供了强大的建模工具。

1.3 当前研究不足

当前研究中, 基于图神经网络的协同过滤推荐算法虽然在处理复杂的用户-项目关系和捕捉高阶协作信号方面展示了巨大的潜力, 但也存在一些显著的局限性。

第一, Li 等人^[48]和 Xu 等人^[49]指出了 GCN 模型在层数堆叠方面的局限性, 无法像卷积神经网络模型那样进行深层堆叠。具体来说, 当使用多层 GCN 进行学习时, 相关的任务效果会显著下降。这是因为图神经网络模型会出现过度平滑问题^[48]。在训练过程中, 随着网络层数和迭代次数的增加, 每个节点的隐层表示趋向于收敛到相同的值 (即在空间上聚集到同一个位置), 这导致节点特征之间的差异性减小, 使得模型难以区分不同节点的特征, 从而使相应的学习任务变得更加困难。此外, 过度平滑问题还会导致模型对噪声和异常数据更加敏感, 进一步影响模型的泛化能力和预测精度。为了克服这一挑战, 研究者

们提出了多种改进方法，例如引入跳跃连接、残差连接或更复杂的图注意力机制，以增强模型对不同层级信息的捕捉能力，同时缓解过度平滑问题。然而，这些方法往往增加了模型的复杂性和计算成本，使得在实际应用中仍然面临诸多挑战。因此，如何在保证模型性能的同时有效解决过度平滑问题，仍然是图神经网络研究中的一个重要方向。

第二，现有的基于图神经网络的协同过滤推荐模型比以往任何时候都更加复杂，旨在捕捉更高阶的协作信号。这种复杂的模型在提升推荐性能的同时，也带来了显著的计算开销，使得模型训练变得异常耗时且资源密集。此外，复杂模型的参数调优也变得更加困难，需要大量的实验和调整来找到最优参数组合。这些因素导致了在大规模图数据上的训练效率和可扩展性显著降低，使得在实际应用中难以部署和运行。这种局限性阻碍了图神经网络推荐模型在更广泛领域的应用，特别是在需要处理数亿级别用户和项目的大规模推荐系统中。为了克服这些挑战，需要在模型设计和优化算法上进行创新，简化模型结构，开发更加高效的训练和推理算法，以提高图神经网络推荐模型的可扩展性和实际应用价值。

第三，基于图的方法可能更容易受到数据噪声的影响^[50]。用户隐式反馈^[51]数据，包括购买历史、浏览历史、搜索模式等，不直接表现用户的明确倾向，其噪声具有严重的负面影响。拟合噪声数据会阻碍推荐模型准确学习实际用户偏好，进而降低推荐算法的性能。例如，用户可能点击或浏览了某个商品，但这并不代表其真实的购买意图，这样的噪声数据会误导模型的训练，导致推荐结果的偏差。此外，推荐系统中还存在着不可忽略的长尾问题。长尾问题意味着推荐系统倾向于推荐热门商品，而忽略冷门商品，导致热门商品越来越受欢迎，而冷门商品则被进一步边缘化。这种趋势不仅影响了用户的个性化体验，还限制了系统的多样性和发现新奇商品的能力。在长尾问题的影响下，用户可能被不断推荐已经热门的商品，而无法接触到潜在符合其兴趣的冷门商品，进而降低了用户对推荐系统的满意度和信任度。解决这些问题需要在模型设计和训练过程中引入有效的去噪方法和长尾商品的曝光机制，以平衡推荐的广度和深度，从而提升推荐系统的整体性能和用户体验。

第四，以往的工作很少对邻域中的关系信息进行显式建模，通常忽略了邻居节点之间的潜在相关性，这在很大程度上限制了模型的表现。具体来说，忽视邻居关系的信息建模不仅无法充分捕捉用户的兴趣和行为模式，还不利于缓解模型的冷启动问题。冷启动问题，即当新用户或新项目缺乏足够的历史交互数据时，推荐系统无法给出准确的推荐。在用户-项目交互图中，用户的偏好可以由其所有相邻的项目表示，因此每个项目的相关性和用户的兴趣之间的关系尤为重要。然而，在训练过程中，往往难以避免邻居节点信息的丢失现

象，这会导致用户偏好表示不准确，影响推荐质量。此外，由于邻域关系信息的缺失，模型在面对数据稀疏和冷启动场景时表现尤为不佳。为了提高推荐系统的性能，需要进一步研究和开发能够有效捕捉邻居节点之间隐式相关性的模型，以确保在训练过程中最大限度地保留和利用这些信息，从而提升推荐的准确性和鲁棒性。

1.4 研究内容及论文结构

1.4.1 研究内容

LightGCN 是推荐算法中受到广泛认可的图神经网络算法。通过邻接矩阵传播信息，以捕捉全局图结构。模型结构简洁，不使用偏置项，训练时采用负采样来构建训练样本，使用适当的归一化来防止信息过度稀疏化。LightGCN 是一个高效、轻量级的推荐模型，特别适用于大规模推荐系统，在算法性能上表现出强大的竞争力。本文在 LightGCN 的基础上展开研究，具体的研究内容如下：

(1) 将自监督图学习框架与简化的图神经网络协同过滤算法相结合，提出一种基于自监督学习的简化图神经网络协同过滤算法 SSGCF(Simplifying Graph Neural Network Collaborative Filtering Based On Self-supervised Learning)。通过使用线性嵌入传播代替图卷积网络中的非线性变化，降低了模型的复杂度。为了缓解过度平滑问题，为输入网络设计了一种压缩图学习操作。通过引入自监督图学习框架，在一定程度上能够缓解数据集噪声问题和长尾问题。通过在 Amazon-Books、Gowalla、Yelp2018 公共数据集上进行一系列实验，证实 SSGCF 算法的可靠性和适用性。

(2) 将自监督图学习框架与邻居感知注意力图神经网络协同过滤算法相结合，提出了一种基于自监督学习的邻居感知注意力图神经网络协同过滤算法 NASGCF(Neighbor-Aware Attention Graph Neural Network Collaborative Filtering Based on Self-supervised Learning)，改进了图神经网络协同过滤算法对于用户和项目之间关系的感知能力，缓解了数据集稀疏和冷启动问题。引入邻居感知注意力机制的目的在于增强模型对邻居节点的关注，从而更有效地发现节点之间的联系。在 Amazon-Books、Gowalla、Yelp2018 公共数据集上进行实验，讨论不同参数对于实验结果的影响，探究自监督学习框架和邻居感知注意力机制对于算法性能的提升，与其他对比算法以及 SSGCF 进行对比实验，实验论证了 NASGCF 算法的合理性和优越性。

1.4.2 论文结构

以下是本文的整体结构：

第一章 引言。本章旨在介绍研究的背景和性质，梳理国内外相关研究现状，探讨协同过滤推荐算法和图神经网络推荐算法在推荐系统领域的应用情况，指出当前研究存在的不足。最后，对论文的结构进行总结。

第二章 相关理论及其介绍。本章对推荐系统的应用场景简要的概括，对推荐系统的分类进行概括并简要介绍，详细说明协同过滤算法的分类及原理，对召回率和归一化折损累计增益两类性能评价指标进行详细介绍；介绍图神经网络基本理论，梳理当前主流的基于图神经网络的协同过滤推荐模型及其核心公式；介绍自监督图学习的研究背景和具体过程；阐述注意力机制的基本原理，并给出具体的计算过程公式。

第三章 基于自监督学习的简化图神经网络协同过滤算法。本章分析了当前工作中存在的不足，为了应对当前挑战，提出基于自监督学习的简化图神经网络协同过滤算法；详细介绍算法的设计和描述；针对提出算法，进行实验设计，介绍实验环境、数据和对比算法；进行参数调优实验，讨论算法抗噪实验以及对比实验，实验结果表明提出算法的有效性。

第四章 基于自监督学习的邻居感知注意力图神经网络协同过滤算法。本章对当前研究中存在的局限性进行分析，为了更加关注邻域之间的隐式相关性，提出一种基于自监督学习的邻居感知注意力图神经网络协同过滤算法；阐述了模型的输入模块、数据增强模块、图神经网络编码器模块以及目标优化模块；对实验进行详细设计，介绍对比算法和实验参数；进行参数调优实验，实施自监督图学习和邻域感知注意力对模型效果提升的消融实验以及对比实验。

第五章 结论与展望。本章对本论文所提出的两种算法的研究背景和具体设计进行回顾，针对当前工作的局限性给出了可行的研究思路。

2 相关理论及其介绍

2.1 推荐系统相关理论及介绍

2.1.1 推荐系统概述

随着大数据时代的来临，人们面临着海量的信息和数据。如何在纷繁复杂的信息数据中快速找到自己感兴趣的内容，成为了一个重要的难题。推荐系统正是这样的背景下应运而生，极大地改变了用户与信息互动的方式。推荐系统的核心任务是通过分析用户的历史行为、兴趣和偏好，利用先进的技术和算法，为用户推荐个性化的信息、产品或服务，极大程度提高了用户获取信息的效率。推荐系统的有着多样化的应用场景：例如在电商平台中，向用户推荐商品，提高用户购物体验，并增加销售额；在短视频平台上，帮助用户发现和观看他们可能喜欢的视频；在社交媒体平台上，推荐用户关注的人、浏览内容、参与讨论等；在新闻网站、博客平台上，推荐用户可能感兴趣的文章、新闻或博客；在旅游平台上，推荐系统可以为用户提供个性化的旅行建议和酒店选择，以及可能感兴趣的餐厅和菜品。推荐系统通过分析用户行为和个性化需求，帮助各类平台为用户提供更好的服务和体验。

推荐系统主要可以分为三类：基于内容的过滤推荐系统，协同过滤推荐系统以及混合推荐系统^[52]。（1）基于内容的过滤推荐与用户喜欢的项目属性相似的项目，并根据项目的信息进行推荐的方法。（2）协同过滤使用用户对项目评价的数据进行建模，预测出与用户喜好相似相似的项目，并进行推荐的模型。（3）混合过滤结合了前两种模型，将协同过滤模型用于特征数据和项目实例数据，基于内容的过滤模型用于增强数据，是一种根据用户对项目的评价与推荐系统预测的评价相符合的程度逐渐调整权重的方法。

推荐系统的推荐流程是一个系统性的过程，从数据的收集和处理开始，经过用户和项目的建模与特征提取，生成推荐候选项目，经过过滤和排序最终呈现给用户。这一过程同时注重用户反馈的收集与模型的优化和更新，形成一个闭环循环，不断持续提升系统的性能和个性化程度。

2.1.2 协同过滤推荐算法

协同过滤推荐算法基本思想是通过分析用户与项目的交互，利用用户或项目之间的相似性来进行推荐。按照广泛认可的协同过滤推荐算法分类方式，划分为基于内存的协同过滤和基于模型的协同过滤。

基于内存的协同过滤可以进一步划分为基于用户的协同过滤和基于项目的协同过滤。前者基于用户之间的相似性进行推荐。如果两个用户在过去的行为中有相似的兴趣，那么他们可能在未来也有相似的兴趣。通过计算用户之间的相似性，然后利用相似用户的历史行为向目标用户进行推荐。而后者关注项目之间的相似性。如果用户喜欢一个项目，那么他们可能也会喜欢与该项目相似的其他项目。通过计算项目之间的相似性，根据用户已喜欢的项目向用户推荐类似的项目。

基于模型的协同过滤算法通过建立模型来预测用户对项目的喜好或评分，实现对用户的个性化推荐。通常利用机器学习或统计模型来学习用户和项目之间的关系，以便在新的用户-项目组合上进行预测。

2.1.3 推荐算法的评价指标

推荐算法评价指标的作用是为了度量和评估推荐算法在不同方面的性能，在实际应用中有助于全面地理解和优化推荐系统的表现。选择适当的评价指标取决于具体的推荐任务和目标，评价指标的合理选择能够提高推荐系统的效果和用户满意度。最常见的推荐算法评价指标可以分成三种：（1）评分预测指标，如均方根误差 (RMSE)，平均绝对误差 (MAE) 等。通过比较实际用户评分与推荐系统生成的预测评分，可以了解模型在预测用户对项目的喜好方面的表现，适用于评分预测任务。（2）集合推荐指标，如准确率 (Precision)，召回率 (Recall)，F1 值等。评估预测的项目集合，适用于 Top-N 推荐任务。（3）排名推荐指标，如归一化折损累计增益 (NDCG)。可以帮助评估推荐系统在给定用户查询或者上下文条件下，对相关项目的排序质量和性能表现。

召回率和归一化折损累计增益分别关注推荐算法的全面性和排序质量。为了更好地评估算法性能，本研究使用召回率和归一化折损累计增益作为算法的评价指标。以下对这两种评价指标进行详细说明：

（1）**召回率**：推荐算法的预测结果和实际结果有四种可能的情况，在混淆矩阵中表示如下：真正性 TP (True Positive)：表示推荐系统成功地将用户实际感兴趣的项目推荐给用户的数量；假负性 FN (False Negative)：表示推荐系统未能将用户实际感兴趣的项目推荐给用户的数量；假正性 FP (False Positive)：表示推荐系统错误地将用户实际不感兴趣的项目推荐给用户的数量； TN (True Negative)：在推荐系统中，通常不涉及。在其他二分类问题中， TN 表示模型正确地将负例（非感兴趣项目）分类为负例的数量。因此，召回率的表达式如公式 2-1 所示^[52]。

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2-1)$$

通过召回率可以计算出基于推荐系统生成的推荐列表与用户实际感兴趣项目集之间的交集关系，其优点在于强调推荐系统覆盖用户实际兴趣的能力，能够全面评估系统推荐的项目在用户感兴趣项目中的占比。

(2) **归一化折损累计增益**：NDCG (Normalized Discounted Cumulative Gain) 是一种用于评估推荐列表排序质量的指标。NDCG 与折损累计增益 (DCG) 和理想折损累计增益 (IDCG) 有着密切联系，其结果是两者的比值。DCG 反映了用户对于位于推荐列表前面的项目的更高偏好。折损累计增益的表达式如公式 2-2 所示^[52]。

$$\text{DCG}_k = \sum_{i=1}^k \frac{\text{rel}_i}{\log_2(i+1)} \quad (2-2)$$

其中， rel_i 表示推荐列表中第 i 个项目的相关性， k 是推荐列表的长度。DCG 越大，表示推荐系统生成的排序列表在相关性和位置上都更好地满足用户的偏好。IDCG 反映了推荐系统在当前任务中可能达到的最佳排序质量。理想折损累计增益的值是折损累计增益在理想情况下的最大值。可以得到归一化折损累计增益的表达式如公式 2-3 所示^[52]，

$$\text{NDCG}@k = \frac{\text{DCG}_k}{\text{IDCG}} \quad (2-3)$$

其中， $\text{NDCG}@k$ 表示考虑前 k 个项目的 NDCG 值，以更细粒度地评估推荐列表的质量，特别适用于排序任务的评估。

2.2 图神经网络相关理论和技术

2.2.1 图神经网络基本理论

在图神经网络中，图结构由两个集合 V 和 E 组成，表示为 $G = (V, E)$ ， V 是节点的集合， E 是边的集合。设 U 和 I 分别为用户和项目的集合。 $v_i \in V$ 表示节点 i ， $e_{ij} = (v_i, v_j)$ 表示从节点 j 到节点 i 的边。同时，将节点 u 的邻居集合定义为 $N(u) = \{v \in V | (v, u) \in E\}$ 。按照不同定

义，图结构有以下几种类型：

有向图/无向图：有向图和无向图的主要区别在于边的方向性。在有向图中，通常使用箭头来表示边的方向，而在无向图中，则不需要指定边的方向。

同构图/异构图：同构图和异构图的主要区别在于图中节点和边的类型和属性是否相同。在同构图中，所有节点和边都是同一类型的，而在异构图中，节点和边可以具有不同的类型和属性。

有权图/无权图：有权图和无权图的主要区别在于对边的权重是否进行了明确的描述。有权图强调边的权重，而无权图只关注节点之间的连接关系。

连通图/非连通图：连通图和非连通图的主要区别在于图中的节点是否能够通过边连接到图中的其他节点。连通图通常用于描述具有一定连通性的网络结构，而非连通图则可能用于描述由多个不相互连接的部分组成的图结构。

简单图/多重图：简单图和多重图的主要区别在于边的重复性和自环的存在。简单图中的边是唯一的且没有自环，而多重图允许边的重复出现和自环的存在。

图神经网络的理念是利用消息传递机制，将节点的邻居信息进行聚合，然后更新节点的特征表示。其核心是将邻域聚合方案应用于图，通过聚合邻域节点来更新自我节点的特征表示。邻域聚合可以表示为公式 2-4^[42]。

$$n_v^{(k)} = AGG^{(k)}(\{h_u^{(k)}, \forall u \in N_v\}) \quad (2-4)$$

其中 $n_v^{(k)}$ 表示节点 v 的邻居节点在第 k 层的聚合后的向量表示， $h_u^{(k)}$ 表示节点 u 在第 k 层的向量表示， N_v 表示节点 v 的邻居集合。 $AGG^{(k)}$ 表示在 k 层的聚集函数，是图卷积的核心，常见的聚合方式包括均值聚合、注意力聚合、图卷积聚合等，考虑目标节点及其邻居节点的第 k 层表示。当获得聚合后的向量表示之后，可以对节点进行更新，表示为公式 2-5^[42]。

$$h_v^{(k+1)} = UPD^{(k)}(h_v^{(k)}, n_v^{(k)}) \quad (2-5)$$

其中， $UPD^{(k)}$ 表示第 k 层更新操作所对应的函数。常用的更新方式包括邻居平均更新、图卷积更新、图注意力神经网络等方式。

2.2.2 基于图神经网络的协同过滤推荐模型

协同过滤算法的主要任务是使用用户与项目相互之间的数据，通过分析用户对项目的历史行为，增强用户向量和项目向量的表征，来预测用户对尚未交互的项目的喜好程度。通过借鉴图神经网络应用于图结构数据的特点、表示学习的能力以及信息传播的模拟等方面，将图神经网络与协同过滤算法相结合，可以更好地利用图结构数据的特性，提高推荐系统的性能和效果。对当前主流的图神经网络架构进行概述如下：

(1) **图卷积网络**：传统卷积神经网络通过卷积和池化操作来获取到图像中的部分信息和整体结构，在图像处理领域取得成功。然而传统卷积神经网络不适用于非规则的图结构数据，无法处理现实生活中的数据。Kipf 等人^[22]受到卷积神经网络的启发，提出了一种用于处理图结构数据的神经网络模型，即图卷积网络。图卷积网络的主要目标是学习图中节点的表示，使得节点表示能够充分地捕捉节点之间的关系和结构信息。Wang 等人^[42]提出一种基于图神经网络的协同过滤推荐算法 NGCF (Neural Graph Collaborative Filtering)，将用户-物品交互数据建模成一个二部图，并通过多层次的图卷积操作来学习用户和物品的嵌入表示，从而实现个性化推荐。He 等人^[44]提出一种简化的图神经网络协同过滤算法 LightGCN，采用了轻量级的图卷积操作，只保留了用户和物品之间的一阶邻居关系，避免了多层次的传播和过度平滑问题。在图卷积网络的设计中，提出了一种基于图拉普拉斯矩阵的卷积操作，通过该操作可以在图结构上执行卷积运算，并利用邻接矩阵来描述节点之间的连接关系。图卷积操作中，通常包括信息传播和表示更新两个步骤，如公式 2-6，公式 2-7 所示^[22]。

邻域聚合：

$$h_v^{(k+1)} = \sigma \left(\sum_{u \in N(v)} \frac{1}{c_v} W^{(k)} h_u^{(k)} \right) \quad (2-6)$$

特征更新：

$$h_v^{(k+1)} = \sigma \left(W^{(k)} h_v^{(k)} + \sum_{u \in N(v)} \frac{1}{c_v} h_u^{(k)} \right) \quad (2-7)$$

其中, $W^{(k)}$ 是第 k 层的权重矩阵, σ 是激活函数。 c_v 是归一化系数, 通常为节点 v 的度数。

图卷积网络可以高效地获取到图结构中的全局结构和局部特征, 广泛应用于社交网络分析、推荐系统、生物信息学等领域, 并且在不断地得到改进和扩展。

(2) **图注意力网络**: 在传统的图卷积网络中, 每个节点的表示是通过对其邻居节点的特征进行加权平均来计算的, 权重通常是固定的。然而, 在实际应用中, 不同的邻居节点对目标节点的贡献可能是不同的, 需要有针对性地加权聚合邻居节点的特征。为了解决这个问题, Velickovic 等人^[53]提出了图注意力网络, 通过引入注意力机制来增强图卷积网络, 来更好地捕捉节点之间的关系和特征表示。Hamilton 等人^[54]提出一种叫做 GraphSage 的基于图注意力网络的推荐算法, 采用了图采样和聚合的方式来学习节点的表示, 利用了节点邻居的信息来生成节点的嵌入表示, 从而实现个性化的推荐。Berg 等人^[55]提出一种利用图注意力网络进行矩阵补全的推荐算法 GCMC, 将用户-物品交互矩阵建模成一个二部图, 并利用图注意力机制来学习用户和物品的嵌入表示, 从而填补缺失的交互信息。图注意力网络使用一个学习到的权重来计算每个邻居节点对目标节点的贡献, 然后将所有邻居节点的贡献加权求和, 得到目标节点的新表示。具体来说, 包括以下两个步骤: 首先, 计算注意力权重系数, 如公式 2-8 所示^[55]。

$$a_{vu}^{(l)} = \frac{\exp(\text{LeakyReLU}(a^T [W^{(l)} h_v^{(l)} \parallel W^{(l)} h_u^{(l)}]))}{\sum_{k \in N(v)} \exp(\text{LeakyReLU}(a^T [W^{(l)} h_v^{(l)} \parallel W^{(l)} h_k^{(l)}]))} \quad (2-8)$$

其中, $a_{vu}^{(l)}$ 是第 l 层节点 u 到节点 v 的注意力权重系数, LeakyReLU 是注意力函数, 用于引入非线性, a^T 是可学习的权重向量, 用于计算注意力权重, $W^{(l)}$ 是可学习的权重矩阵, 用于第 l 层将节点特征映射到新的特征空间。最后, 使用注意力权重系数对节点 v 的邻居节点特征进行加权聚合, 得到最终的节点特征表示, 如公式 2-9 所示^[55]。

$$h_v^{(l+1)} = \sigma \left(\sum_{u \in N(v)} a_{vu}^{(l)} W^{(l)} h_u^{(l)} \right) \quad (2-9)$$

其中, σ 是激活函数。图注意力网络能够动态地调整节点之间的关系权重, 从而更好

地捕捉图结构中的重要信息。当前广泛应用于图结构数据的节点分类、链接预测等任务，并在许多领域取得了优异的性能。

(3) **图采样与聚合网络**：在大规模图数据中，传统的图神经网络模型（比如图卷积网，GCN）存在着计算和内存需求巨大的问题。为了应对大规模图数据的挑战，Hamilton 等人^[54]提出图采样与聚合网络。与传统的图卷积网络不同，图采样与聚合网络不依赖于全局的图结构信息，而是通过局部采样和聚合来实现节点表示的学习。其核心思想是利用邻居节点的特征来更新目标节点的表示，然后通过多层的采样和聚合操作来逐步扩展节点的上下文信息。如公式 2-10，公式 2-11 所示^[54]。

邻居采样与聚合：

$$h_v^{(k)} = AGG^{(k)} \left(\{h_u^{(k-1)} : u \in N_{(v)} \cup \{v\}\} \right) \quad (2-10)$$

邻居采样与更新：

$$h_v^{(k)} = \sigma \left(W^{(k)} \cdot CONCAT \left(h_v^{(k-1)}, MEAN \left(h_u^{(k-1)} : u \in N_{(v)} \cup v \right) \right) \right) \quad (2-11)$$

其中， $AGG^{(k)}$ 是在第 k 层的聚合函数，用于聚合节点 v 及其邻居节点的表示。 $CONCAT$ 是将两个向量拼接在一起， $MEAN$ 是对输入向量的均值池化操作。

(4) **图同构网络**：传统的图神经网络模型，如图卷积网络，通常通过对节点的邻居信息进行聚合来学习节点的表示。然而，这种方法存在不变性问题和可区分性问题。为了解决这些问题，Xu 等人^[56]提出图同构网络，将图结构中的节点和边转换为低维向量表示，并通过神经网络对这些向量进行聚合和转换，从而学习图的整体表示。Wang 等人^[57]提出 HAN 算法用于处理异构图数据。该模型通过引入注意力机制来学习不同类型节点之间的关系，并使用多层图注意力网络来学习节点的表示，能够有效地捕获异构图中的复杂关系，并实现个性化的推荐效果。Yun 等人^[58]提出 GTN 算法，利用了 Transformer 模型的思想，通过自注意力机制来学习节点之间的关系。GTN 可以有效地捕获图数据中的全局信息，并生成节点的高质量表示，用于推荐任务。Dong 等人^[59]提出一种 Metapath2Vec 算法，用于学习异构图中节点的表示。它通过随机游走和 Skip-gram 模型来学习节点的嵌入表示，从

而实现了对节点之间复杂关系的建模和推荐任务的高效实现。与传统的图卷积网络 (GCN) 不同，图同构网络不依赖于节点的邻居信息，而是直接对节点的特征进行聚合和更新，具体公式 2-12，公式 2-13 如下^[56]。

聚合操作：

$$h_v^{(k)} = MLP^{(k)} \left((1 + \epsilon^{(k)}) \cdot h_v^{(k-1)} + \sum_{u \in N(v)} h_u^{(k-1)} \right) \quad (2-12)$$

其中， $MLP^{(k)}$ 是一个多层感知机， $\epsilon^{(k)}$ 是一个可学习的参数。

全局汇聚操作：

$$h_{global} = MLP_{global} \left(\sum_{v \in V} h_v^{(k)} \right) \quad (2-13)$$

上式对所有节点的特征表示进行求和，并通过 MLP 进行全局汇聚，得到图的整体表示 h_{global} 。

2.3 自监督学习

2.3.1 监督学习、半监督学习和无监督学习

(1) 监督学习：在监督学习中，训练数据集包含输入特征和对应的标签。学习算法通过使用这些已标记的数据来训练模型，并根据输入特征预测或分类标签。监督学习的目标是从已知的训练数据中学习到一个能够对新数据做出准确预测或分类的映射关系。典型的监督学习任务包括分类和回归。

(2) 无监督学习：在无监督学习中，训练数据集不包含标签，只有输入特征。学习算法试图从这些未标记的数据中发现数据的内在结构或模式，以便对数据进行更深入的理解。典型的无监督学习任务包括聚类、降维和关联规则挖掘等。

(3) 半监督学习：半监督学习旨在利用少量的标记数据和大量的未标记数据来训练模型。在标记数据稀缺的情况下，通过结合已标记和未标记数据来提高模型性能。这种方法通常用于解决标注数据昂贵或难以获取的问题。

2.3.2 自监督学习

自监督学习 (SSL) 是在面对数据稀缺或成本高昂的情况下提出的，特别是在无监督学

习领域。传统的监督学习方法通常需要大量标记的数据，然而在许多情况下，获取标记数据的成本很高，或者数据本身就很难标记。为了充分利用大量的未标记数据，自监督学习提出了一种无需人工标注的标签即可训练模型的方法。通过这种方式，自监督学习将无监督学习转化为监督学习任务，从而可以使用监督学习算法来训练模型。因此，自监督学习同时具有监督学习和无监督学习的特点。自监督学习在计算机视觉 (CV)^[7]和自然语言处理 (NLP)^[9]中很流行，但在推荐算法领域的应用不足。Wu 等人^[60]将自监督学习应用于图神经网络，提出自监督图学习算法 (SGL)，有效地提高了基于图神经网络推荐算法的准确性，以及对交互噪声的鲁棒性。

2.3.3 自监督图学习过程

自监督图学习 (SGL)，核心思想是利用图数据中的自然结构或关系来生成监督信号，从而训练模型学习图数据的表示或特征。与传统的监督学习不同，自监督图学习不需要大量的手动标注数据。相反，它通过设计一些辅助任务（也称为预训练任务）来从图中挖掘监督信息，例如预测节点的上下文、预测边的存在性或预测图的属性等。这些辅助任务能够帮助模型学习到图数据的内在结构和关系，并生成对下游任务有用的特征表示。自监督图学习的一个关键优势是它能够在没有大量标注数据的情况下学习图数据的表示，这使得它特别适用于那些标注数据稀缺或难以获得的场景。此外，自监督图学习还可以提高模型的泛化能力，因为它学习的特征通常是更通用的，并且可以应用于不同的图数据集和任务中。

自监督图学习由两个关键部分组成：一是数据增强，为每个节点生成多个视图；二是对比学习，与其他节点相比，最大程度地增强同一节点在不同视图之间的相同性。对于基于用户-项目图的图神经网络，图结构作为输入数据，在表示学习中起着至关重要的作用。通过节点丢弃、边丢弃和随机行走的方式，修改图的邻接矩阵来建立未标记数据的空间。最后，将对比学习用于修改后的模型进行基于图神经网络的深度学习。

2.4 注意力机制

注意力机制^[61]是一种模拟人类视觉和注意力过程的机制，在神经网络中被广泛应用于提高模型对输入数据的关注和处理能力。其核心功能在于提升模型对输入数据的精细关注和处理能力。在面对庞大而复杂的输入数据时，注意力机制能够动态地、智能地调整对不同部分的注意力权重，使模型能够精准地聚焦于那些关键、有价值的信息上。在处理复杂的输入数据时，动态地分配不同部分的注意力权重，从而使得模型能够更加有效地关注重

要的信息，提高模型的性能和泛化能力。

在实际应用中，注意力机制在自然语言处理^[9]、计算机视觉^[7]等邻域得到了普遍运用，在自然语言处理中，注意力机制可以帮助模型更好地理解解析文本中的关键信息，从而提高文本分类、情感分析、机器翻译等任务的性能。在计算机视觉领域，注意力机制则可以帮助模型更加准确地识别图像中的关键特征，从而改善图像分类、目标检测等任务的准确度。值得注意的是，注意力机制在推荐算法方面也取得了显著的成就。在海量数据中筛选出用户可能感兴趣的内容，是推荐算法的核心任务之一。而注意力机制能够帮助推荐算法更加精准地捕捉用户的兴趣和偏好，从而为用户推荐更加符合其需求的内容。这种技术的应用，不仅提升了推荐算法的准确度，极大改善了用户体验。

2.4.1 注意力机制计算过程

注意力机制的具体数学公式通常包含三个主要部分：计算注意力分数、计算注意力权重和加权求和。假设有输入特征 $X \in \mathbb{R}^{N \times d}$ 和上下文信息 $C \in \mathbb{R}^{N \times d'}$ ，其中 N 表示样本数， d 表示输入特征的维度， d' 表示上下文信息的维度。

(1) 计算注意力分数 (Attention Scores) :

$$Scores(X, C) = ReLU(XW_x + CW_c + b) \quad (2-14)$$

其中， $W_x \in \mathbb{R}^{d \times d_s}$ 和 $W_c \in \mathbb{R}^{d' \times d_s}$ 是可学习的权重矩阵， $b \in \mathbb{R}^{d_s}$ 是偏置向量， d_s 是注意力分数的维度。 $ReLU$ 函数用于引入非线性，增强模型的表达能力。

(2) 计算注意力权重 (Attention Weights) :

$$Weights = softmax(Scores(X, C)) \quad (2-15)$$

通过 $softmax$ 函数将注意力分数归一化为概率分布，得到注意力权重。

(3) 加权求和 (Weighted Sum) :

$$Sum = Weights^T X \quad (2-16)$$

使用注意力权重对输入特征进行加权求和，得到最终的加权表示结果。

2.5 本章小结

本章第一节对推荐系统的相关理论进行了介绍，对协同过滤推荐算法进行了概述，详细介绍了常用的推荐算法评价指标；第二节对图神经网络相关理论和技术进行了详细介绍，总结梳理了当前主流的图神经网络模型架构，并给出模型的核心公式；第三小节介绍了自监督学习相关背景，描述了自监督图学习的过程，为后续论述自监督图学习算法框架做铺垫；第四节介绍了注意力机制的原理，总结了注意力机制的计算过程。

3 基于自监督学习的简化图神经网络协同过滤算法

基于用户历史行为数据的协同过滤推荐算法广泛应用于推荐系统，经过分析用户和项目之间的交互关系，来推断出用户可能感兴趣的项目。随着深度学习的发展，图神经网络在推荐系统领域大放异彩。最近的大量研究中，将图神经网络应用于基于协同过滤的推荐系统，通过将用户-项目交互建模为二部图，取得了优异的性能。图神经网络能够充分利用用户-项目交互数据构建图结构，通过多层特征聚合和信息传播来捕捉用户和项目之间的高阶关系，从而提高推荐系统的准确性和个性化程度。

然而，这些复杂的模型在大型图上的非线性激活训练中面临计算复杂度高、数据集稀疏的困难，并且带来了效率和扩展性方面的问题，这阻碍了模型的广泛应用。另外，在多层图神经网络中，随着多次信息传播和聚合后，节点表示逐渐趋向于相似或均匀，失去了原始图结构中的细节和差异性，产生过度平滑问题。因此，无法对更深的层进行建模，影响模型的表达能力和推断结果的准确性。此外，图神经网络推荐算法更容易受到数据噪声的影响。噪声数据使得模型训练不稳定，模型对数据的拟合程度下降，最终导致推荐系统无法准确地理解用户的兴趣和偏好。

为了应对上述挑战，提出了一种基于自监督学习的简化图神经网络协同过滤算法 (SSGCF, Simplifying Graph Neural Network Collaborative Filtering Based On Self-supervised Learning)。简化图神经网络协同过滤模型是对当前基于图卷积网络的模型的改进，通过消除卷积网络中的非线性部分，将结果函数降低复杂程度，使得模型易于训练，可扩展至大型数据集。其次，使用一种有效的划分技术，用于减少输入图结构的规模，避免无限层的显式消息传递。通过计算压缩图上的网络嵌入来获得图中每个节点的嵌入的初始化，减轻了图卷积聚合操作中的过度平滑问题。将自监督图学习算法架构与简化图神经网络协同过滤算法相结合，降低噪声数据对于模型训练的干扰，有利于提高模型整体的鲁棒性和准确性。

3.1 模型设计与描述

自监督图学习算法通过设计与上下文相关的预训练任务，不依赖于大量标注数据，能在图结构数据中自动生成监督信号，提升模型的表示学习能力。在基于自监督学习的简化图神经网络协同过滤算法模型中，将自监督图学习与基于图神经网络协同过滤算法结合进行多任务学习，其优势在于可以利用未标记数据进行有效的表示学习，从而提高模型的泛化能力和性能，能够有效的缓解数据集噪声问题，进而增强模型对图数据的理解 and 应用能

力。自监督图学习算法框架如算法 3-1 所示。

算法 3-1 自监督图学习算法

自监督图学习算法 (SGL)

Input: 图结构 G , 嵌入矩阵 E , 图卷积网络层数 L , 自监督学习参数 λ_1 , 正则化参数 λ_2 , 温度系数 τ , 概率系数 ρ , 总训练序列 s

Output: 节点表示矩阵 H

```

1  count = 0
2  while count < s do
3    switch 数据增强 do
4      case 节点抛弃 do
5        根据概率  $\rho$  生成  $M_1, M_2 \in 0,1^{|V|}$ 
6         $s_1(G) = (M_1 \odot V, E)$ , //其中,  $M_1, M_2 \in \{0,1\}^{|V|}$ 是两个掩码向量
7         $s_2(G) = (M_2 \odot V, E)$ , //用于在节点集 $V$ 上生成两个子图
8      end
9      case 边抛弃 do
10       根据概率  $\rho$  生成  $M_1, M_2 \in 0,1^{|E|}$ 
11        $s_1(G) = (V, M_1 \odot E)$ , //其中,  $M_1, M_2 \in \{0,1\}^{|E|}$ 是边集合中
12        $s_2(G) = (V, M_2 \odot E)$ , //两个掩码向量
13     end
14     case 随机游走 do
15       for  $l \in \{1, \dots, L\}$  do
16         根据概率  $\rho$  生成  $M_1^{(l)}, M_2^{(l)} \in 0,1^{|E|}$ 
17          $s_1^{(l)}(G) = (V, M_1^{(l)} \odot E)$ , //其中,  $M_1^{(l)}, M_2^{(l)} \in \{0,1\}^{|E|}$ 表示 $l$ 层的
18          $s_2^{(l)}(G) = (V, M_2^{(l)} \odot E)$ , //边集合 $E$ 的两个掩码向量
19       end
20     end
21   end
22   foreach 训练小批量 do
23      $H = AGG(E, G)$ 
24     计算每个卷积层的节点增强表示矩阵  $H_1^{(l)}$  and  $H_2^{(l)}$ 
25     计算最终的节点增强表示矩阵  $H_1$  and  $H_2$ 
26     计算损失函数
27     通过梯度下降算法更新模型参数
28   end
29   count ++
30 end

```

具体来说, SSGCF 模型中主要包括四个部分。

输入模块: 将用户和项目交互的图结构作为模型的输入;

数据增强模块: 对输入数据通过节点丢弃、边丢弃和随机游走的方式生成增强数据;

图神经网络编码器：使用简化图神经网络协同过滤模型，对初始模型数据和增强数据进行图卷积操作，生成节点表示矩阵；

目标优化模块：首先进行有监督学习，使用初始模型数据中的监督信号；其次，使用增强数据构造的辅助监督信号，来进行对比学习；最终，确定总体优化目标。

模型的流程如下：首先，将图结构数据通过节点丢弃、边丢弃、随机游走的方式为每个节点生成多个视图，进行数据增强。接着对图结构划分构造出压缩图结构，进行特征传播。将用户和项目的邻居节点对应的嵌入向量汇入自适应嵌入传播层，使用迭代方法将每个节点根据其邻居的嵌入更新自己的嵌入，直到达到收敛。通过比较学习提高同一节点在不同视图图中的一致性，并采用多任务训练策略联合优化推荐任务和自监督学习任务。

SSGCF 模型如图 3-1 所示。

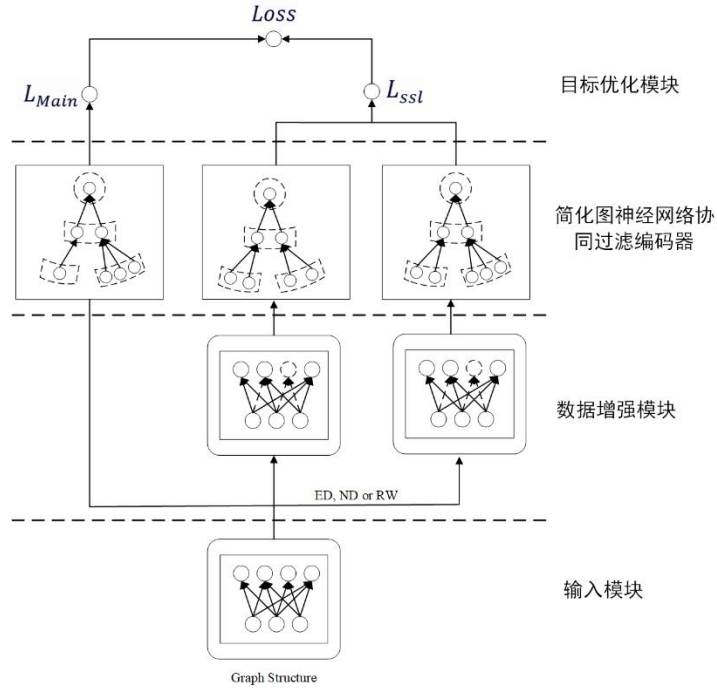


图 3-1 SSGCF 模型架构

3.1.1 输入模块设计

从推荐场景中抽象出来的图中有两种节点，即用户节点和项目节点。输入模块的目的是将用户 u 和项目 i 映射成嵌入向量 $e_u^{(0)} \in \mathbb{R}^d$ 和 $e_i^{(0)} \in \mathbb{R}^d$ ，其中 d 表示嵌入的维数。根据构建嵌入向量表 E 来定义用户和项目的参数矩阵，如公式 3-1 所示。

$$E = [e_{u_1}, ..., e_{u_M}, e_{i_1}, ..., e_{i_N}] \quad (3-1)$$

其中， M 和 N 分别表示用户和项目的数量。

3.1.2 数据增强模块设计

节点丢弃、边丢弃和随机行走的运算符可以统一表示为公式 3-2 所示。

$$H_1^{(l)} = AGG(H_1^{(l-1)}, s_1(G)), H_2^{(l)} = AGG(H_2^{(l-1)}, s_2(G)), s_1, s_2 \sim S \quad (3-2)$$

在图 G 上创建节点 $H_1^{(l)}$ 和 $H_2^{(l)}$ 的视图，使用两个随机选择 s_1 和 s_2 。将增广运算符描述如下：

- (1) **节点丢弃 (ND)**：每个节点连同其连接的边一起从图中被丢弃的概率为 ρ 。 s_1 和 s_2 可以建模为公式 3-3 所示。

$$s_1(G) = (M' \odot V, E), s_2(G) = (M'' \odot V, E) \quad (3-3)$$

其中， $M', M'' \in \{0,1\}^{|V|}$ 是两个掩码向量，用于在节点集 V 上生成两个子图。

- (2) **边丢弃 (ED)**：以丢失率 ρ 在图中丢弃边。如公式 3-4 所示。

$$s_1(G) = (V, M_1 \odot E), s_2(G) = (V, M_2 \odot E) \quad (3-4)$$

其中， $M_1, M_2 \in \{0,1\}^{|E|}$ 是边集合中的两个掩码向量。为了可以获取节点部分结构中的有用模式，将两个子图结合在一起，在接下来给予更好的鲁棒性。

- (3) **随机游走 (RW)**：节点丢弃和边丢弃创建了一个在所有图卷积层中共享的子图。

为了提升模型的表征能力，将不同的层给予不同的子图。如公式 3-5 所示。

$$s_1^{(l)}(G) = (V, M_1^{(l)} \odot E), s_2^{(l)}(G) = (V, M_2^{(l)} \odot E) \quad (3-5)$$

其中, $M_1^{(l)}, M_2^{(l)} \in \{0,1\}^{|E|}$ 表示 l 层的边集合 E 的两个掩码向量。

3.1.3 图神经网络编码器设计

图神经网络编码器使用简化图神经网络协同过滤模型, 作为 SSGCF 模型的核心部分, 如图 3-2 所示。

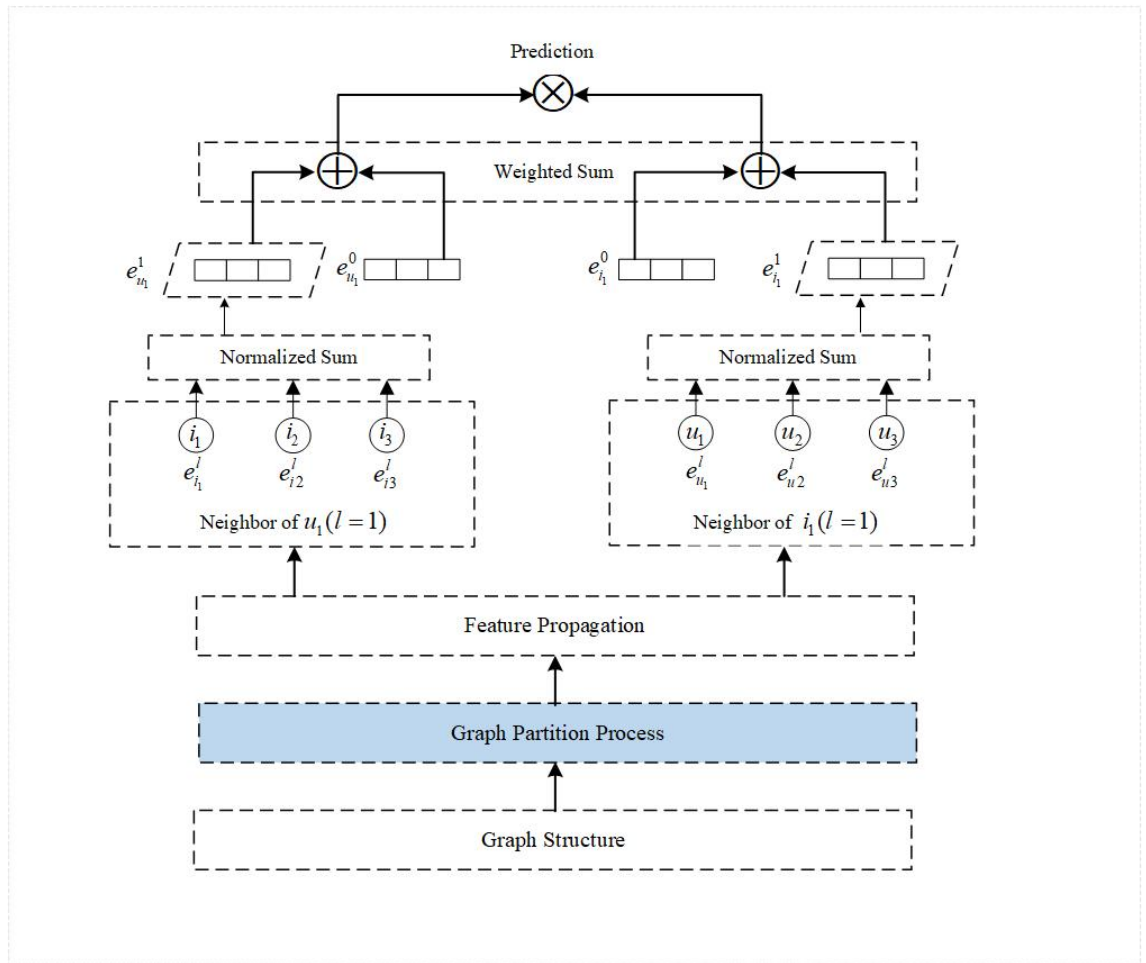


图 3-2 SSGCF 模型核心部分——简化图神经网络协同过滤模型

(1) 简化特征传播: 在图神经网络模型中, 为了降低模型的复杂程度, 去掉了每层之间的非线性过渡函数。将每个用户 u 和每个项目 i 的更新嵌入表示为公式 3-6, 公式 3-7 所示。

$$e_u^{k+1} = \left[\frac{1}{d_u} e_u^k + \sum_{j \in N_u} \frac{1}{d_j + d_u} e_j^k \right] W^k \quad (3-6)$$

$$e_i^{k+1} = \left[\frac{1}{d_i} e_i^k + \sum_{u \in N_i} \frac{1}{d_i + d_u} e_u^k \right] W^k \quad (3-7)$$

其中, d_i 和 d_u 为用户和项目二部图 G 中项目 i (用户 u) 的对角度。 N_u 和 N_i 为图 G 中用户和项目的邻居节点。 W^k 表示非线性变换。

(2) 图划分技术: 当前许多基于图卷积网络模型及其变体模型, 在深度 $K=2$ 时达到最佳性能^[62]。图卷积网络的性能随着 K 从 0 到 1 增加而增加, 随着 K 继续增加而迅速下降。更深层的节点嵌入会导致过度平滑问题, 叠加多层消息传递可能会引入无信息、有噪声或有歧义的问题, 这在很大程度上影响了训练的效率和效果。为了解决这个问题, 通过利用图划分技术来减少输入网络的规模, 并构造压缩图。

为了构造压缩图 $G_c = (V_c, E_c)$, 首先得到 G 的一个分区 P , 表示为 $P = \{V_1, V_2, \dots, V_k\}$ 。图划分的目标是 (k, σ) 平衡, 其中 $0 < \sigma < 1$, 满足如下公式 3-8。

$$\max_{1 \leq i \leq k} |V_i| \leq (1 + \sigma) \left\lfloor \frac{|V|}{k} \right\rfloor \quad (3-8)$$

将切边的大小最小化为:

$$\bigcup_{1 \leq i, j \leq k} \{(v, u) \in E | v \in V_i, u \in V_j\} \quad (3-9)$$

通过对每个子图 $V' \in P$ 创建压缩节点和连接两个压缩节点 v_a, u_a 的权重为 $w(v_a, u_a)$ 的压缩边 $edge(v_a, u_a)$, 来构建一个压缩图 G_c 。 G_c 中压缩节点的数量为 G 的分区数, 压缩边的数量受边切割大小的限制。在关于 k 的选取上, 如果 k 很小, 那么从相应抽象节点继承 G 的每个节点的初始嵌入将失去有效性。另外, 如果 k 很大, 那么压缩图 G_c 也会很大, 相应的初始阶段成本也会增加。为了达到一个平衡, 设置 $k = \lceil \sqrt{|V|} \rceil$ 。

为了计算压缩图 G_c 的嵌入, 在每次迭代中, 每个节点 $v \in V$ 首先对 v 相邻节点的嵌入进

行聚合，得到平均嵌入 $e_{avg}(v)$ 然后，我们将 v 的嵌入更新为 e_{avg} 及其自身嵌入 $e_i v$ 的聚合。

基于上述压缩输入图进行偏好学习，将各层的嵌入进行串联，如公式 3-10 所示。

$$\begin{aligned}
 \hat{y}_{ui} &= \hat{y}_{ui}^{k-1} + \langle e_u^k, e_i^k \rangle \\
 &= \hat{y}_{ui}^{k-2} + \langle e_u^{k-1}, e_i^{k-1} \rangle + \langle e_u^k, e_i^k \rangle \\
 &= \hat{y}_{ui}^{k-2} + \langle e_u^{k-1}, e_i^{k-1} \rangle + \langle e_u^k, e_i^k \rangle \\
 &= \hat{y}_{ui}^0 + \langle e_u^1, e_i^1 \rangle + \dots + \langle e_u^k, e_i^k \rangle \\
 &= \langle e_u^0 || e_i^1 || \dots || e_u^k || e_i^1 || \dots || e_i^k \rangle
 \end{aligned} \tag{3-10}$$

(3) 模型学习：在贝叶斯个性化排名^[63]中采用了基于排名的损失函数，如公式 3-11 所示。

$$L_{main} = \sum_{a=1}^M \sum_{i,j \in D_a} -\ln(s(\hat{y}_{ai} - \hat{y}_{aj})) \tag{3-11}$$

其中， D_a 表示压缩图的维度。采用 Adam SGD^[64]优化器，并以小批量方式使用它。

3.1.4 目标优化模块设计

(1) 对比学习：

建立节点的增强视图后，将同一节点的视图视为正样本对，例如 $\{(h_u', h_u'') | u \in U\}$ ，而任意不同节点的视图被视为负样本对，例如 $\{(h_u', h_v'') | u, v \in U, u \neq v\}$ 。正样本对的辅助监督有助于保持同一节点的不同观点之间的一致性，而负样本对的监督则加强了不同节点之间的区分，如公式 3-12 所示。

$$L_{ssl}^{user} = \sum_{u \in U} -\log \frac{\exp(s(h_u', h_u'')/\tau)}{\sum_{v \in U} \exp(s(h_u', h_v'')/\tau)} \tag{3-12}$$

其中， τ 是温度系数， s 是余弦相似度函数，表示两个向量之间的相似度。同理，可以得到项目的相对损失函数 L_{ssl}^{item} 。最后，目标函数如公式 3-13 所示。

$$L_{ssl} = L_{ssl}^{user} + L_{ssl}^{item} \quad (3-13)$$

(2) 多任务训练：采用多任务训练策略，对经典推荐任务和自监督学习任务进行共同优化，如公式 3-14 所示。

$$L_{BPR} = L_{main} + \lambda_1 L_{ssl} + \lambda_2 ||W||^2 \quad (3-14)$$

其中， W 为 L_{main} 中的参数，超参数 λ_1 和 λ_2 分别用于调节自监督图学习和 L_2 正则化强度。

3.2 实验设计

3.2.1 实验设置

本实验在 Windows 11 操作系统上实施，Python3.x 为开发语言，Pytorch1.2 深度学习工具作为开发框架，Pycharm2.5 作为开发平台。

(1) 硬件环境

操作系统：Microsoft Windows 11 操作系统

内存：16GB 2400MHz DDR4

CPU：AMD Ryzen 9 7945HX with Radeon Graphics 2.50 GHz

(2) 软件环境

深度学习框架：Pytorch1.2

开发语言：Python3.8

(3) 数据集介绍

为了评估基于自监督学习的简化图神经网络协同过滤算法的性能，实验在 Amazon-Books、Gowalla 和 Yelp2018 基准数据集上进行，三个数据集的统计数据总结如表 3-1 所示。

表 3-1 Amazon-Books、Gowalla、Yelp2018 数据集

Dataset	#Users	#Items	#Interactions	Density
Amazon-Books	52643	91599	2984108	0.062%
Gowalla	29858	40981	1027370	0.084%
Yelp2018	31668	38048	1561406	0.130%

Amazon-Books 数据集：Amazon-Books 数据集是由亚马逊网站上的图书评分数据构建

而成的，其中包含了用户对图书的评分信息以及图书信息（如作者、出版社等）。实验中，使用与 LightGCN[44]相同的 10 核设置来确保每个用户和项目至少有 10 次交互。

Gowalla 数据集：作为一个地理位置数据集，Gowalla 数据集包含了用户在社交网络平台 Gowalla 上的地理位置签到数据。每个签到数据记录了用户在特定时间和地点进行签到的信息，使用 10 核设置来确保数据质量。

Yelp2018 数据集：Yelp2018 数据集是由 Yelp 网站上的用户评论数据构建而成的，其中包含了用户对商家的评论、评分以及商家信息（如行业类别、地理位置等）。同样，采用 10 核设置，以保证每个用户和项目至少有 10 次交互。

在表 3-1 中，分别统计了三个数据集的用户数量，项目数量和用户与项目的交互次数。计算得到数据集密度（用户-项目评分矩阵中的非空条目与总条目数之比）。观察可知，Amazon-Books 数据集中的交互次数远大于其他两个数据集，Yelp2018 数据集拥有更加稠密的数据。

(4) 评价指标介绍

实验中主要关注 $Top-k$ 推荐任务，评价指标选用了推荐系统中常用的 $Recall@K$ 和 $NDCG@K$ ， K 值设定为 20。模型性能评估时，将未与用户有交互的测试集项目被视为负样本。

3.2.2 对比算法

对比算法包括基于图神经网络方法的算法，如 NGCF[42]、LightGCN[44]、DGCF[65]；基于矩阵分解方法的算法，如 GRMF[66]、BPR-MF[63]；网络嵌入方法的算法，如 Deep-Walk[67]、Node2Vec[68]。算法具体描述如下：

NGCF[42]：利用图卷积网络对用户-项目交互图进行建模，进行节点嵌入学习。使用多层图卷积网络层叠信息传播，实现高阶连接建模。

Light-GCN[44]：简化了原始的 GCN 模型，去除了特征变换和非线性激活函数，引入残差连接和层归一化，加速训练过程，提高模型泛化能力。

DGCF[65]：利用双层图卷积网络分别对用户行为图和知识图进行建模，提出了基于图注意力的交互建模机制，可以识别重要的用户-商品历史交互。

GRMF[66]：图的正则化使得连接的用户或项目具有相似的潜在特征表示。对矩阵分解引入图约束，学习到的用户和项目特征更加平滑连续。

BPR-MF[63]：基于 BPR 优化框架，直接以 Item 之间的排序关系作为训练信号，将矩阵

分解作为排序的预测模型，进行个性化的参数学习。

Deep-Walk^[67]：节点是从图上的随机游走中采样的，通过预测节点的局部邻域来学习嵌入。

Node2Vec^[68]：一种优秀的图表示学习方法，通过随机游走来获取到网络中的邻近度，然后将所有节点映射低维表示空间中。

3.2.3 参数设置

在模型训练上，将所有模型的嵌入向量大小固定为 64，采用 Adam 优化器来进行优化，并使用 Xavier 方法初始化嵌入参数^[69]，使用默认学习率 0.001 和默认小批量大小 1024（为了提高训练速度，在 Amazon-Books 上将批量大小增加到 2048）。将 L_2 正则化强度的超参数 λ_2 设定为 LightGCN^[44] 建议的最优值 $1e^{-4}$ ，将自监督图学习参数 λ_1 调优范围设定在 $\{0.01, 0.001, 1e^{-4}, 1e^{-5}, 1e^{-6}\}$ ，按照 SGL^[60] 的经验将温度系数 τ 和概率 ρ 分别设定为 0.5 和 0.1，早期停止和验证策略与 LightGCN 相同。

3.3 实验结果与分析

3.3.1 SSGCF 模型训练 Loss 曲线

如图 3-3 所示，在 Amazon-Books 数据集上探究了 SSGCF 的训练随着 epoch 的增加损失函数的变化情况。不难发现 SSGCF 算法在 epoch 取值 500 时 loss 几乎已经不再变化，曲线收敛，故在接下来的对比实验中的训练轮次选取为 500。

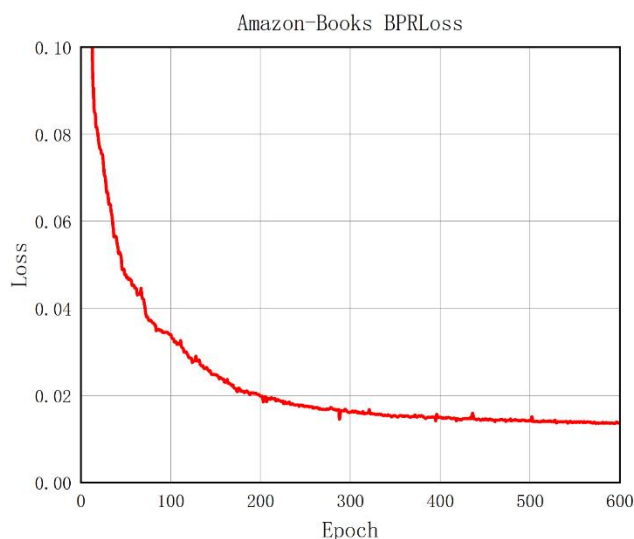


图 3-3 SSGCF 深度学习 Loss 曲线

3.3.2 SSGCF 的 λ_1 参数实验

在 Amazon-Books 与 Yelp 数据集上, 探究自监督图学习参数 λ_1 的取值对于 SSGCF 算法表现的影响, λ_1 的取值从 0.01、0.001、 $1e^{-4}$ 、 $1e^{-5}$ 、 $1e^{-6}$ 之间进行实验, 对用户进行 $Top20$ 推荐, 选取了 $Recall@20$ 、 $NDCG@20$ 两种评价指标进行模型评估。如表 3-2 所示, λ_1 的取值为 0.001 的时候 $Recall@20$ 、 $NDCG@20$ 的值最高。 λ_1 取值小于 0.001 的时候, 随着 λ_1 值增大算法性能基本不变。 λ_1 取值大于 0.001 的时候, 算法性能迅速下降。

由于 SSGCF 中存在领域聚合, SSGCF 随着 λ_1 变化的变化不明显, 不容易过拟合并且具备出色的性能。因此, 将自监督图学习参数 λ_1 定为 0.001。

表 3-2 不同 λ_1 的值对于 SSGCF 实验结果影响

Dataset	Amazon-Books		Yelp	
λ_1	Recall@20	NDCG@20	Recall@20	NDCG@20
1E-6	0.0485	0.0383	0.0657	0.0543
1E-5	0.0487	0.0384	0.0657	0.0543
1E-4	0.0488	0.0384	0.0657	0.0542
0.001	0.0493	0.0387	0.0666	0.0551
0.01	0.0074	0.0059	0.0116	0.0093

3.3.3 SSGCF 与 LightGCN 在不同传播层的性能比较

在 Amazon-Books 与 Yelp 数据集上, 探究 SSGCF 与 LightGCN 在不同传播层对性能表现的影响, 如表 3-3 所示。

表 3-3 将 SSGCF 与 LightGCN 在不同传播层数上进行对比实验 ($\lambda_1 = 0.001$)

Dataset		Amazon-Books		Yelp	
Layers	Algorithm&Improvement	Recall@20	NDCG@20	Recall@20	NDCG@20
1 layer	LightGCN	0.0392	0.0302	0.0639	0.0521
	SSGCF	0.0454	0.0364	0.0603	0.0496
	%Improv	15.8%	20.5%	-5.6%	-4.8%
2 layers	LightGCN	0.0373	0.0289	0.0628	0.0505
	SSGCF	0.0475	0.0378	0.0639	0.0526
	%Improv	27.3%	30.8%	1.8%	4.2%
3 layers	LightGCN	0.0345	0.0267	0.0581	0.04685
	SSGCF	0.0484	0.0382	0.0659	0.0543
	%Improv	40.3%	43.1%	13.4%	15.9%
4 layers	LightGCN	0.0319	0.0246	0.0543	0.0437
	SSGCF	0.0493	0.0387	0.0666	0.0551
	%Improv	54.5%	57.3%	22.7%	26.1%

在大多数情况下, SSGCF 比 LightGCN 有显著改进, 尤其是 Amazon-Books 数据集上。可以发现, SSGCF 的性能在一到四层逐层上升。层数为 1 时, 在 Yelp 数据集上, SSGCF 的性能不如 LightGCN, 随着层数增长, 在层数为 4 时, SSGCF 的性能达到最好, 相比 LightGCN 提升明显。

原因在于简化的嵌入传播机制不起作用并且退化为与 LightGCN 使用的类似的领域聚合模式。然而, 随着层数的增加, SSGCF 随着层数增加, 其优势变得明显并且性能有了极大提高。

3.3.4 SSGCF 噪声实验

为了探究噪声对提出模型的影响, 将 SSGCF 与 LightGCN 在 Yelp2018 和 Amazon-Books 上进行对比实验, 使用召回率和性能下降率进行评估, 验证模型使用自监督图学习的鲁棒性。通过添加一定比例的用户未交互的项目, 作为对抗样本来污染数据集。

如图 3-4、图 3-5 所示, 展示了 SSGCF 与 LightGCN 的实验结果。

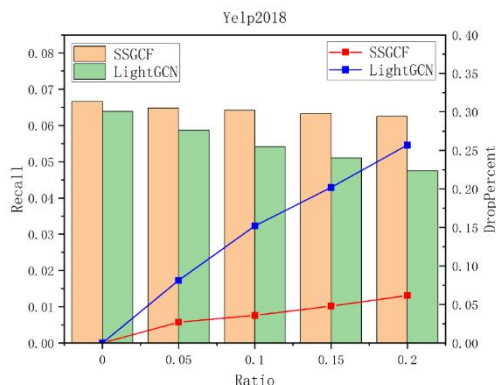


图 3-4 SSGCF 与 LightGCN 在 Yelp 数据集上召回率和丢失率表现

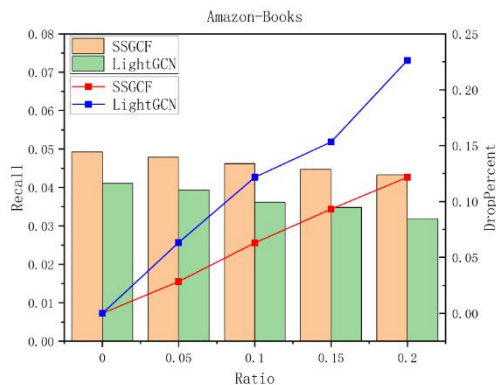


图 3-5 SSGCF 与 LightGCN 在 Amazon-Books 数据集上召回率和丢失率表现

其中横坐标表示噪声数据所占比例，纵坐标柱状图表示 $Recall@20$ ，而折线图表示模型性能下降率。

(1) 向数据集中添加噪声数据，SSGCF 和 LightGCN 模型性能均有下降，在 Amazon-Books 数据集上尤其模型。这是由于 Amazon-Books 数据集比 Yelp2018 数据集更加稀疏，添加噪声数据会使得数据集稀疏性加剧，并且进一步混淆数据，使得模型更难以从中提取有效信息，从而影响模型的性能。

(2) 然而随着噪声数据的比例增加，SSGCF 性能下降程度小于 LightGCN，并且两者之间差距越来越大。这表明，SSGCF 采用更为简单直观的自监督图学习任务，对数据的特征提取更加鲁棒，注重学习数据的内在特征表示，而不是过度拟合数据的噪声。因此，能够更好地过滤掉噪声数据的影响。

(3) 在 Yelp2018 数据集上，SSGCF 即使在噪声干扰占据 20% 的情况下，其召回率仍然与未受噪声干扰的 LightGCN 持平。在 Amazon-Books 数据集上，SSGCF 在噪声干扰影响下，全面领先未受干扰的 LightGCN。这表明，SSGCF 具有很好的鲁棒性。

3.3.5 对比实验

在 Amazon-Books、Gowalla 以及 Yelp 三个数据集上，使用 $Recall@20$ 和 $NDCG@20$ 两种评价指标，评估 SSGCF 与对比算法的性能。实验结果如表 3-4 所示。观察结果与原理分析如下：

1) SSGCF 在所有三个数据集上始终优于其他对比算法。特别是，在 Amazon-Books 数据集上，SSGCF 评价指标 $Recall@20$ 和 $NDCG@20$ 的表现对比算法中基于图神经网络的最强基线分别大幅提高了 16.82%和 19.44%。提出的 SSGCF 模型始终优于 LightGCN，显示了在自监督图学习下，通过数据增强和对比学习对用户偏好进行建模的有效性。与其他基线相比，SSGCF 的核心部分一种线性模型，易于训练可扩展到大型数据集，可以通过计算压缩图上的网络嵌入，得到图中各节点初始化嵌入，缓解了交互数据稀疏的图卷积聚合操作中的过平滑问题，这些优势共同导致了 SSGCF 与其他先进模型相比的优越性。

2) 在基线算法中，LightGCN 和 DGCF 大幅优于所有其他基线，这表明非线性变换对于推荐系统的表示学习模型是无用的，因为推荐系统除了用户和项目的 ID 嵌入之外没有其他参数，从侧面说明了 SSGCF 核心部分使用线性模型的合理性。

3) 在对比算法中网络嵌入模型的表现远不如基于图卷积网络的模型，尤其是在

Gowalla 数据集上。原因可能是，在许多网络嵌入方法中，强大的图卷积比传统的随机游走更有效地捕获协作信息以进行推荐。SSGCF 对传统图卷积模型的改进，性能有着显著提升。

表 3-4 整体性能比较

Dataset	Amazon-Books		Gowalla		Yelp	
Algorithm	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
NGCF	0.0344	0.0265	0.1570	0.1327	0.0579	0.0477
LightGCN	0.0411	0.0315	0.1830	0.1554	0.0649	0.0530
DGCF	0.0422	0.0324	0.1842	0.1561	0.0654	0.0534
GRMF	0.0354	0.0270	0.1477	0.1205	0.0571	0.0462
BPR-MF	0.0338	0.0261	0.1616	0.1366	0.0549	0.0445
DeepWalk	0.0346	0.0264	0.1034	0.0740	0.0476	0.0378
Node2Vec	0.0402	0.0309	0.1019	0.0709	0.0452	0.0360
SSGCF	0.0493	0.0387	0.1849	0.1589	0.0666	0.0551

3.4 本章小结

本章第一节分析了当前研究中仍然存在的问题，介绍了提出算法的研究动机；第二节是对提出算法模型的具体设计与实现，详细介绍了将自监督图学习与简化图神经网络协同过滤模型相结合的过程；第三节对实验进行了详尽的设计，简要地介绍了对比算法以及参数设置；第四节从实验结果和方法原理上，说明了提出算法的合理性与优越性。

4 基于自监督学习的邻居感知注意力图神经网络协同过滤算法

现代推荐算法的核心是学习用户和项目的信息表示（又名嵌入），传统的协同过滤推荐算法通常只考虑用户和项目之间的历史交互数据，而忽略了用户之间的社交关系、项目之间的相似性等丰富的图结构信息。近些年来，图神经网络作为一种能够有效处理图数据的技术，具有很强的表征学习能力，能够全面地利用图结构信息，包括用户-用户、项目-项目之间的关系，以及更高阶的关系，从而更准确地捕捉用户和项目之间的关联，提高推荐的准确性和个性化程度。

以往的研究中，由于没有对邻域中的关系信息进行建模，导致邻域之间的隐式相关性被忽视。物以类聚，人以群分。用户在做决定的时候，除了受到自身对项目偏好影响的同时，也会受到邻居用户的影响。例如，在用户-项目交互图中，用户的偏好可以由所有相邻的项目表示。另外，在传统的基于图神经网络的协同过滤方法中，用户和项目的特征向量取决于他们之间的互动。这些用户和项目节点本身并不具备任何内在的语义信息，它们的表达在一开始只是作为识别符，用以区分不同的节点。因此，当传统图卷积网络协同过滤方法进行特征转换时，可能增加不必要的计算复杂性，使得模型的训练过程更加艰难，难以实现最佳表现。此外，数据集噪声问题对于算法性能的消极影响仍然不可忽视。

为了应对上述问题，提出一种基于自监督学习的邻居感知注意力图神经网络协同过滤算法 (NASGCF, Neighbor-Aware Attention Graph Neural Network Collaborative Filtering Based on Self-supervised Learning)，来捕获推荐任务中节点邻居之间的隐式相关性。SSGCF 与 NASGCF 两个模型均为在自监督图学习框架下对 LightGCN 模型的改进，但是侧重点有所不同。SSGCF 强调对模型的进一步简化，而 NASGCF 更加注重对邻域关系信息进行建模。具体来说，引入了邻居感知图注意层来计算用户的相邻项目和项目的相邻用户的邻居感知注意系数，邻居感知注意系数在图神经网络的主用作聚合系数。为了简化计算过程，通过对邻居之间每个成对注意力函数的结果进行平均来计算注意力系数。使用邻居嵌入和注意力系数的线性组合作为每个节点的下一层嵌入。使用第一跳邻居后，进一步堆叠更多邻居感知图注意层以收集来自高阶邻居的有影响力的信号。另外，为了简化模型结构，减少参数数量和计算复杂度，使得模型更加轻量化和高效，在模型中删除了被证实对协同过滤有负面影响^[44]的特征变换和非线性激活。此外，仍然使用自监督图学习算法架构与邻居感知注意力图神经网络协同过滤算法相结合，降低噪声数据对于模型训练的干扰，有利于提高模型整体的鲁棒性和准确性。

4.1 模型设计与描述

NASGCF 模型中主要包括四个部分，如图 4-1 所示。

输入模块：将用户和项目交互的图结构作为模型的输入；

数据增强模块：对输入数据通过节点丢弃、边丢弃和随机游走的方式生成增强数据；

图神经网络编码器：使用邻居感知注意力图神经网络协同过滤模型，对初始模型数据和增强数据进行图卷积操作，生成节点表示矩阵；

目标优化模块：首先进行有监督学习，使用初始模型数据中的监督信号；其次，通过增强数据构造的辅助监督信号，进行对比学习；最终，确定总体优化目标。

由于自监督图学习中的输入模块与数据增强模块在第三章已经详细叙述，因此本章将重点介绍图神经网络编码器和目标优化模块部分。

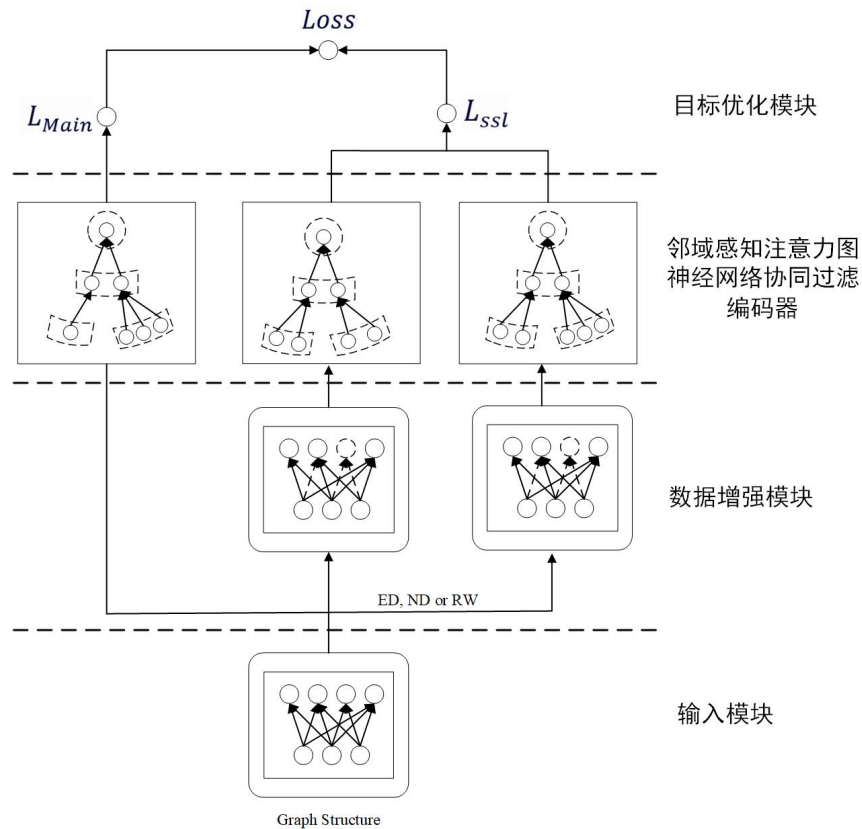


图 4-1 NASGCF 模型架构

模型的流程如下：首先，将图结构数据通过节点丢弃、边丢弃、随机游走的方式为每个节点生成多个视图，进行数据增强。接着使用一个可训练的嵌入查找表来构建消息传播的嵌入层，计算邻居感知注意系数得到当前节点的邻居感知表示，进一步堆叠更多的邻居

感知图注意层，从高阶邻居中收集有影响的信号。使用比较学习提高同一节点在不同视图中的 consistency，并采用多任务训练策略联合优化推荐任务和自监督学习任务。

4.1.1 图神经网络编码器设计

图神经网络编码器使用邻居感知注意力图神经网络协同过滤模型，作为 NASGCF 模型的核心部分，如图 4-2 所示。

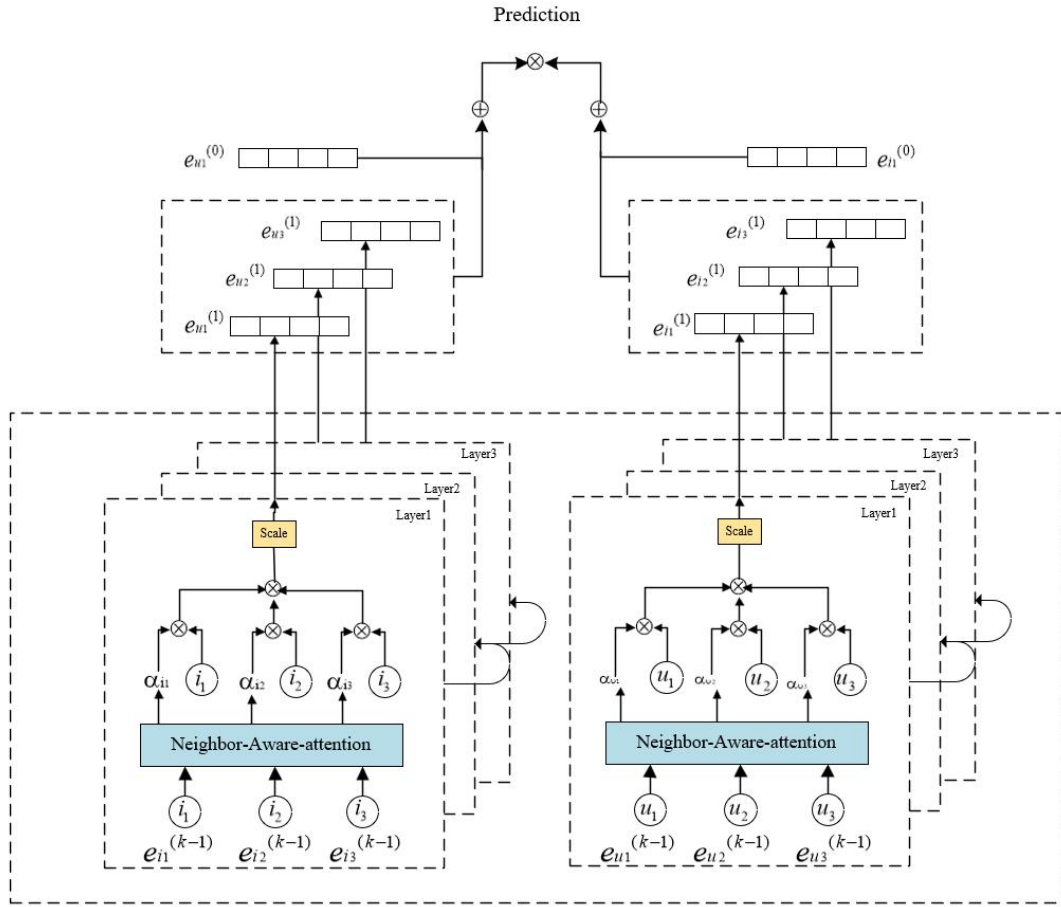


图 4-2 NASGCF 模型核心部分——邻居感知注意力图神经网络协同过滤模型

(1) 邻居感知图注意层：在用户-项目交互图的推荐任务中，使用与用户的所有相邻项目之间的相关性来表示项目对用户的重要程度。因此，如图 4-3 所示，定义了一种新颖的邻居感知图注意机制，通过第 k 层上的隐式相关性来计算邻居感知注意力系数，如公式 4-1，公式 4-2 所示。

$$a_{i \rightarrow u}^{(k-1)} = \frac{1}{|N_u|} \sum_{j \in N_u} RELU\left(\cos\left(e_i^{(k-1)}, e_j^{(k-1)}\right)\right) \quad (4-1)$$

$$a_{u \rightarrow i}^{(k-1)} = \frac{1}{|N_i|} \sum_{v \in N_i} RELU\left(\cos\left(e_u^{(k-1)}, e_v^{(k-1)}\right)\right) \quad (4-2)$$

其中, $a_{i \rightarrow u}^{(k-1)}$ 为 i 的注意力系数聚合为 u , N_u 和 N_i 是图中用户和项目的邻居数。为了简化计算过程, 这里对每个成对注意力函数计算的结果进行平均。通过上式, 可以得到消息传播过程中每个节点的邻居感知关注系数, 如公式 4-3, 公式 4-4 所示。

$$e_u^{(k)} = \frac{1}{\sqrt{|N_u|}} \sum_{i \in N_u} a_{i \rightarrow u}^{(k-1)} \cdot e_i^{(k-1)} \quad (4-3)$$

$$e_i^{(k)} = \frac{1}{\sqrt{|N_i|}} \sum_{u \in N_i} a_{u \rightarrow i}^{(k-1)} \cdot e_u^{(k-1)} \quad (4.4)$$

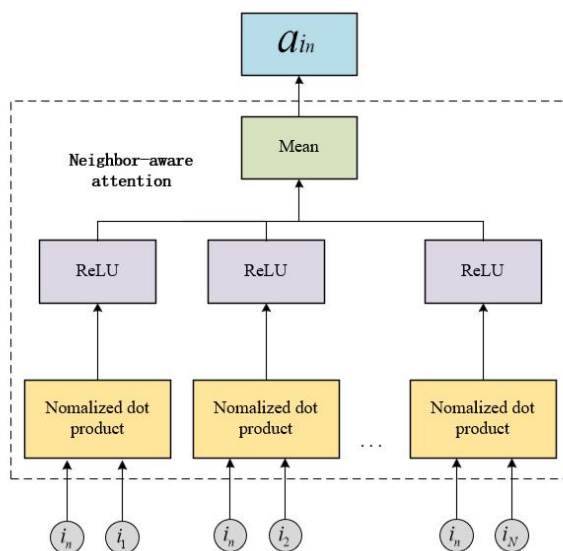


图 4-3 邻居感知图注意机制

(2) 层组合：使用第一跳邻居后，进一步堆叠更多邻居感知图注意层以收集来自高阶邻居的有影响力的信号。在 K 层之后，将不同层的嵌入相加作为最终的嵌入，如公式 4-5

所示：

$$e_u^* = \sum_{k=0}^K e_u^{(k)}, e_i^* = \sum_{k=0}^K e_i^{(k)} \quad (4-5)$$

(3) 采样策略：在之前的工作中^[44]，消息传播采用了完整的邻域集合。由于真实的推荐情景数据集中含有大量的用户和项目，使用完整邻域集不利于模型的训练效率。因此，执行 Max-M 子图采样策略，即在采样的子图中，每个节点的邻居数量不超过 M 。例如，如图 4-4 所示， k 表示条数，将采样用户标记为灰色，将采样项目标记为蓝色。假设 $M = 3$ ， u_1 和 i_2 有 4 个邻居，随机选择 3 个邻居 $\{i_1, i_2, i_3\}$ 和 $\{u_1, u_2, u_{10}\}$ ， i_1 的邻居数小于 3， i_3 邻居数等于 3，因此选择它们的所有邻居。

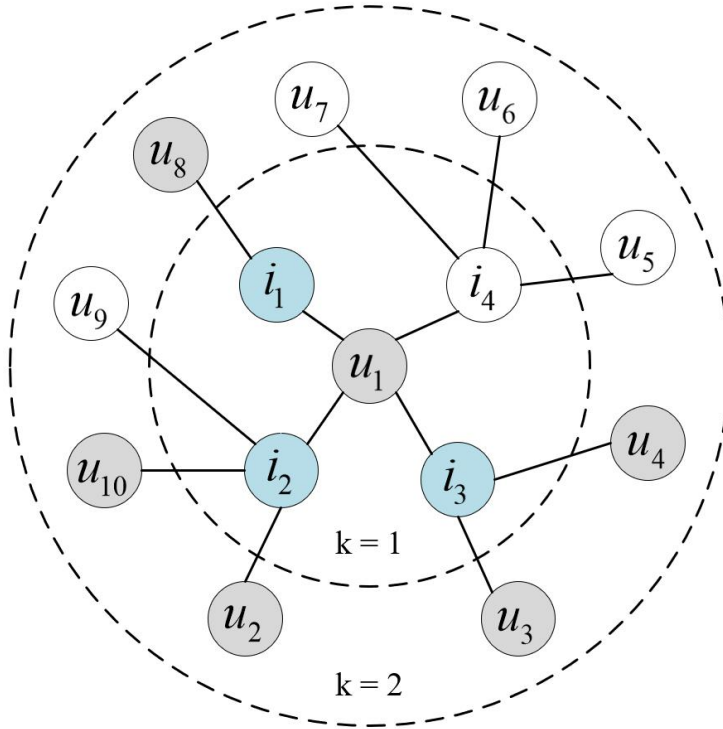


图 4-4 Max-M 采样策略

(4) 模型学习：通过邻居感知注意力机制嵌入传递和聚合后，得到 e_u^* 和 e_i^* 。通过内积来预测用户和项目之间的匹配程度，如公式 4-6 所示。

$$\hat{y}_{ui} = e_u^{*T} e_i^* \quad (4-6)$$

为了考虑观察到的和未观察到的交互之间的相对顺序，采用贝叶斯个性化排名损失对模型进行优化。

$$L_{main} = \frac{1}{|\mathcal{B}|} \sum_{(u,i,j) \in \mathcal{B}} -\sigma(\hat{y}_{ui} - \hat{y}_{uj}) \quad (4-7)$$

其中， $\mathcal{B} \subseteq \{(u, i, j) | (u, i) \in \mathcal{R}^+, (u, j) \in \mathcal{R}^-\}$ 表示小批量的采样数据， \mathcal{R}^+ 表示观察到的相互作用， \mathcal{R}^- 表示未观察到的相互作用， σ 是 $Softplus$ 激活函数。

4.1.2 目标优化模块设计

(1) 对比学习：

建立节点的增强视图之后，将同一节点的视图视为正对，如 $\{(h'_u, h''_u) | u \in U\}$ ，以及任意不同节点的视图都是负对，如 $\{(h'_u, h''_v) | u, v \in U, u \neq v\}$ 。对比损失函数，如公式 4-8 所示。

$$L_{ssl}^{user} = \sum_{u \in U} -\log \frac{\exp(s(h'_u, h''_u)/\tau)}{\sum_{v \in U} \exp(s(h'_u, h''_v)/\tau)} \quad (4-8)$$

其中， τ 是温度系数， s 是余弦相似度函数，表示两个向量之间的相似度。类似地，可以得到项目的相对损失函数 L_{ssl}^{item} 。最终，可以得到自监督任务的目标函数为公式 4-9。

$$L_{ssl} = L_{ssl}^{user} + L_{ssl}^{item} \quad (4-9)$$

(2) 多任务训练：采用多任务训练策略，对经典推荐任务和自监督学习任务进行共同优化，如公式 4-10 所示。

$$L_{BPR} = L_{main} + \lambda_1 L_{ssl} + \lambda_2 \|\Theta\|_2^2 \quad (4-10)$$

其中， Θ 为 L_{main} 中的参数， $\Theta = \{e_u^{(0)}, e_i^{(0)} | u \in U, i \in I\}$ ， λ_1 和 λ_2 分别是控制自监督图学习和 L_2 正则化强度的超参数。

4.2 实验设计

4.2.1 实验设置

本实验在 Windows 11 操作系统上实施，Python3.x 为开发语言，Pytorch1.2 深度学习工具作为开发框架，Pycharm2.5 作为开发平台。

(1) 硬件环境

操作系统：Microsoft Windows 11 操作系统

内存：16GB 2400MHz DDR4

(2) CPU：AMD Ryzen 9 7945HX with Radeon Graphics 2.50 GHz

软件环境

深度学习框架：Pytorch1.2

开发语言：Python3.8

(3) 数据集介绍

为了评估基于自监督学习的简化图神经网络协同过滤算法的性能，实验在 Amazon-Books、Gowalla 和 Yelp2018 基准数据集上进行，三个数据集的统计数据总结如表 3-1 所示。

表 3-1 Amazon-Books、Gowalla、Yelp2018 数据集

Dataset	#Users	#Items	#Interactions	Density
Amazon-Books	52643	91599	2984108	0.062%
Gowalla	29858	40981	1027370	0.084%
Yelp2018	31668	38048	1561406	0.130%

Amazon-Books 数据集：Amazon-Books 数据集是由亚马逊网站上的图书评分数据构建而成的，其中包含了用户对图书的评分信息以及图书信息（如作者、出版社等）。实验中，使用与 LightGCN[44]相同的 10 核设置来确保每个用户和项目至少有 10 次交互。

Gowalla 数据集：作为一个地理位置数据集，Gowalla 数据集包含了用户在社交网络平台 Gowalla 上的地理位置签到数据。每个签到数据记录了用户在特定时间和地点进行签到的信息，使用 10 核设置来确保数据质量。

Yelp2018 数据集：Yelp2018 数据集是由 Yelp 网站上的用户评论数据构建而成的，其中包含了用户对商家的评论、评分以及商家信息（如行业类别、地理位置等）。同样，采用 10 核设置，以保证每个用户和项目至少有 10 次交互。

(4) 评价指标介绍

实验中主要关注 $Top-k$ 推荐任务，评价指标选用了推荐系统中常用的 $Recall@K$ 和 $NDCG@K$ ， K 值设定为 20。模型性能评估时，将未与用户有交互的测试集项目被视为负样本。

4.2.2 对比算法

本实验对比算法包含 SSGCF 和第三章中的对比算法，还加入了基于图神经网络方法的算法，如 NIA-GCN^[70]、LR-GCCF^[71]；基于矩阵分解方法的算法，如 ENMF^[72]；基于变分自动编码器的隐反馈协同过滤方法的算法 Mult-VAE^[73]。算法具体描述如下：

NIA-GCN^[70]：通过双线性邻域聚合器对邻域之间的交互与元素级乘积进行建模，使用门控循环单元结合时间编码来捕捉用户的动态兴趣。

LR-GCCF^[71]：利用图卷积网络对用户-商品图进行建模，基于随机游走，定义了符合推荐场景的归一化图卷积操作。

ENMF^[72]：一种用于社交感知推荐的高效自适应迁移神经网络，通过联合网络嵌入和矩阵分解，实现了兼顾推荐效果和解释性的图基推荐算法。

Mult-VAE^[73]：一种变分自编码器模型，用于生成多模态数据。通过联合变分编码器框架进行多模态表示学习，实现了模态解耦和生成。

4.2.3 参数设置

在模型训练上，将所有模型的嵌入向量大小固定为 64，采用 Adam 优化器来进行优化，并使用 Xavier 方法初始化嵌入参数^[69]，使用默认学习率 0.001 和默认小批量大小 1024（为了提高训练速度，在 Amazon-Books 上将批量大小增加到 2048）。对于 1 跳到 4 跳的采样，在数据集上将最大采样邻居数设置为 {20, 40, 60, 80, 100, 120}。将 L_2 正则化强度的超参数 λ_2 设定为 LightGCN^[44] 建议的最优值 $1e^{-4}$ ，将自监督图学习参数 λ_1 调优范围设定在 {0.01, 0.001, $1e^{-4}$, $1e^{-5}$, $1e^{-6}$ }，按照 SGL^[60] 的经验将温度系数 τ 和概率 ρ 分别设定为 0.5 和 0.1，早期停止和验证策略与 LightGCN 相同。

4.3 实验结果与分析

4.3.1 模型训练 Loss 曲线与 Recall 曲线

为了进一步探究邻域感知注意力机制对于模型训练效率的影响，在其他设置均相同情况下，对 NASGCF 和 SSGCF 模型使用 Yelp2018 数据集对模型在不同 epoch 上进行性能测试的实验，结果如图 4-5，图 4-6 所示。

相比较 SSGCF, NASGCF 在 $Loss$ 值和 $Recall@20$ 上的性能都更加优秀, 说明 NASGCF 训练数据的拟合效果较好, 邻域感知注意力机制可以有效提升模型性能。

在 $Loss$ 训练结果中, 由于 NASGCF 模型结构更复杂, 训练难度更大, 因此 SSGCF 在初期的拟合速度 $Loss$ 值方面具有一定优势。

在 $Recall@20$ 评价指标的训练结果上, NASGCF 相较于 SSGCF, 虽然在起初的训练中落后于 SSGCF, 但是其训练效果的上升趋势十分明显, 并且在 100epoch 时反超了 SSGCF。推测是因为 NASGCF 训练初期需要邻域感知层进行非线性变换, 在该部分可能存在一些扰动因素对模型性能的提升产生一些影响, 并且随着训练 epoch 的增加, 注意力机制和邻域感知模块逐渐发挥作用, 可以更好地建模用户的潜在兴趣和用户之间的高阶关系, 从而验证了邻域感知注意力机制设计的有效性, 其潜力需要充足的训练才能发挥出来。

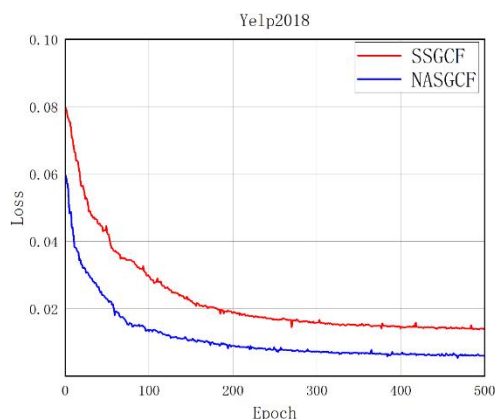


图 4-5 SSGCF 和 NASGCF 在 Yelp2018 数据集上深度学习 Loss 曲线

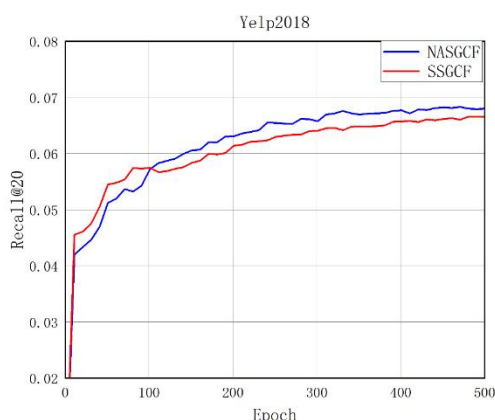


图 4-6 SSGCF 和 NASGCF 在 Yelp2018 数据集上深度学习 Recall@20 训练曲线

4.3.2 NASGCF 的 λ_1 参数实验

在 Yelp2018 和 Amazon-Books 上研究了传播层数为四层网络时不同 λ_1 参数值对于 NASGCF 的性能影响, 如图 4-7, 4-8 所示。

当 λ_1 不大于 0.001 时, NASGCF 的性能没有显著变化, 而当 λ_1 设置为 0.001 时, NASGCF 达到最佳性能, 因此三个数据集的最优值为 0.001 时。

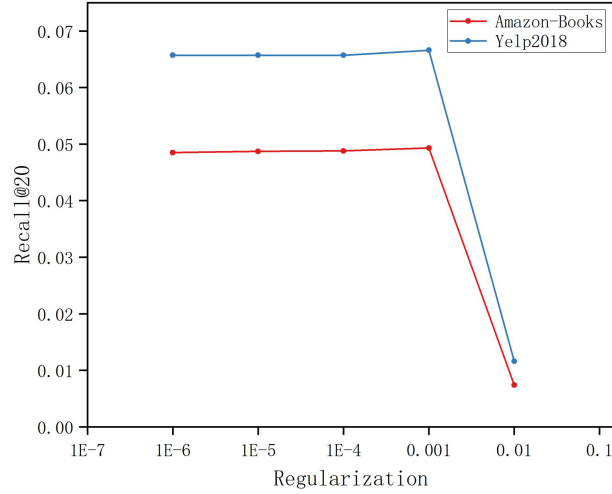


图 4-7 不同 λ_2 参数值对于 NASGCF 的 *Recall@20* 的影响

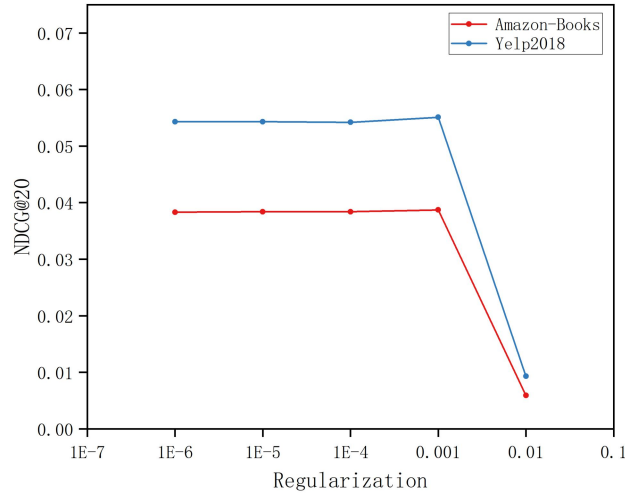


图 4-8 不同 λ_2 参数值对于 NASGCF 的 *NDCG@20* 的影响

4.3.3 NASGCF 邻居数实验

采用 Max-M 采样策略来有效地训练模型。对于具有不同稀疏度的数据集, 选择适当数量的采样邻居非常重要。如图 4-9, 图 4-10 所示, 在 Yelp2018 上, NASGCF 的效果随着

M 的增加而增加，并且在 M 大于 100 后改善并没有明显放缓，而在 Amazon-Books 上，当 M 从 20 增加到 40 时，NASGCF 模型的效果得到了显着的改善，但是当 M 大于 40 时，NASGCF 模型的效果不再改变。

由于 Yelp2018 比 Amazon-Books 更密集，这意味着每个用户和项目有更多的邻居。因此我们的实验结果表明，合适的 M 与数据集的密度密切相关。在稀疏数据集上，即使采样的邻居很少，NASGCF 也可以实现显着的改进。

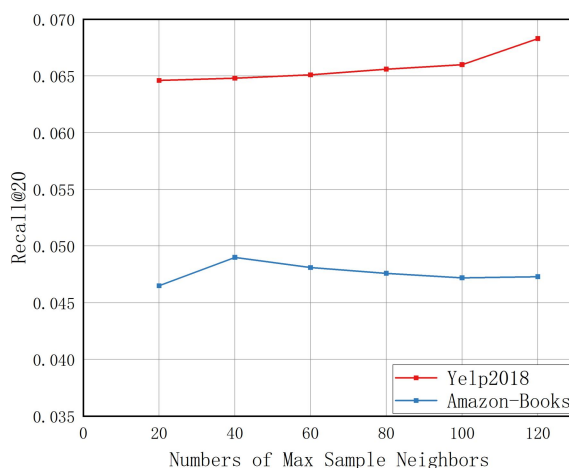


图 4-9 最大邻居数对于 NASGCF 的 Recall@20 的影响

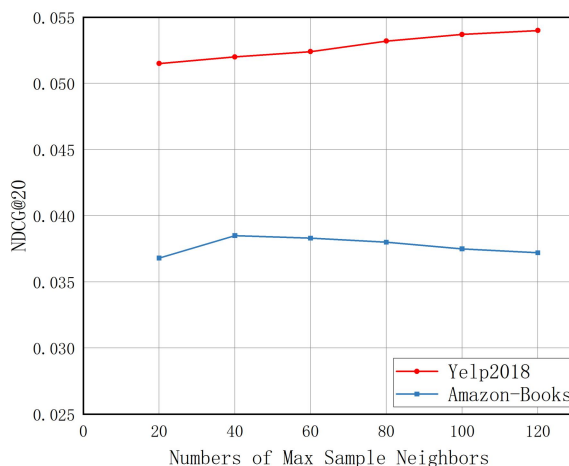


图 4-10 最大邻居数对于 NASGCF 的 NDCG@20 的影响

4.3.4 NASGCF 不同变体的消融实验

为了验证自监督学习和邻域感知注意力机制的有效性，进行了消融实验，探究 Gowalla 和 Yelp 数据集下，NASGCF 算法自监督学习和邻域感知注意力机制对于模型的作用，实

验结果如图 4-11, 图 4-12 所示。NASGCF-S 将 NASGCF 中的自监督学习替换为 LightGCN 中使用的传统方法, 实验结果表明 $Recall@20$ 和 $NDCG@20$ 评价指标下降。这表明自监督学习能够有效地过滤数据噪声, 缓解长尾问题。另一方面, 从 NASGCF 中移除邻域感知注意力机制并使用固定层组合权重的 NASGCF-N 与 NASGCF 相比, 性能有所下降。这表明邻域感知注意力机制可以捕获节点邻居之间的隐含相关性, 用于自适应地为用户和物品分配合适的层权重。因此, 证明了自监督学习和邻域感知注意力机制的有效性。

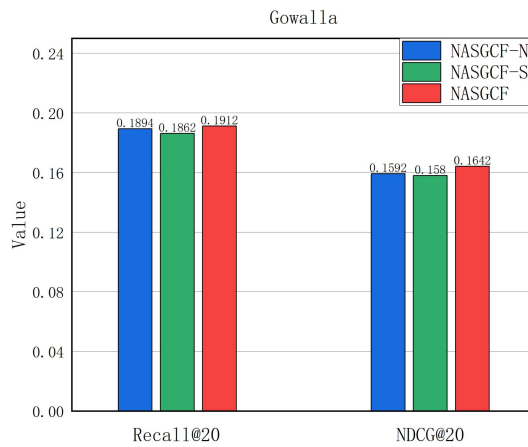


图 4-11 在 Gowalla 数据集上不同变体的对比实验

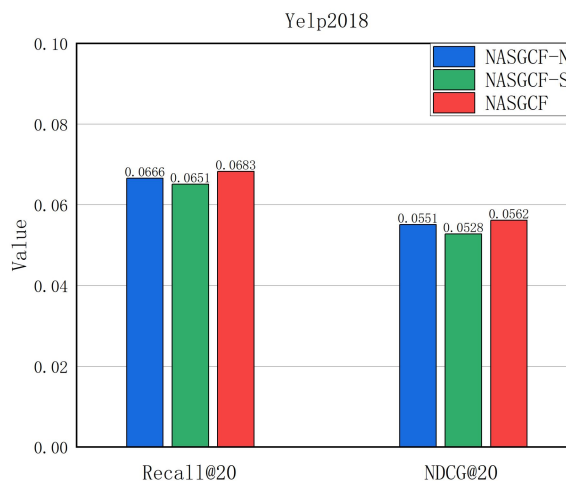


图 4-12 在 Yelp 数据集上不同变体的对比实验

4.3.5 对比实验

在 Amazon-Books、Gowalla 以及 Yelp 三个数据集上，使用 $Recall@20$ 和 $NDCG@20$ 两种评价指标，评估 NASGCF 与对比算法的性能。实验结果如表 4-1 所示。观察结果与原理分析如下：

(1) 提出的 NASGCF 在三个数据集上始终优于其他对比算法。相对对比算法中的最好表现， $Recall@20$ 在 Amazon-Books、Gowalla 和 Yelp2018 上分别提升 18%、3.8%和 4.43%， $NDCG@20$ 在 Amazon-Books、Gowalla 和 Yelp2018 上分别为 22.2%、5.19%和 5.24%，证明了 NASGCF 的优越性。特别是在大型数据集上，NASGCF 的性能提升明显，反映了模型具有更强的泛化能力、学习能力和稳定性，能够更好地适应不同的数据情况和应用场景。与 SSGCF 相比较，算法性能尽管略有提升，也证明了邻居感知注意机制的有效性，NASGCF 能够以显式的方式探索邻居之间的隐式相关性，提高算法的性能。

(2) 尽管 NIA-GCN 在建模节点之间关系时考虑了邻域交互的影响，提出通过逐位计算嵌入对相邻节点交互进行成对显式编码。然而，NIA-GCN 没有研究邻居之间的相关性对聚合系数的影响，并且仍然采用非线性变换，这在 LightGCN^[44]中已经被证明对协同过滤有负面影响。NASGCF 工作重点则是通过对于节点的不同邻居之间的隐式相关性进行建模，以便给这些邻居赋予不同的权重。

表 4-1 整体性能比较

Dataset	Amazon-Books		Gowalla		Yelp	
Algorithm	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
NGCF	0.0344	0.0265	0.1570	0.1327	0.0579	0.0477
Mult-VAE	0.0407	0.0315	0.1641	0.1335	0.0584	0.0450
LightGCN	0.0411	0.0315	0.1830	0.1554	0.0649	0.0530
NIA-GCN	0.0369	0.0287	0.1359	0.1106	0.0599	0.0491
LR-GCCF	0.0335	0.0265	0.1519	0.1285	0.0561	0.0343
DGCF	0.0422	0.0324	0.1842	0.1561	0.0654	0.0534
GRMF	0.0354	0.0270	0.1477	0.1205	0.0571	0.0462
BPR-MF	0.0338	0.0261	0.1616	0.1366	0.0549	0.0445
ENMF	0.0359	0.0281	0.1523	0.1315	0.0624	0.0515
DeepWalk	0.0346	0.0264	0.1034	0.0740	0.0476	0.0378
Node2Vec	0.0402	0.0309	0.1019	0.0709	0.0452	0.0360
SSGCF	0.0493	0.0387	0.1849	0.1589	0.0666	0.0551
NASGCF	0.0498	0.0396	0.1912	0.1642	0.0683	0.0562

4.4 本章小结

本章第一节讨论了以往的研究存在的局限性，提出算法的研究动机进行了介绍；第二节是对提出算法模型的具体设计与实现，详细介绍了将自监督图学习与邻居感知注意力图神经网络协同过滤模型相结合的过程；第三节对实验进行了详尽的设计，简要地介绍了对比算法以及参数设置；第四节进行了消融实验以及对比实验，从实验结果和方法原理上说明了提出算法的合理性与优越性。

5 结论与展望

5.1 论文结论

随着互联网的快速发展和信息爆炸式增长，人们面临着纷繁复杂的信息。在这种情况下，在这样的背景下，如何高效地筛选信息，为用户提供个性化的推荐服务，成为关注的焦点。为了更好地给用户的信息筛选和喜好推荐的服务，推荐系统发挥着关键作用，其重要性日益凸显，得到了广泛的关注和应用。当前，传统推荐算法已经难以满足人们日益发展的信息需求。随着机器学习、数据挖掘和人工智能等技术的不断发展和成熟，深度学习模型以其强大的表征学习能力和对复杂数据的处理能力，为推荐系统带来了新的思路和方法。图神经网络作为深度学习在图数据结构上的一种应用，可以有效地捕捉节点之间的复杂关系和拓扑结构，并学习到更高层次的表示。在当前图神经网络推荐算法的研究中，仍然存在一些不足和局限性。将自监督图学习框架应用于图神经网络，能够在一定程度上提高模型泛化能力，适应不同的任务和领域；增强模型的鲁棒性，对于数据中的噪声和异常情况有更好的处理能力，保持稳定和可靠的性能；降低人工标注成本，使得图神经网络在更广泛的应用场景中得到应用和推广。本文根据对当前推荐系统所遇挑战的深入剖析，结合当前广泛应用的图神经网络推荐算法展开研究。对于这些挑战，提出了针对性的算法，以期为解决推荐系统当前面临的问题提供新的思路和方法。总结如下：

(1) 基于自监督学习的简化图神经网络协同过滤算法。本算法的创新点在于，将自监督图学习模型框架与简化图神经网络协同过滤模型相结合，能够降低噪声数据对于模型训练的干扰；去除图卷积网络层之间的非线性部分，降低模型的复杂程度，使得模型易于训练，可扩展至大型数据集；使用图划分技术减少图数据结构的输入规模，缓解了图卷积聚合操作中的过度平滑问题。在实验部分，对模型进行调优寻找模型训练最佳参数值，进行噪声实验证明了自监督图学习对于模型缓解噪声数据负面影响的有效性，对抗噪声交互的鲁棒性明显提升。经过在公开数据集上进行的多次对比实验，结果表明提出的算法在性能上相较于其他基线算法展现出了明显的优势。

(2) 基于自监督学习的邻居感知注意力图神经网络协同过滤算法。将自监督图学习模型框架与邻居感知注意力图神经网络协同过滤相结合，能够有效提高模型的鲁棒性和精确度；物以类聚，人与群分。通过建模邻域内的关系信息，强调邻域之间的隐式相关性，从而更有效地缓解了算法冷启动和数据集稀疏两大关键问题，提升了算法的适用性和准确性；删除了模型中对协同过滤有负面影响的特征变换和非线性激活，使得模型训练更加高效。

在实验部分，对提出的两个算法进行对比实验，证明了邻域感知注意力机制对于模型训练效率提升的有效性。对模型进行调优训练得到最佳的参数值，采用 Max-M 采样策略能够有效地训练模型。进行消融实验，验证了自监督图学习和邻域感知注意力机制的有效性。在真实数据集上进行了大量的对比实验，实验数据展示出提出的算法比当前研究中先进的协同过滤推荐算法和图神经网络推荐算法，均展现出了显著的推荐性能提升，进一步说明了自监督图学习和邻域感知注意力机制的合理性与优越性。

5.2 未来展望

通过深入研究图神经网络协同过滤推荐算法，提出创新算法，这些算法在一定程度上提高了模型的性能，有效解决了传统推荐系统的局限性问题。为了不断完善和提升提出的推荐算法性能，制定以下研究思路：

(1) 在当前工作实验部分的参数设置方面，按照自监督图学习的理论经验设置温度系数和概率系数，可能存在适应性差和解释性差的问题。未来的工作中，计划进一步优化模型的参数设置，以进一步提升算法的性能和效果，具体来说：一是将对模型中的关键参数进行系统性的调优和优化。通过使用自动调参技术或者网格搜索等方法，寻找最佳的参数组合，以达到最优的模型性能；二是考虑对模型的结构进行改进和优化。通过引入新的层或者改进现有层的设计，尝试提高模型的表达能力和学习能力，以更好地捕捉用户和物品之间的复杂交互关系；三是进一步探索和优化模型的评估指标。除了常规的召回率和归一化折损累计增益外，还将考虑引入其他评估指标，如多样性、覆盖率等，以更全面地评估模型的性能和效果。

(2) 在未来研究中，将专注于长尾问题的解决，通过引入更加有效的机制和方法，来改善推荐系统在长尾数据上的表现。具体而言，持续探索基于自监督学习的图神经网络方法，来捕捉长尾数据中的潜在关联和特征，并提高推荐系统的个性化能力。引入长尾目标函数来优化推荐系统的性能，并开展大量的实验研究。通过设计适用于长尾商品的损失函数或者目标函数，将直接优化长尾部分的推荐效果，从而提高长尾商品的推荐质量和用户满意度。在接下来的研究中，持续关注领域内的最新进展和研究动态，积极借鉴和应用前沿技术，为推荐系统的持续发展做出更多的贡献。

参考文献

- [1] Jacoby J, Speller D E, Berning C K. Brand choice behavior as a function of information load: Replication and extension[J]. Journal of consumer research, 1974, 1(1): 33-42.
- [2] Bawden D, Holtham C, Courtney N. Perspectives on information overload[C]//Aslib proceedings. MCB UP Ltd, 1999, 51(8): 249-255.
- [3] Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry[J]. Communications of the ACM, 1992, 35(12): 61-70.
- [4] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th international conference on World Wide Web. 2001: 285-295.
- [5] Koren Y, Rendle S, Bell R. Advances in collaborative filtering[J]. Recommender systems handbook, 2022: 91-142.
- [6] Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006, 18(7): 1527-1554.
- [7] Wang M, Deng W. Deep face recognition: A survey[J]. Neurocomputing, 2021, 429: 215-244.
- [8] Khurana L, Chauhan A, Naved M, et al. Speech recognition with deep learning[C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 1854(1): 012047.
- [9] Otter D W, Medina J R, Kalita J K. A survey of the usages of deep learning for natural language processing[J]. IEEE transactions on neural networks and learning systems, 2020, 32(2): 604-624.
- [10] Grigorescu S, Trasnea B, Cocias T, et al. A survey of deep learning techniques for autonomous driving[J]. Journal of Field Robotics, 2020, 37(3): 362-386.
- [11] Zhang S, Yao L, Sun A, et al. Deep learning based recommender system: A survey and new perspectives[J]. ACM computing surveys (CSUR), 2019, 52(1): 1-38.
- [12] Wu Y, Lian D, Xu Y, et al. Graph convolutional networks with markov random field reasoning for social spammer detection[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(01): 1054-1061.
- [13] Bruna J, Li X. Community detection with graph neural networks[J]. stat, 2017, 1050: 27.
- [14] Shchur O, Günnemann S. Overlapping community detection with graph neural networks[J]. arXiv preprint arXiv:1909.12201, 2019.
- [15] Sanchez-Gonzalez A, Heess N, Springenberg J T, et al. Graph networks as learnable physics engines

- for inference and control[C]//International Conference on Machine Learning. PMLR, 2018: 4470-4479.
- [16] Battaglia P, Pascanu R, Lai M, et al. Interaction networks for learning about objects, relations and physics[J]. *Advances in neural information processing systems*, 2016, 29.
- [17] Fout A, Byrd J, Shariat B, et al. Protein interface prediction using graph convolutional networks[J]. *Advances in neural information processing systems*, 2017, 30.
- [18] Park N, Kan A, Dong X L, et al. Estimating node importance in knowledge graphs using graph neural networks[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 596-606.
- [19] Khalil E, Dai H, Zhang Y, et al. Learning combinatorial optimization algorithms over graphs[J]. *Advances in neural information processing systems*, 2017, 30.
- [20] Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. *IEEE transactions on neural networks and learning systems*, 2020, 32(1): 4-24.
- [21] Ioannidis V N, Marques A G, Giannakis G B. A recurrent graph neural network for multi-relational data[C]//ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 8157-8161.
- [22] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. *arXiv preprint arXiv:1609.02907*, 2016.
- [23] Kipf T N, Welling M. Variational graph auto-encoders[J]. *arXiv preprint arXiv:1611.07308*, 2016.
- [24] Wu Z, Pan S, Long G, et al. Graph wavenet for deep spatial-temporal graph modeling[J]. *arXiv preprint arXiv:1906.00121*, 2019.
- [25] Li Y, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting[J]. *arXiv preprint arXiv:1707.01926*, 2017.
- [26] Jain A, Zamir A R, Savarese S, et al. Structural-rnn: Deep learning on spatio-temporal graphs[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 5308-5317.
- [27] Yan S, Xiong Y, Lin D. Spatial temporal graph convolutional networks for skeleton-based action recognition[C]//Thirty-second AAAI conference on artificial intelligence. 2018.
- [28] Herlocker J, Konstan J A, Riedl J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms[J]. *Information retrieval*, 2002, 5(4): 287-310. Hofmann T. Latent

- semantic models for collaborative filtering[J]. ACM Transactions on Information Systems (TOIS), 2004, 22(1): 89-115.
- [29] Bell R M, Koren Y. Improved Neighborhood-based Collaborative Filtering[J]. 2007.
- [30] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [31] Yildirim H, Krishnamoorthy M S. A random walk method for alleviating the sparsity problem in collaborative filtering[C]//Proceedings of the 2008 ACM conference on Recommender systems. 2008: 131-138.
- [32] Wang R, Jiang Y, Lou J. ADCF: Attentive representation learning and deep collaborative filtering model[J]. Knowledge-Based Systems, 2021, 227: 107194.
- [33] Sedhain S, Menon A K, Sanner S, et al. Autorec: Autoencoders meet collaborative filtering[C]//Proceedings of the 24th international conference on World Wide Web. 2015: 111-112.
- Gori M, Monfardini G, Scarselli F. A new model for learning in graph domains[C]//Proceedings. 2005 IEEE international joint conference on neural networks. 2005, 2(2005): 729-734.
- [34] 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.
- [35] He X, Du X, Wang X, et al. Outer product-based neural collaborative filtering[J]. arXiv preprint arXiv:1808.03912, 2018.
- [36] Gori M, Monfardini G, Scarselli F. A new model for learning in graph domains[C]//Proceedings. 2005 IEEE international joint conference on neural networks. 2005, 2(2005): 729-734.
- [37] 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.
- [38] Ying R, He R, Chen K, et al. Graph convolutional neural networks for web-scale recommender systems[C]//Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018: 974-983.
- [39] Zhang J, Shi X, Zhao S, et al. Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems[J]. arXiv preprint arXiv:1905.13129, 2019.
- [40] Xu C, Zhao P, Liu Y, et al. Graph Contextualized Self-Attention Network for Session-based Recommendation[C]//IJCAI. 2019, 19: 3940-3946.

- [41] Ma C, Ma L, Zhang Y, et al. Memory augmented graph neural networks for sequential recommendation[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(04): 5045-5052.
- [42] Wang X, He X, Wang M, et al. Neural graph collaborative filtering[C]//Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval. 2019: 165-174.
- [43] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[J]. Advances in neural information processing systems, 2017, 30.
- [44] He X, Deng K, Wang X, et al. Lightgcn: Simplifying and powering graph convolution network for recommendation[C]//Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 2020: 639-648.
- [45] Mao K, Zhu J, Xiao X, et al. UltraGCN: ultra simplification of graph convolutional networks for recommendation[C]//Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021: 1253-1262.
- [46] Wang X, Jin H, Zhang A, et al. Disentangled graph collaborative filtering[C]//Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. 2020: 1001-1010.
- [47] Wang X, Wang R, Shi C, et al. Multi-component graph convolutional collaborative filtering[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(04): 6267-6274.
- [48] Li Q, Han Z, Wu X M. Deeper insights into graph convolutional networks for semi-supervised learning[C]//Thirty-Second AAAI conference on artificial intelligence. 2018.
- [49] Xu K, Li C, Tian Y, et al. Representation learning on graphs with jumping knowledge networks[C]//International conference on machine learning. PMLR, 2018: 5453-5462.
- [50] Wang W, Feng F, He X, et al. Denoising implicit feedback for recommendation[C]//Proceedings of the 14th ACM international conference on web search and data mining. 2021: 373-381.
- [51] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets[C]//2008 Eighth IEEE international conference on data mining. Ieee, 2008: 263-272.
- [52] Ko H, Lee S, Park Y, et al. A survey of recommendation systems: recommendation models, techniques, and application fields[J]. Electronics, 2022, 11(1): 141.

- [53] Velickovic P, Cucurull G, Casanova A, et al. Graph attention networks[J]. *stat*, 2017, 1050(20): 10-48550.
- [54] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[J]. *Advances in neural information processing systems*, 2017, 30.
- [55] Berg R, Kipf T N, Welling M. Graph convolutional matrix completion[J]. *arXiv preprint arXiv:1706.02263*, 2017.
- [56] Xu K, Hu W, Leskovec J, et al. How powerful are graph neural networks?[J]. *arXiv preprint arXiv:1810.00826*, 2018.
- [57] Wang X, Ji H, Shi C, et al. Heterogeneous graph attention network[C]//*The world wide web conference*. 2019: 2022-2032.
- [58] Yun S, Jeong M, Kim R, et al. Graph transformer networks[J]. *Advances in neural information processing systems*, 2019, 32.
- [59] Dong Y, Chawla N V, Swami A. metapath2vec: Scalable representation learning for heterogeneous networks[C]//*Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017: 135-144.
- [60] 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.
- [61] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. *Advances in neural information processing systems*, 2017, 30.
- [62] He L, Chen H, Wang D, et al. Click-through rate prediction with multi-modal hypergraphs[C]//*Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2021: 690-699.
- [63] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[J]. *arXiv preprint arXiv:1205.2618*, 2012.
- [64] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. *arXiv preprint arXiv:1412.6980*, 2014.
- [65] Yin Z, Xu X, Fan K, et al. DGCF: A Distributed Greedy Clustering Framework for Large-scale Genomic Sequences[C]//*2019 IEEE International Conference on Bioinformatics and Biomedicine*

- (BIBM). IEEE, 2019: 2272-2279.
- [66] Rao N, Yu H F, Ravikumar P K, et al. Collaborative filtering with graph information: Consistency and scalable methods[J]. Advances in neural information processing systems, 2015, 28.
- [67] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations[C]//Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014: 701-710.
- [68] Grover A, Leskovec J. node2vec: Scalable feature learning for networks[C]//Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016: 855-864.
- [69] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks[C]//Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2010: 249-256.
- [70] Sun J, Zhang Y, Guo W, et al. Neighbor interaction aware graph convolution networks for recommendation[C]//Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. 2020: 1289-1298.
- [71] Chen L, Wu L, Hong R, et al. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(01): 27-34.
- [72] Ma X, Dong D. Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks[J]. IEEE transactions on knowledge and data engineering, 2017, 29(5): 1045-1058.
- [73] Liang D, Krishnan R G, Hoffman M D, et al. Variational autoencoders for collaborative filtering[C]//Proceedings of the 2018 world wide web conference. 2018: 689-698.

致谢

时光荏苒，岁月如梭，行文至此，思绪万千。求学之路二十载，转眼间我的研究生生涯即将画上圆满的句号。回首这段宝贵的时光，心中充满感激之情。在此，我要向所有给予我支持、鼓励和帮助的人致以最诚挚的谢意！

“幸遇太平，共襄盛举”，感谢伟大的党和祖国，让我生于华夏盛世，共享天下太平。个人的成长和发展离不开祖国的繁荣和安定，在今后的日子里，我将怀揣赤子之心，报效党和祖国！

“桃李不言，下自成蹊”，首先我要衷心感谢我的恩师：牛云云教授。您严谨的治学态度、深厚的学术造诣以及高尚的师德风范，都深深地影响着我。在您的悉心指导下，我不仅学会了如何进行科研工作，更学会了如何为人处世。您的严格教诲和悉心关怀，让我在研究生的道路上不断成长，不断进步，也为我今后的工作和生活奠定了坚实的基础。祝愿老师在今后的日子里，身体健康，万事如意！

“亲情似海，恩重如山”，我要感谢我的家人。在我求学的道路上，你们一直是我坚强的后盾。当我遇到困难和挫折时，家人总是给我无尽的温暖和力量，让我能够重新振作起来，继续前行。感谢我的父母，替我遮风挡雨让我衣食无忧从未吃过生活之苦；感谢我的父母，让我能够站在你们的肩膀上，拥有更加绚烂的世界，看到更加广阔的天地！

“愿岁并谢，与友长兮”，我还要感谢我的同学们和朋友们。在研究生期间，我们共同学习、共同进步，一起度过了许多难忘的时光。你们的陪伴和帮助，让我在求学的道路上不再孤单。我们之间最纯粹的情谊，将成为我人生中宝贵的精神财富。

“艰苦朴素，求真务实”，我要感谢学校提供的良好的学习环境和丰富的学术资源。四通八达的地理位置，尽职尽责的老师辅导员，无微不至的后勤部门，虚心倾听的校长信箱都让我难以忘怀，那八个字的校训也早已深深刻在心底。

“道阻且长，行则将至”，生活可能不像你想象的那么好，但是也不会像你想象的那么糟，人的软弱和坚强都超乎了自己的想象，有时候可能脆弱的一句话就会泪流满面，有时候你发现自己咬着牙，已经走过了很长的路。这一路上有顺境也有逆境，感谢求学路上坚持不懈、努力坚强的自己，一步一个脚印地不断前行。恰同学少年，风华正茂。希望未来的我不忘初心，砥砺前行！

“轻舟已过万重山”，在人的一生中最为辉煌的一天，并不是功成名就的那一天，而是从绝望和悲叹中产生对人生挑战的欲望，并且勇敢地迈向这种挑战的那一天。我的研究

生生涯是一段充满挑战与收获的旅程，这段宝贵经历让我有勇气面对以后人生中的各种挑战和挫折。在此，我要再次向所有给予我关心、支持和帮助的人表示衷心的感谢。我会将这段美好的经历珍藏在心底，继续前行，在今后的道路上更为坚强，更为正直，更为对未来充满希望！

附录

一、 个人简介

万双玮，男，中共党员，1998年3月9日出生，天津市滨海新区人，2020年6月取得河南大学计算机与信息工程学院网络工程学士学位。2021年9月至2024年6月于中国地质大学(北京)信息工程学院攻读计算机技术专业全日制工程硕士学位，专注于推荐算法方向的研究。在研究生阶段，致力于大数据挖掘和推荐算法的前沿领域研究工作，取得一定成果。目前，已经与华夏银行总行签订三方协议，即将入职。

二、 硕士期间主要研究成果

1. 在 SCI 一区 Information Fusion 期刊上发表过关于推荐算法的论文
2. 在 SCI 三区 Journal of Geochemical Exploration 期刊上发表过关于深度学习识别矿石的论文
3. 软件著作权：基于深度学习的匹配推荐系统 V1.0，登记号：2023SR1389524