



南京大學

NANJING UNIVERSITY

研究生畢業論文
(申請碩士學位)

論文題目 基于深度學習與歷史交互序列建模的

推薦排序算法研究

作者姓名

劉雅輝

學科、專業名稱

計算機科學與技術

研究方向

人工智能

指導教師

申富饒教授

2021年5月27日

学 号：MG1833050

论文答辩日期：2021年5月24日

指导教师：

(签字)

Research on Ranking Algorithms of Recommender Systems Based on Deep Learning and Historical Interaction Sequence Modeling

by

Yahui Liu

Supervised by

Professor Furao Shen

A dissertation submitted to
the graduate school of Nanjing University
in partial fulfilment of the requirements for the degree of

MASTER

in

Computer Science and Technology



Department of Computer Science and Technology
Nanjing University

May 27, 2021

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目： 基于深度学习与历史交互序列建模的
推荐排序算法研究

计算机科学与技术 专业 2018 级硕士生姓名： 刘雅辉

指导教师(姓名、职称)： 申富饶教授

摘 要

在互联网技术高速发展的时代背景下，人们在享受获取丰富资源的便捷性的同时，也面临着信息过载的问题。推荐系统的诞生为人们从海量数据中筛选出感兴趣的信息提供了新的可能性，因此，近年来推荐系统在电子商务、在线广告系统、社交媒体等领域的实际应用中扮演着越来越重要的角色。

在实践中，完整的推荐系统通常包括候选集生成阶段和排序阶段，本文主要关注排序阶段的算法研究工作。推荐排序算法的关键在于如何有效利用丰富的特征及其复杂的关系，实现精确预测用户对物品偏好的目的。为了实现推荐的个性化，近年来研究者纷纷基于注意力机制等技术设计更加精细化的推荐排序模型加强对历史交互信息的利用。然而，在实际推荐场景中广泛存在物品长尾问题。由于交互信息不充分，针对长尾物品的推荐比针对高曝光度物品的推荐更加困难。此外，新用户与新物品不断出现，如何在缺少历史交互信息的情况下进行推荐成为比长尾物品推荐问题更加困难的挑战，逐渐形成了面向冷启动问题的推荐这一研究方向。

本文以历史交互序列信息为着手点，研究如何更好地利用历史交互序列信息提升推荐排序模型的预测性能，以及在历史交互序列信息不足的情况下，如何设计算法充分利用仅有的信息快速生成高质量的替代特征。在此基础上，本文提出了两个基于深度学习的推荐算法针对性地解决上述问题，并将其应用在实际的图书推荐系统中。本文的主要研究内容与贡献如下：

首先，本文从历史交互信息的有效利用角度出发，将用户与物品的交互建模为一个动态二部图，在此基础上提出一个多向交互图注意力网络推荐模型，简称 MIGAT。MIGAT 模型基于注意力机制分别考察历史交互序列中的每一项交

互对其交互对象、推荐对象以及推荐上下文的重要度，同时考虑到交互发生时间的早晚对交互置信度的影响，设计置信度嵌入向量结构以区分不同交互的置信度。综合多个维度下的重要度衡量指标，MIGAT 模型基于图神经网络与前馈神经网络结构生成表达能力更丰富的用户特征表示、物品特征表示以及自适应的交互序列特征表示。经实验验证，该模型在多个数据集上表现优于当前先进的推荐排序模型，在长尾问题比较严重的情况下性能提升更加明显。

其次，本文针对新用户或新物品缺少历史交互信息的问题，提出一个基于元学习的伪交互序列生成器模型，简称 MePIGen。该模型基于元学习算法对参数进行学习，使得模型经过少量几次对新用户与新物品的探索之后就能生成高质量的伪交互序列特征。伪交互序列特征将作为真实交互序列的替代特征用于推荐排序模型中，以提升推荐排序模型的预测性能。MePIGen 的结构基于 Transformer 而设计，将现有的历史交互序列与属性特征作为输入，得到输出的伪交互序列，保证生成伪交互序列的质量与效率。经实验验证，该模型在多个数据集上使得当前先进的推荐排序模型的冷启动推荐性能得到提升。

最后，本文所提出的两个推荐模型应用于实际搭建的图书推荐系统中，获得良好的推荐性能。该图书推荐系统中，从针对新用户的冷启动阶段，到已经积累一定交互数据样本进入常规阶段，构成一个完整的推荐体系，充分展示了模型的实际应用价值。

关键词：推荐系统，历史交互，冷启动，深度学习

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Research on Ranking Algorithms of Recommender Systems
Based on Deep Learning and Historical Interaction Sequence Modeling
SPECIALIZATION: Computer Science and Technology
POSTGRADUATE: Yahui Liu
MENTOR: Professor Furao Shen

ABSTRACT

In the context of the rapid development of Internet technology, people are facing the dilemma of information overload while enjoying the convenience of acquiring abundant resources. The emergence of recommender systems provides new possibilities for people to select information of interest from massive amounts of data. Therefore, in recent years, recommender systems have played an increasingly important role in practical applications in e-commerce, online advertising systems, and social media.

In practice, a complete recommender system usually contains a candidate generation stage and a ranking stage. This article mainly focuses on the algorithm research work in the ranking stage. The key to the ranking models of recommender systems is how to effectively utilize rich features and their complex relationships to achieve the goal of accurately predicting the user preference for each item. To realize the personalized recommendation, in recent years, researchers design a lot of refined ranking models based on advanced technologies such as the attention mechanism to strengthen the use of historical interaction information. However, the long-tail problem of items widely exists in actual recommendation scenarios. Due to the insufficiency of interaction information, the recommendation for long-tail items is more difficult than recommendations for highly exposed items. Besides, new users and new items keep appearing in the system. How to make recommendations in the absence of historical interaction information has become a more difficult challenge than the long-tail item recommendation problem, and gradually formed the research direction of cold-start recommendation.

This article takes historical interaction sequence information as a starting point to study how to make better use of historical interaction sequence information to improve the prediction performance of the ranking models of recommender systems, and how to design algorithms to make full use of only information to quickly generate high-quality alternative features when the historical interactive sequence information is insufficient. On this basis, we propose two recommendation algorithms based on deep learning to specifically solve the above problems, and apply them in the actual book recommendation system. The main research contents and contributions of this article are as follows:

Firstly, from the perspective of effective utilization of historical interactive information, we first model the interaction between users and items as a dynamic bipartite graph, and then propose a recommendation model called multi-directional interactive graph attention network (MIGAT). The MIGAT model respectively considers the importance of each interaction in the historical interaction sequence to its interactive objects, recommended objects, and recommended context through the attention mechanism. At the same time, the impact of the occurring time of interactions on the interaction confidence is taken into the consideration. The confidence embedding vectors are used to distinguish the confidence of different interactions. Integrating multiple importance measurements, the MIGAT model generates more expressive user feature representations, item feature representations, and adaptive interaction sequence feature representations based on graph neural network and feed-forward neural network structures. Experimental shows that this model performs better than the state-of-the-art recommendation ranking models on multiple datasets and the performance improvement is more obvious when the long-tail problem is more serious.

Secondly, to handle the problem that new users or new items lack historical interaction, we propose a meta-learned pseudo interaction generator model (MePIGen). The MePIGen model learns the parameters based on the meta-learning algorithm so that the model can generate high-quality pseudo interaction sequence features after a few explorations of new users and new items. The pseudo interaction sequence features will be used as alternative features of the real interaction sequence in ranking models, to im-

prove the prediction performance of the ranking models. The structure of the MePIGen is designed based on the Transformer. It takes the existing historical interaction sequence and profile features as input and obtains the output pseudo interaction sequence. Such structure ensures the quality and efficiency of the generated pseudo interaction sequence. Experimental shows that this model improves the cold-start recommendation performance of the state-of-the-art ranking models on multiple datasets.

Finally, we apply the two recommendation models proposed in this article to the actual book recommender system we built. Results show that recommendation performance is satisfied. In our book recommendation system, a complete recommendation environment is constituted from the user cold-start phase to the regular phase that the new user has accumulated a certain amount of interaction data, which demonstrates the practical application value of the proposed models.

KEYWORDS: Recommender Systems, Historical Interaction, Cold-Start, Deep Learning

目 录

中文摘要	i
英文摘要	iii
目 录	vii
插图清单	xi
附表清单	xiii
1 绪论	1
1.1 研究背景与意义	1
1.2 研究现状与挑战	3
1.2.1 推荐排序算法	3
1.2.2 面向冷启动问题的推荐算法	4
1.3 研究内容与贡献	5
1.4 论文纲要	6
2 预备知识与相关工作	9
2.1 推荐排序问题	9
2.2 基于传统机器学习的推荐排序算法	10
2.2.1 基于广义线性模型的推荐算法	10
2.2.2 基于 FM 模型的推荐算法	11
2.3 基于深度学习的推荐排序算法	12
2.3.1 Embedding&MLP 范式	13
2.3.2 基于Embedding&MLP 范式的经典推荐模型	14
2.3.3 基于历史交互信息的推荐算法	15
2.4 面向冷启动问题的推荐算法	16
2.4.1 元学习	17
2.4.2 基于元学习的冷启动推荐算法	18
2.5 本章小结	20
3 基于历史交互信息的推荐排序模型	21
3.1 研究动机	21
3.2 关键技术	22

3.2.1 注意力机制	22
3.2.2 图神经网络	23
3.3 多向交互图注意力网络	24
3.3.1 特征表示	25
3.3.2 模型架构	26
3.3.3 学习算法	30
3.4 实验与分析	30
3.4.1 实验设置	30
3.4.2 对比实验	33
3.4.3 消融实验	35
3.4.4 综合分析讨论	38
3.5 本章小结	38
4 基于模拟交互序列生成的冷启动推荐模型	39
4.1 研究动机	39
4.2 关键技术	41
4.2.1 序列到序列模型	41
4.2.2 贪心搜索解码算法	43
4.3 基于元学习的伪交互序列生成模型	43
4.3.1 特征表示	44
4.3.2 模型架构	45
4.3.3 元学习算法	47
4.4 实验与分析	49
4.4.1 实验设置	49
4.4.2 对比实验	52
4.4.3 消融实验	53
4.4.4 综合讨论与分析	56
4.5 本章小结	57
5 应用：图书推荐系统	59
5.1 相关背景	59
5.2 系统设计	60
5.2.1 系统需求	60
5.2.2 系统架构	60
5.3 系统实现	61
5.3.1 数据采集与存取	61
5.3.2 数据预处理与特征提取	62
5.3.3 图书推荐	62

5.4 效果展示·····	63
5.4.1 智慧教育平台·····	63
5.4.2 图书推荐系统·····	64
5.5 本章小结·····	65
6 总结与展望·····	67
参考文献·····	69
A 符号与函数说明·····	79
A.1 基础符号说明·····	79
A.2 基础函数说明·····	79
致谢·····	81
简历与科研成果·····	83
学位论文出版授权书·····	85

插图清单

2-1	推荐排序算法的基本架构	9
2-2	Embedding&MLP 范式的基本架构	13
2-3	MAML 算法示意图 ^[1]	18
3-1	模型架构。图的左半部分展示了我们提出的模型 MIGAT 的网络结构，包括一个用来获取类别特征的嵌入表示的嵌入层，一组用来区分不同交互的置信度嵌入向量，一个用来生成表达能力丰富的特征表示的交互嵌入生成器，以及最终进行预测推荐值的 MLP 层。图的右半部分展示了 MIGAT 中交互嵌入生成器的结构，它基于注意力机制计算得到多个维度下的重要度信息，综合考虑多种重要度信息生成表达能力丰富的特征表示。	26
3-2	不同长尾阈值下的性能对比（以 AUC 度量）	35
3-3	置信度嵌入向量在长尾设定下的效果（以 AUC 度量）	36
3-4	长尾设定下使用不同注意力函数的模型性能（以 AUC 度量）	37
4-1	编码器-解码器框架示意图	41
4-2	Transformer 模型结构 ^[2]	42
4-3	基于 MePIGen 的冷启动推荐模型架构	45
4-4	MePIGen 模型结构	46
4-5	Movielens 数据集上不同层数的 Transformer 对冷启动推荐性能的影响（以 AUC 度量）	55
4-6	Movielens 数据集上支持集大小对冷启动推荐性能的影响（以 AUC 度量）	56
5-1	图书推荐系统架构	61
5-2	智慧教育平台基础页面	63
5-3	图书推荐效果	64

附表清单

3-1	数据集统计信息	31
3-2	四个数据集上整体对比实验的 AUC 结果	33
3-3	动态交互特征的效果 (以 AUC 度量)	36
3-4	置信度嵌入向量的效果 (以 AUC 度量)	36
3-5	使用不同注意力函数的模型性能 (以 AUC 度量)	37
4-1	数据集统计信息	51
4-2	数据集划分情况	51
4-3	三个数据集上整体对比实验的 AUC 结果	53
4-4	Movielens 数据集上使用不同学习算法对模型性能的影响	54

第一章 绪论

1.1 研究背景与意义

随着互联网的高速发展,越来越多的信息涌入到人们的生活中。然而,爆炸性增长的信息量远远超过了人们的接受能力,从海量信息中找到自己需要的信息成为一件异常困难的事情,大量冗余信息给人们带来沉重的负担。因此,如何帮助人们进行信息筛选和与过滤,将人们从信息过载的环境中解放,获取到最需要的信息,成为需要迫切解决的问题。在这样的背景下,为用户筛选出其可能感兴趣的物品或内容的推荐系统 (recommender systems) 出现在人们的视野中,并在电子商务、在线广告系统、社交媒体等领域的实际应用中扮演着越来越重要的角色。

推荐系统的主要工作是根据用户及物品的特征或历史行为记录,学习出用户的兴趣爱好,预测用户对给定物品的偏好程度,进而为用户推荐其可能感兴趣的物品。在工业界,推荐系统通常由候选集生成阶段和排序阶段组成^[3]。在候选集生成阶段,以协同过滤 (collaborative filtering, 简称 CF) 模型^[4-10] 为代表的推荐召回模型将从巨大的物品语料库中筛选出一个较小规模的物品集合作为推荐候选集。由于这一阶段涉及到的数据量十分可观,因此推荐召回模型通常都具有相对简单但运行高效的特点。在排序阶段,更加复杂而精细的推荐排序模型基于丰富的稀疏特征对候选集中的每个候选物品计算偏好评分,然后按照该评分进行排序,将评分最高的物品列表展示给用户作为最终的推荐结果。

随着深度学习 (deep learning) 在图像处理^[11]、语音识别^[12]、自然语言处理^[13] 等领域取得巨大成功,基于深度学习的推荐模型走入人们的视野^[14-15]。深度学习模型具有自动捕获高阶和非线性等复杂特征关系的能力,降低了模型对人工构造特征的依赖性,大大减轻了特征工程的工作量,并在大规模数据场景下普遍取得更好的效果。因此,基于深度学习的推荐模型逐渐成为推荐系统领域的主流研究方向,并广泛应用于谷歌、阿里巴巴、字节跳动、美团等公司的实际业务场景中。

推荐系统的关键在于个性化，即针对每个用户的推荐都应当是专属于该用户的。我们考虑在线购物的场景，尽管许多用户拥有相同的年龄、性别、地区，但每个人的喜好都各不相同，最终选择购买的商品也会各不相同；尽管同类商品中有许多相似的替代物，但每个商品总是会吸引不同的顾客来购买。因此，为了达到个性化推荐的目的，仅根据用户和物品的属性是不足以准确把握用户的偏好和物品所吸引的人群的，更加个性化的信息蕴涵在用户和物品的历史交互信息中。历史交互信息是指用户或物品在过去一段时间所发生的交互，其中交互往往体现为浏览、点击、点赞、评论或评分等行为。对历史交互信息的利用一直以来都是推荐召回阶段的研究热点。随着注意力机制 (attention mechanism)^[16]等技术的发展，通过更加精细化的模型加强对历史交互信息的利用也成为了近年来推荐排序阶段的研究热点^[17-21]。

然而，在实际推荐场景中，通常只有少部分物品能够得到充分的曝光，拥有充分的交互记录，剩下的大部分物品则只有少量交互记录^[22-23]，这种现象被称为物品分布的长尾 (long-tail) 现象，这些未得到充分曝光的物品被称为长尾物品。Yin 等人^[23]首先提出了长尾物品推荐问题，也即物品分布呈长尾现象的条件下的推荐问题。由于交互信息不充分，针对长尾物品的推荐将会比针对高曝光度物品的推荐更加困难。同时，长尾物品拥有着巨大的体量，长尾物品的推荐性能将会对整个系统的推荐性能产生极大的影响。因此，在设计推荐模型时充分考虑物品的长尾现象，提升长尾物品推荐性能，对充分发挥推荐系统的作用具有重要的意义。

此外，新用户与新物品不断出现，如何在缺少历史交互信息的情况下进行推荐成为比长尾物品推荐问题更加困难的挑战，逐渐形成了面向冷启动问题的推荐这一研究方向。由于缺少历史交互信息，基于历史交互信息的模型结构将部分失效，出现预测效果大打折扣甚至无法预测的问题。另一方面，许多模型都会根据用户或物品的标识符 (identifier, 简称 ID) 构建对应的嵌入向量，而新用户和新物品对应的 ID 特征表示向量没有足够的样本用来训练，与随机生成的向量效果相当，同样降低了模型的性能。因此，冷启动场景下的推荐任务更加困难。由于在对新用户或新物品一无所知的情况下进行个性化推荐是不具备可行性的，基于该问题的研究通常都建立在能够获得一定辅助信息或允许通过一段时间的探索调整的条件下进行。其中，冷启动的探索过程应当是尽可能短的，因为对

于新用户，如果初始一段时间推荐的物品都不是自己感兴趣的，用户则很可能从系统中流失；对于新物品，如果初始一段时间都被推荐到错误的人群中，物品的价值将难以被发掘进而在系统中沉寂。因此，针对推荐系统的冷启动问题设计模型，使得针对新用户与新物品的推荐能够快速达到较好的效果，对用户、平台而言都有着重要意义。

本文主要关注推荐排序阶段的深度学习算法研究，以及冷启动场景下如何提升模型的性能。在当前数据量急剧上升、推荐系统与生活息息相关、深度学习技术高速发展的时代背景下，研究如何基于深度学习技术设计模型，以实现高效利用用户和物品的历史交互信息进行个性化推荐的目的，并在历史交互信息不充足的情况下提供良好的性能保障机制，具有重要的意义。

1.2 研究现状与挑战

1.2.1 推荐排序算法

推荐排序阶段的主要目的是准确预测用户偏好，通常用户偏好反映在用户的评分、点击、购买、点赞等行为中。模型在预测时，往往能够获取到丰富的稀疏类别特征，例如用户的 ID、性别、年龄段等用户属性信息，物品的 ID、类别、价格等物品属性信息，以及用户与物品的历史交互信息等。因此，如何设计模型使其有效地利用这些繁复的特征去精确预测用户偏好成为推荐排序领域的核心问题，在近年来得到学术界和工业界的广泛关注^[17,24-28]。

传统方法通常使用以逻辑回归 (logistic regression, 简称 LR) 为代表的广义线性模型^[29]，这类模型简单而高效，但无法捕获特征之间的关系，往往需要大量的人工特征工程来构造特征。为了自动捕获特征之间的复杂关系，嵌入向量被引入到推荐排序模型中以表示类别特征。通常，每个类别特征通过一个嵌入向量来表示，后续复杂的特征组合关系将通过学习嵌入向量的学习来获取。较早时期，模型通过固定的函数（如内积）来建模特征关系，例如分解机 (factorization machine, 简称 FM)^[24] 使用内积刻画每一对特征之间的关系。然而，特征关系是复杂多变的，仅依靠固定的低阶函数是不足以刻画的，在强调推荐精准性的推荐排序任务上仍有巨大的提升空间。

随着深度学习的发展, 基于深度学习的推荐排序模型受到越来越广泛地关注。基于深度学习的模型通常都满足 Embedding&MLP 范式^[17], 丰富的特征通过嵌入向量进行表示, 然后送到多层感知机 (multilayer perceptron, 简称 MLP) 网络中学习复杂的特征组合关系。得益于深度学习在复杂特征表示上的优越性, 基于深度学习的推荐排序模型能够捕获复杂的高阶非线性特征关系, 为推荐性能带来巨大提升。Deep Crossing^[28] 将所有嵌入向量拼接起来, 通过残差单元^[30] 学习特征关系。Wide&Deep^[25] 将人工构造特征与 MLP 网络学得特征组合关系结合起来进行最终的预测。YoutubeNet^[3] 将用户历史交互特征的表示向量求平均得到固定长度的交互特征表示, 然后与其他特征拼接起来通过 MLP 网络进行学习。此外, 还有大量 FM 模型与深度学习相结合的研究工作^[31-33,33-34], 这些模型能够同时自动捕获低阶与高阶特征关系。

在此基础上, 为了更好地实现个性化推荐, 近年来的模型更加关注对用户历史交互信息的利用, 基于注意力机制来提取表达能力更为强大的历史交互特征表示。DIN^[17] 使用注意力机制来获得针对候选物品的用户行为表示。ATRank^[18] 使用自注意力机制来建模异质的用户行为。DIEN^[19] 提出了一种基于注意力机制与 GRU^[35] 相结合的结构来建模用户兴趣的演化过程。DSIN^[20] 使用 Bi-LSTM 网络结构来抽取不同会话阶段的用户兴趣演变, 然后基于注意力机制学得不同阶段的用户兴趣对候选物品的影响。

然而, 这些模型通常都偏好于历史交互丰富的用户与物品, 因为丰富的历史交互意味着用户或物品的特征得到了足够样本支撑的训练, 更容易取得较强的表示能力, 帮助模型提升预测性能。对于推荐系统中普遍存在的长尾物品, 其历史交互不够充分, 这些模型涉及到长尾物品的预测往往难以取得令人满意的效果。因此, 如何设计模型使得整体性能与长尾物品推荐性能都有所提升, 仍是目前推荐排序模型面临的巨大挑战。

1.2.2 面向冷启动问题的推荐算法

针对推荐系统中不断出现的新用户与新物品, 在缺少历史交互数据的情况下进行推荐, 可以从数据和模型两个角度着手进行研究。

传统方法通常从数据角度入手, 通过引入辅助信息的方式进行推荐。早期的方法通过引入丰富的用户属性和物品属性信息^[36-38] 或社交平台信息^[39-40] 进

行推荐。2017 年异质信息关系网络被提出^[41]，在用户与物品交互的基础上增加了更多实体之间的关系信息，基于该关系网络的冷启动算法随之诞生^[42-43]。这些方法基于引入的辅助信息预测新用户或新物品可能的偏好或受众，因而可以与其他数据实体之间建立联系，获得相对可信的推荐结果。这类方法会导致拥有相似属性的用户或物品的推荐结果总是相似的。但在实际中，属性相同的用户或物品可能依然会具有非常不同的行为模式。这种情况下，这类引入辅助信息进行推荐的模型的性能将严重受限。此外，基于协同过滤算法的研究中，通常采用数据增广技术解决冷启动问题。Hwang 等人提出采用 Zero-Injection 策略预测用户不感兴趣的物品，并把预测的信息加入到评分矩阵中^[44]。近年来，许多模型基于生成对抗网络 (generative adversarial network, 简称 GAN) 生成评分矩阵中缺失的数据^[45-49]。有研究表明^[50]，尽管这些模型大大缓解整个数据集上的数据稀疏性问题，但针对新用户和新物品的推荐性能提升仍不明显。

从模型角度入手，则考虑允许对新用户和新物品进行一段时间探索性的推荐，并在这个过程中不断调整，最终达到令人满意的推荐效果。基于上下文赌博机的方法^[51-54]将推荐过程建模为基于特征的探索与利用过程，通过探索性地推荐不断更新自己的试错策略，达到提升推荐性能的目的。由于过长的探索过程会导致用户流失以及物品沉寂的问题，越来越多的算法研究开始关注探索过程的收敛速度，希望能够通过尽可能少的探索次数就达到较好的预测效果。随着元学习在小样本学习、强化学习等领域的成功应用，研究者发现元学习经过少量样本的训练即可快速适应新任务的优势非常适合解决推荐系统中的冷启动问题，因此基于元学习的推荐算法研究逐渐成为当前面向冷启动问题的算法研究热点^[43,55-57]。其中，大部分基于元学习的方法都采用模型无关元学习 (model-agnostic meta-learning, MAML)^[1] 框架设计算法，在探索过程中更新模型参数，保证参数可以尽快收敛，并且能够在后续的推荐过程中作为初始化参数，提供有保障的性能。

1.3 研究内容与贡献

本文围绕推荐排序算法进行研究，以历史交互序列信息为着手点，研究如何更好地利用历史交互序列信息提升推荐排序模型的预测性能，以及在历史交

互序列信息不足的情况下,如何设计算法充分利用仅有的信息快速生成高质量的替代特征。在此基础上,本文提出了两个推荐算法针对性地解决上述问题,并将其应用在实际的图书推荐系统中。本文的主要研究内容与贡献如下:

- 本文从历史交互信息的有效利用角度出发,将用户与物品的交互建模为一个动态二部图,在此基础上提出一个多向交互图注意力网络推荐模型,简称 MIGAT。MIGAT 模型基于注意力机制分别考察历史交互序列中的每一项交互对其交互对象、推荐对象以及推荐上下文的重要度,同时考虑到交互发生时间的早晚对交互置信度的影响,设计置信度嵌入向量结构以区分不同交互的置信度。综合多个维度下的重要度衡量指标, MIGAT 模型基于图神经网络与前馈神经网络结构生成表达能力更丰富的用户特征表示、物品特征表示以及自适应的交互序列特征表示。经实验验证,该模型在多个数据集上表现优于当前先进的推荐排序模型,在长尾问题比较严重的情况下性能提升更加明显。
- 本文针对新用户或新物品缺少历史交互信息的问题,提出一个基于元学习的伪交互序列生成器模型,简称 MePIGen。该模型基于元学习算法对参数进行学习,使得模型经过少量几次对新用户与新物品的探索之后就能生成高质量的伪交互序列特征,伪交互序列特征将作为真实交互序列的替代特征用于推荐排序模型中,以提升推荐排序模型的预测性能。MePIGen 的结构基于 Transformer 而设计,将现有的历史交互序列与属性特征作为输入,得到输出的伪交互序列,保证生成伪交互序列的质量与效率。经实验验证,该模型在多个数据集上使得当前先进的推荐排序模型的冷启动推荐性能得到提升。
- 本文所提出的两个推荐模型应用于实际搭建的图书推荐系统中,获得良好的推荐性能。该图书推荐系统中,从针对新用户的冷启动阶段,到已经积累一定交互数据样本进入常规阶段,构成一个完整的推荐体系,充分展示了模型的实际应用价值。

1.4 论文纲要

本文由六个章节构成,各章的主要内容概述如下:

第一章为绪论,主要介绍推荐系统及推荐排序算法的相关背景与研究进展,

以及本文针对上述问题所进行的研究内容及主要贡献。

第二章为相关工作，主要介绍推荐排序算法的基本概念与相关工作，重点是基于深度学习的推荐排序算法与面向冷启动问题的推荐算法。

第三章介绍本文提出的多向交互图注意力网络推荐模型 **MIGAT**，首先由当前推荐排序阶段存在的问题引出我们的研究动机，并对关键技术进行概述；然后详细阐述了 **MIGAT** 模型的设计细节，以及相对于现有模型在充分利用动态交互信息及解决长尾问题上的优越性；最后通过实验验证了所提出模型的有效性。

第四章介绍本文提出的基于元学习的伪交互序列生成模型 **MePIGen**，首先由当前冷启动推荐问题的研究中存在的问题引出我们的研究动机，并对关键技术进行概述；然后详细阐述了 **MePIGen** 模型的设计细节，以及训练和测试使用的元学习算法，在此基础上阐述我们提出的模型相对于现有模型在利用交互信息与常规阶段模型构建联系，以及提升冷启动阶段预测性能等方面的优越性；最后通过实验验证了所提出模型的有效性。

第五章主要介绍我们搭建的应用本文所提出推荐模型的图书推荐系统，首先介绍系统的相关背景，然后详细介绍系统的设计与实现，并展示系统最终的推荐效果。

第六章对全文进行总结回顾，并对未来基于历史交互的推荐排序模型研究方向进行展望。

此外，附录 A 中还列出了本文所出现的符号及相关定义。

第二章 预备知识与相关工作

本章对推荐排序算法研究领域的基础知识及相关概念进行总结，并介绍相关研究工作，具体包括推荐排序问题的基本概念、基于传统机器学习的算法研究工作、基于深度学习的算法研究工作、面向冷启动问题的推荐算法研究工作，以及这些研究中所用到的关键技术。通过对这些研究工作进行思考与分析，为本文所提出的推荐排序算法及冷启动探索机制打下良好的基础。

2.1 推荐排序问题

作为推荐系统的主要阶段之一，推荐排序阶段的目的在于基于丰富的属性特征、历史交互特征与上下文特征，精确预测用户对物品候选集中每个物品的偏好概率，然后将用户可能最感兴趣的物品推荐给用户。用于预测偏好概率的算法被称为推荐排序算法，是解决推荐排序问题的关键。图 2-1 描述了推荐排序算法的基本架构。

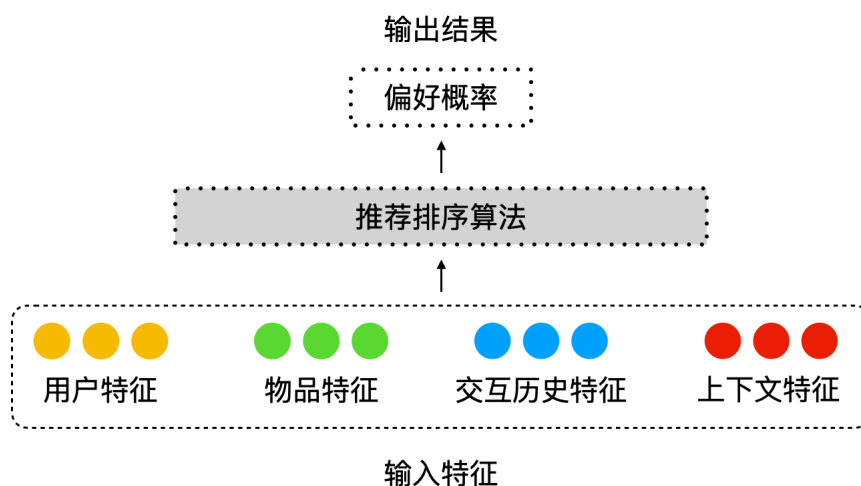


图 2-1: 推荐排序算法的基本架构

推荐排序算法所完成的用户偏好预测任务通常被描述为一个二分类有监督任务。对于训练样本 (\mathbf{x}, y) ， \mathbf{x} 表示由所有输入特征拼接而成的向量， y 表示用户的偏好指示标签，推荐排序算法完成的工作是给定输入特征 \mathbf{x} ，预测概率 $\Pr[y = 1|\mathbf{x}]$ 。通常，输入特征由大量的类别特征构成，例如用户属性中的标识

符 (identifier, 简称 ID)、性别、地区以及物品属性中的 ID、类别等。将这些离散型类别特征的编号直接以数值方式输入到机器学习模型中是没有意义的, 因此需要对这些特征进行编码处理。常见的编码方式是根据每个特征所含特征值的数目, 将特征通过 One-hot 编码或 Multi-hot 编码转化为一个高维稀疏二值特征向量进行表示, 示例如下:

$$\begin{array}{ccc} \underbrace{[1, 0, \dots, 0]}_{\text{用户 ID}=0} & \underbrace{[0, 1, 0, \dots, 0]}_{\text{物品 ID}=1} & \underbrace{[0, 0, 1, 0, \dots, 0]}_{\text{物品类别=喜剧}} \\ \\ \underbrace{[1, 0, 1, 0, \dots, 0]}_{\text{用户交互序列}=\{0,2\}} & \underbrace{[0, 1, 0, 1, 0, \dots, 0]}_{\text{物品交互序列}=\{1,3\}} & \end{array}$$

推荐排序算法的研究重点在于合理地设计机器学习模型, 充分利用丰富的输入特征进行预测, 使得预测结果能够有效反映用户的真实偏好, 进而可以基于预测结果进行排序得到与用户偏好相符的推荐结果列表。下文将对近年来推荐排序算法的研究工作进行介绍与总结。

2.2 基于传统机器学习的推荐排序算法

早期的推荐排序算法基于传统机器学习技术设计模型, 模型相对简单高效, 且具有较好的可解释性, 在工业界大规模数据场景下仍然占有一席之地。然而, 这类模型难以捕获特征之间复杂的组合关系, 往往需要依赖特征工程得到的更有效的特征作为模型的输入。其中, 最典型的两类方法是基于广义线性模型 (generalized linear models, 简称 GLMs) 的推荐算法与基于分解机 (factorization machine, 简称 FM) 的推荐算法, 下文将分别介绍这两类方法。

2.2.1 基于广义线性模型的推荐算法

基于广义线性模型的推荐算法直接将用户偏好预测任务当作普通的二分类任务, 然后利用广义线性模型进行学习。其中, 广义线性模型是指在线性模型的基础上, 施加以一个单调可微的联系函数对输出进行变换得到的模型, 即

$$y = f(\mathbf{x}) = g(\mathbf{w}^T \mathbf{x} + b), \quad (2-1)$$

其中 f 为广义线性模型, g 为联系函数, \mathbf{w} 和 b 为线性模型的参数。其中, 当 g 取式 (A-2) 所定义的 Sigmoid 函数时, 模型即为逻辑回归 (logistic regression, 简称 LR) 模型:

$$y = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}. \quad (2-2)$$

LR 是经典的机器学习分类模型, 通过对特征向量的每一维加权求和并由 Sigmoid 函数将结果归一化到 0 和 1 之间, 达到预测标签为 1 的概率的目的。基于 LR 的推荐模型^[29,58] 是早期工业界使用最广泛的推荐模型, 其简单高效、可解释性强的特点使其在计算资源相对有限的大规模推荐场景下迅速占据主导地位。

从式 (2-2) 可以看到, 基于 LR 的模型只能独立地考虑每一种特征的重要度, 而无法刻画特征之间的组合关系。但实际推荐任务中特征之间往往存在着多种多样的复杂关系, 此时就需要依赖人工构造的组合特征。采用人工构造特征的方式, 一方面造成巨大的特征工程开销; 另一方面仅靠专家知识难以筛选出所有有意义的组合特征, 在一定程度上限制了基于 LR 的推荐排序模型的预测性能。

2.2.2 基于 FM 模型的推荐算法

FM^[24] 在线性模型的基础上, 将每一对特征组合的重要度纳入了考虑范围。对于输入特征 $\mathbf{x} = (x_1, x_2, \dots, x_n)$, FM 模型的形式化表达如下:

$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j, \quad (2-3)$$

其中, w_0 为全局偏置参数, w_i 表示第 i 维特征的权重参数, \mathbf{v}_i 表示第 i 维特征对应的隐向量, $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ 表示第 i 维特征与第 j 维特征的组合权重。注意到, 式 (2-3) 中没有直接通过参数表示特征组合的权重, 而是用隐向量的内积运算得到权重。这样一方面可以降低组合特征的数量; 另一方面由于所有包含 x_i 的特征组合的权重均可以随着 \mathbf{v}_i 的更新而更新, 每对特征组合的权重得到训练的次数将大大增加, 即便是对于新出现的特征组合, 其权重也极有可能已经得到了一定量的训练, 从而缓解了数据稀疏性问题, 增强了模型的泛化性能。

从式 (2-3) 可以看到, FM 模型具备刻画成对特征之间的关系, 大大降低了

广义线性模型对特征工程的依赖，同时 FM 在一定程度上解决了推荐系统普遍存在的数据稀疏性问题。此外，通过对式 (2-3) 进行化简，FM 的时间开销可以降低到 $O(n)$ 。因此直到今天，FM 模型仍广泛应用于工业界。然而，FM 中对特征关系的利用方式仍比较简单，实际任务中仍存在许多复杂的高阶非线性特征关系。尽管 FM 可以扩展到任意 d 阶形式

$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{l=2}^d \sum_{i_1=1}^n \cdots \sum_{i_{l-1}=i_{l-2}+1}^n \left(\prod_{j=1}^l x_{i_j} \right) \left(\sum_{f=1}^{k_l} \prod_{j=1}^l v_{i_j, f}^{(l)} \right), \quad (2-4)$$

其中第 l 阶交互通过 PARAFAC 模型^[59] 进行因子化，但高阶形式的 FM 模型所带来的高时间复杂度使得其难以在实践中得到应用。

域感知分解机 (field-aware factorization machine, FFM)^[60] 在 FM 的基础上引入域的概念，通过将特征划分为不同的域，每个特征为自己不属于的其他域分别维护不同的隐向量，在预测时使用与自己进行组合的特征所在域的隐向量进行内积运算，从而更精细化地捕获到特征组合关系：

$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_{if_j}, \mathbf{v}_{jf_i} \rangle x_i x_j, \quad (2-5)$$

其中 f_i 表示特征 i 所在的域， \mathbf{v}_{if_j} 表示特征 i 针对域 f_j 的隐向量。然而，FFM 模型依然不具备刻画高阶非线性特征关系的能力。

2.3 基于深度学习的推荐排序算法

随着深度学习在图像处理^[11]、语音识别^[12]、自然语言处理^[13] 等领域取得突破性进展，基于深度学习的推荐排序算法逐渐得到研究人员的广泛关注并成为近年来的研究热点。深度学习强大的特征表示能力对于推荐算法捕获特征之间复杂的关系十分有益，大大减少了模型对特征工程的依赖，并且取得比传统机器学习算法更优秀的性能表现。基于深度学习的推荐排序算法通常都满足 Embedding&MLP 范式^[17]，下文将具体介绍此范式的基本架构与基于此范式的经典推荐算法，以及基于历史交互信息精细化设计模型结构的典型推荐算法。

2.3.1 Embedding&MLP 范式

Embedding&MLP 范式将编码后的稀疏特征映射到嵌入向量 (embedding vectors), 然后通过多层感知机 (multilayer perceptron, 简称 MLP) 结构学习复杂特征关系及特征表示, 并得到最终的预测结果。

如图 2-2 所示, Embedding&MLP 范式包括通用的嵌入层与多层感知机层, 以及可选的模型特异化的网络结构。

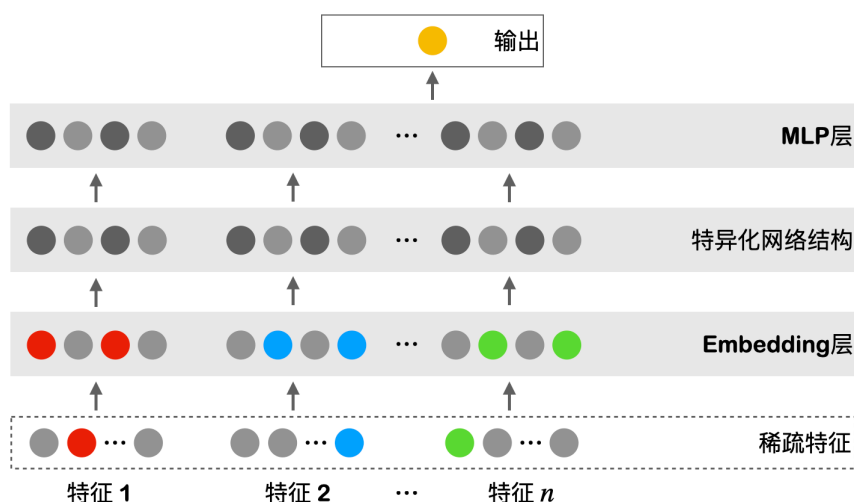


图 2-2: Embedding&MLP 范式的基本架构

在嵌入层, 高维稀疏向量通过查看嵌入表 (embedding table) 的形式转化为低维稠密向量。具体地, 给定嵌入表 $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N] \in \mathbb{R}^{H \times N}$ 及输入向量 $\mathbf{x} \in \{0, 1\}^N$, 则 \mathbf{x} 的嵌入向量表示为 $\mathbf{e} = [\mathbf{e}_{k_1}, \mathbf{e}_{k_2}, \dots, \mathbf{e}_{k_n}]$, 其中 $j \in \{k_1, k_2, \dots, k_n\}$ 当且仅当 $x_j = 1$ 。注意到, 对于采用 One-hot 编码的特征, 其嵌入向量表示为一个代表该特征所属类别的单独的嵌入向量; 对于采用 Multi-hot 编码的特征, 其嵌入向量表示为一系列代表该特征所包含类别的嵌入向量组合。

在多层感知机层, 由前序结构得到的特征表示将拼接起来, 然后通过若干全连接层 (fully connected layers) 构成的 MLP 结构学习特征之间的复杂关系, 并预测用户偏好概率。早期的深度学习模型通常不包含特异位网络结构, 主要通过设计 MLP 结构以提升模型对特征组合关系的学习能力。

近年来, 基于 Embedding&MLP 范式的深度学习模型的更加关注对特异位网络结构的设计。在嵌入层与多层感知机层的基础上, 研究者从不同的角度出发, 设计特殊的网络结构帮助模型更好地学得高质量的特征表示, 对于模型的

预测性能提升至关重要。

2.3.2 基于 Embedding&MLP 范式的经典推荐模型

近年来,研究者从不同角度出发,在 Embedding&MLP 基本架构的基础上设计深度学习模型的网络结构,帮助模型更好地学得高质量的特征表示。下面介绍几种经典模型的网络结构。

Deep Crossing^[28] 基于 He 等人在论文^[30]中提出的残差单元(residual unit)思想设计网络结构,将所有嵌入向量拼接起来,然后通过由两层全连接层代替卷积层的残差单元结构作为 MLP 层进行学习。Wide&Deep^[25] 模型结构分成 Wide 部分和 Deep 部分,其中 Wide 部分为 2.2.1 节所介绍的广义线性模型,Deep 部分为基础的嵌入层与 MLP 结构,两部分结果拼接起来通过最后的前馈神经网络进行融合得到最终的预测结果。YoutubeNet^[3] 在嵌入层之后首先对用户历史交互特征的表示向量进行平均池化(average pooling)操作得到固定长度的交互特征表示,然后与其他嵌入向量一同拼接起来通过 MLP 网络进行学习。PNN^[26] 引入了乘积层,通过嵌入向量的内积或外积刻画特征组合,并与原始嵌入向量拼接起来作为 MLP 层的输入。

此外,由于 2.2.2 节所介绍的 FM 模型使用的隐向量与嵌入向量的思想不谋而合,近年来诞生了一系列基于深度学习化的 FM 模型的研究工作。FNN^[61] 首先对 FM 模型进行深度学习化,将其中一阶项和二阶项的参数分别当作维度为 1 与 k 的嵌入向量,其中 k 为隐向量的维度大小,然后将所有的嵌入向量与全局偏置参数拼接起来送入 MLP 层进行学习。NFM^[31] 引入一个 Bi-Interaction 层,将 FM 中隐向量的内积操作替换为按位置相乘(element-wise product)操作,然后将得到的所有特征对的乘积相加得到最终的向量表示,再通过 MLP 层进行学习。AFM^[32] 在 NFM 的基础上,通过单层前馈神经网络学得每对特征组合的权重,基于该权重对特征对按位置乘积得到的向量进行加权求和。DeepFM^[62] 将 Wide&Deep 模型的 Wide 部分替换为 FM 模型,并且使得两部分共用嵌入向量以增强模型的泛化性能。xDeepFM^[33] 模型在 DeepFM 模型的基础上,将 FM 的内积操作替换为 CIN 网络结构,通过多层迭代地与嵌入向量组成的特征矩阵求哈达玛积(Hadamard product)并进行加权求和,实现对有限高阶特征关系的刻画。

2.3.3 基于历史交互信息的推荐算法

推荐系统中用户与物品的历史交互信息作为直观反映用户兴趣偏好与物品吸引人群的特征,对于提升个性化推荐的效果起到重要作用。早期的研究工作基于卷积神经网络 (convolutional neural network, 简称 CNN)^[63] 或循环神经网络 (recurrent neural network, 简称 RNN)^[64] 提取历史交互序列中隐含的信息^[27,65-67]。然而,仅基于 CNN 的模型更关注序列中相邻元素之间的关系而难以捕捉到整个序列的信息,基于 RNN 的模型则存在无法并行化所带来的时间开销大的问题。此外,由于 CNN 与 RNN 结构均只支持固定长度的特征向量表示,故难以同时保证长序列与短序列的特征提取能力。因此,仅基于 CNN 和 RNN 对历史交互建模的推荐模型未能在实践中得到广泛应用。

随着注意力机制在自然语言处理领域取得的巨大成功^[2,13,68],引入注意力机制对推荐系统中的历史交互信息进行建模逐渐形成主流研究趋势。ATRank^[18]考虑多种用户历史交互行为,将所有历史行为按行为类型分组后内部再按时间排序得到整体的历史交互行为特征表示,然后将该特征映射到多个语义空间,并分别在每个语义空间中使用自注意力 (self-attention) 机制来学习异质历史交互序列中隐含的用户兴趣表示。DIN^[17]使用注意力机制来获得每一项用户历史交互与当前候选物品的相关性,并以该相关性作为权重对历史交互加权求和,从而获得动态适应于候选物品的用户行为表示。DIEN^[19]通过 GRU^[35]建模用户的交互序列特征,然后使用带注意力更新门的 GRU 结构来增强与候选物品相关度更大的用户兴趣的影响,从而达到建模用户兴趣演化过程的目的。DSIN^[20]将用户历史交互按时间划分为不同的会话 (session),然后基于 Transformer^[2]结构对每个会话内的历史交互特征进行建模,接着使用 Bi-LSTM 结构刻画不同会话之间的演变关系,最后基于注意力机制学得不同阶段的用户兴趣对候选物品的影响。

基于注意力机制从历史交互序列中提取交互特征的模型,能够充分考虑序列中每一项交互的相关度或重要度,从而实现更加精细化的交互特征刻画。此外,注意力机制能够实现高度并行化的计算,即便在大规模推荐场景下也不会引入难以承受的时间开销。因此,基于注意力机制对历史交互序列建模的推荐模型在实践中得到越来越广泛的应用。

2.4 面向冷启动问题的推荐算法

工业界推荐系统中，往往会面临新用户与新物品频繁出现的境况，而新用户与新物品由于缺少历史交互信息并且自身特征无法得到充分训练，严重限制了推荐算法的性能发挥，这就是推荐系统面临的冷启动问题。面向冷启动问题的推荐算法研究工作通常从数据与模型两个维度进行考虑。从数据入手，通常采用引入辅助信息或使用数据增广技术的方式设计推荐算法。从模型入手，通常基于一定时间内的探索不断对模型更新调整以达到较好的推荐性能。

早期针对冷启动问题的研究工作几乎都采用向模型中引入辅助信息，并设计合理的模型结构使其充分利用辅助信息的方式。引入用户属性和物品属性的方法^[36-38,69]基于属性刻画用户或物品的特征，通过公共属性将新用户与新物品同拥有某种相同属性的老用户与老物品建立联系以实现冷启动推荐。其中 DropoutNet^[69]通过类似于 Dropout 操作^[70]的思想，在训练时对用户或物品的历史交互特征随机置 0，仅使用属性信息，实现对冷启动场景的模拟，从而达到了在协同过滤算法中引入辅助信息解决冷启动问题的目的。引入社交平台信息的方法^[39-40]，基于用户在 Twitter、微博等社交平台上关注和发表的信息，获得用户的特征表示及兴趣偏好信息。引入物品内容信息的方法^[71-72]基于物品本身的文本、视频或音频特征进行推荐而不考虑其历史交互。此外，还可以引入更加复杂的关系网络（例如知识图谱、异质信息关系网络^[41]等）进行推荐^[42-43,73]，通过特征属性实体之间丰富的关系信息辅助构建用户或物品的特征。上述方法通过辅助信息构建新用户或新物品与其他数据实体之间的联系，从而获得相对可信的推荐结果。然而，实际中往往会存在辅助信息不可用的情况，并且辅助信息相似的用户或物品可能会有完全不同的偏好表现。因此，基于辅助信息的推荐算法性能仍有很大的提升空间。

与基于辅助信息的推荐算法从数据角度出发解决冷启动问题不同，基于元学习 (meta-learning) 的推荐算法从模型的角度出发，通过在少量探索样本上的训练，快速适应于对新用户与新物品的预测任务，达到较好的预测效果。元学习的机制保证了探索过程的收敛速度较快，避免因探索过程过长而导致的用户流失以及物品沉寂的问题。基于元学习解决推荐系统中的冷启动问题已成为当前面向冷启动问题的推荐算法研究热点。下文将对元学习的基本概念进行简单介绍，

然后给出元学习框架下的冷启动问题定义，并介绍近年来基于元学习解决冷启动问题的典型工作。

2.4.1 元学习

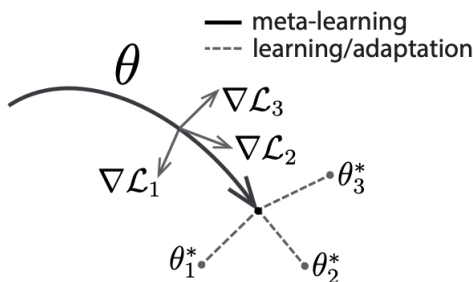
元学习的目的是通过在不同学习任务 (tasks) 上对模型进行训练，总结先验知识，然后基于先验知识在少量新任务的数据样本上训练以快速适应新任务^[74]。这个思想来源于人类的学习过程，希望模型能够像人一样学得一种根据通用规律解决新问题的学习方法。元学习方法通常被分成基于度量的方法、基于模型的方法与基于优化的方法。

与传统机器学习直接以样本为单位进行训练不同，元学习以任务作为学习单位，每个任务均包含一定量的数据样本，并且将样本划分为两个互斥的子集：支持集 (support set) 和查询集 (query set)，分别扮演着传统机器学习中训练集与测试集的角色。元学习模型由元学习器 (meta-learner) 与基学习器 (base-learner) 构成，元学习器负责为基学习器提供一个好的度量或参数，使得基学习器可以快速适应于新任务。在元学习的训练过程中，每一轮都在各个任务的支持集上训练并局部更新基学习器的参数，然后在查询集上使用局部更新的参数进行预测以检验元学习的效果，并基于此更新元学习器的参数。

注意到，元学习与模型的预训练过程有所区别。预训练只训练一个用于实际预测任务的模型，并且希望找到一组参数在大部分任务上都能表现好。而元学习涉及到基学习器与元学习器两个模型，并且希望找到元学习器的一组参数，帮助基学习器快速适应于未来的新任务。

接下来，我们以模型无关元学习 (model-agnostic meta-learning, 简称 MAML) 算法为例，介绍元学习的具体学习过程。MAML 属于基于优化的元学习算法，该方法通过元学习器学得基学习器参数的高质量初始化权重，从而实现快速适应新的学习任务的目的。形式化地，我们考虑模型 f ，其可以通过参数 θ 表示为 f_θ ，以及学习任务 \mathcal{T} 上的分布 $\rho(\mathcal{T})$ 。如图 2-3 所示，MAML 的目的是寻找一组对任务变化敏感的参数，使得对于任意 $\rho(\mathcal{T})$ 中的新任务 \mathcal{T}_i ，参数 θ 沿着 \mathcal{T}_i 的损失函数 \mathcal{L}_i 的梯度方向进行小幅度的更新得到 θ_i^* ，即可对损失函数 \mathcal{L}_i 产生大幅度的影响，也即适应于任务 \mathcal{T}_i 。

对于 MAML，其训练过程包括内层循环的局部更新，以及外层循环的全局

图 2-3: MAML 算法示意图^[1]

更新。内外层更新分别建立在查询集与支持集包含的样本上。具体地，模型首先对参数 θ 进行随机初始化，然后在每一轮都从分布 $\rho(\mathcal{T})$ 中采样一批学习任务 $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_N$ 进行训练。在每一轮训练中，模型对每个学习任务 \mathcal{T}_i 进行处理，基于支持集数据计算损失函数的梯度 $\nabla_{\theta} \mathcal{L}_i(f_{\theta})$ ，然后使用梯度下降 (gradient descent) 法对参数 θ 进行局部更新得到：

$$\theta_i^* = \theta - \alpha \nabla_{\theta} \mathcal{L}_i(f_{\theta}), \quad (2-6)$$

其中 α 为局部更新的学习率。在完成本轮所有任务上的局部更新后，模型基于所有任务的查询集数据，按照局部更新后的参数计算损失函数之和 $\sum_{i=1}^N \mathcal{L}_i(f_{\theta_i^*})$ ，然后使用梯度下降法对参数 θ 进行全局更新：

$$\theta = \theta - \beta \nabla_{\theta} \sum_{i=1}^N \mathcal{L}_i(f_{\theta_i^*}), \quad (2-7)$$

其中 β 为全局更新的学习率。

在测试阶段，模型基于测试任务的支持集样本进行局部微调 (fine-tune) 使模型快速适应于测试任务，然后在作为测试任务查询集数据的真实无标签样本上进行预测。

2.4.2 基于元学习的冷启动推荐算法

在推荐系统中，每一个用户或物品相关的全部样本可以被定义为一个学习任务。形式化地，我们将输入特征 \mathbf{x} 改写为 $\mathbf{x} = (\mathbf{x}_u, \mathbf{x}_i, \mathbf{x}_{I_u}, \mathbf{x}_{u_i}, \mathbf{x}_c)$ ，其中 $\mathbf{x}_u, \mathbf{x}_i, \mathbf{x}_{I_u}, \mathbf{x}_{u_i}, \mathbf{x}_c$ 分别表示用户 u 的属性特征，物品 i 的属性特征，用户 u 的历史交互

物品序列特征，物品 i 的历史交互用户序列特征，以及其他上下文相关特征。于是，用户偏好预测模型 f 可以表示为

$$f(\mathbf{x}) = f(\mathbf{x}_u, \mathbf{x}_i, \mathbf{x}_{I_u}, \mathbf{x}_{U_i}, \mathbf{x}_c) = g_u(\mathbf{x}_i, \mathbf{x}_{I_u}, \mathbf{x}_{U_i}, \mathbf{x}_c) = h_i(\mathbf{x}_u, \mathbf{x}_{I_u}, \mathbf{x}_{U_i}, \mathbf{x}_c), \quad (2-8)$$

其中 $g_u(\cdot, \cdot, \cdot, \cdot) = f(\mathbf{x}_u, \cdot, \cdot, \cdot, \cdot)$ 是用户 u 特有的函数， $h_i(\cdot, \cdot, \cdot, \cdot) = f(\cdot, \mathbf{x}_i, \cdot, \cdot, \cdot)$ 是物品 i 特有的函数。因此，我们可以定义推荐系统中的用户学习任务与物品学习任务为：

定义 2-1 (用户学习任务与物品学习任务) 对于给定的用户 u ，以及用户 u 相关的互斥数据集 \mathcal{S}_u 与 \mathcal{Q}_u ，我们定义用户学习任务 \mathcal{T}_u 为通过在支持集 \mathcal{S}_u 上训练模型 g_u ，使得对于任意查询集样本 $(\mathbf{x}, y) \in \mathcal{Q}_u$ ，模型的预测结果 $g_u(\mathbf{x})$ 为 $y = 1$ 的概率。类似地，对于给定的物品 i ，我们可以定义物品相关的支持集 \mathcal{S}_i ，查询集 \mathcal{Q}_i ，预测模型 h_i ，以及物品学习任务 \mathcal{T}_i 。

在推荐系统中产生过交互的用户和物品分别被称作老用户与老物品。我们定义老用户集合为 \mathcal{U}^{old} ，老物品集合为 \mathcal{I}^{old} ，则与其对应的老用户学习任务集合 $\mathcal{T}_U^{\text{old}}$ 与老物品学习任务集合 $\mathcal{T}_I^{\text{old}}$ 为

$$\begin{aligned} \mathcal{T}_U^{\text{old}} &= \{\mathcal{T}_u | u \in \mathcal{U}^{\text{old}}\}, \\ \mathcal{T}_I^{\text{old}} &= \{\mathcal{T}_i | i \in \mathcal{I}^{\text{old}}\}. \end{aligned} \quad (2-9)$$

在元学习框架下，我们基于老用户学习任务 $\mathcal{T}_U^{\text{old}}$ 训练元学习器，使得基学习器能够快速适应新用户 $u \notin \mathcal{U}^{\text{old}}$ 对应的学习任务；基于老物品学习任务 $\mathcal{T}_I^{\text{old}}$ 训练元学习器，使得基学习器能够快速适应新物品 $i \notin \mathcal{I}^{\text{old}}$ 对应的学习任务。注意到，在新用户学习任务或新物品学习任务的支持集上进行更新的过程即为冷启动的探索过程，通过在少量支持集样本上对基学习器进行微调，达到提升针对新用户或新物品的推荐性能的目的。元学习机制加快了探索过程的收敛，从而保证在相对少的探索次数下，模型能够达到比传统监督学习机制下更好的推荐性能。

近年来，研究者对冷启动问题进行深入思考，基于元学习机制提出许多面向冷启动问题的推荐算法。Vartak 等人在^[75]中提出两种基于度量的元学习模型来解决物品冷启动问题。此后的大部分方法都基于模型无关元学习算法进行设

计^[43,55-57]，通过为推荐模型学得一个好的初始化参数，使得模型能够快速更新参数实现对新用户与新物品的有效预测。其中^[56]采用引入辅助信息与元学习结合的方法设计模型，基于辅助信息学得物品 ID 的嵌入向量初始化参数，从而提升模型的性能。

2.5 本章小结

本章介绍了推荐排序问题的基本概念，以及近年来典型的推荐排序算法研究工作。经历从基于传统机器学习的推荐排序算法到基于深度学习的推荐排序算法的研究发展历程，当前主流推荐算法自动化捕获复杂特征关系的能力逐渐增强，用户偏好预测能力逐渐提升。在此之中，基于用户和物品的历史交互特征进行考虑的模型，因更加充分地利用到个性化偏好和吸引力信息，往往能够取得更优的性能。然而，推荐系统中不断出现的新用户与新物品，由于缺乏历史交互信息，严重限制了常规推荐排序算法的性能。面向冷启动问题的推荐算法研究从数据与模型两个角度进行思考，提升针对冷启动用户或冷启动物品的推荐性能，为整个推荐系统的性能提供了良好的保障。

第三章 基于历史交互信息的推荐排序模型

本章提出一个多向交互图注意力网络 (multi-directional interactive graph attention network, 简称 MIGAT) 推荐模型, 旨在充分利用用户和物品的历史交互信息, 为用户和物品之间建立更加紧密的直接联系, 从而提升推荐系统排序阶段的性能。

3.1 研究动机

推荐排序阶段的主要目的是有效预测推荐值作为推荐结果排序的依据, 其关键在于如何从大量特征中捕获用户兴趣及物品吸引力, 并判断二者的匹配程度, 进而预测出推荐值。传统方法通常使用以 LR 为代表的广义线性模型^[29], 这类模型简单而高效, 但无法捕获特征之间的关系, 往往需要大量的人工特征工程来构造特征。FM^[24] 将特征嵌入到隐向量中, 然后通过隐向量的内积操作来建模每一对特征之间的关系。FM 对于低阶特征关系的有效刻画使其拥有相当可观的效果, 但由于高阶 FM 具有高复杂度, 在实践中难以通过 FM 高效地刻画高阶特征关系。深度学习模型^[3,17,25,28,33] 将大规模的稀疏特征映射到嵌入向量, 然后通过特定的神经网络结构学习特征之间复杂的关系表示。由于其捕获高阶和非线性特征关系的优越性, 深度学习模型已成为当前主流的推荐排序模型。

在实际推荐系统中, 物品分布呈现明显的长尾 (long-tail) 现象。然而, 现有推荐排序模型在解决长尾物品推荐问题上有所欠缺。尽管近几年许多推荐排序模型对于充分利用用户历史交互特征构建用户兴趣表示进行了深入研究, 但这些模型很少关注到如何利用物品历史交互特征构建物品吸引力特征表示。缺少物品历史交互特征, 物品所能够吸引的用户具备的特征无法直接被捕获, 进而增加衡量一个物品对特定用户吸引力大小的难度。这严重限制了模型的预测性能, 尤其是在涉及到由于没有得到充分曝光进而特征表示不够有效的物品时。

此外, 由于用户兴趣和物品吸引力是动态变化的, 不同时间发生的交互在

表示用户兴趣或物品吸引力时,应当具有不同的置信度。然而,据我们所知,尚未有考虑历史交互的置信度的推荐排序模型。

3.2 关键技术

为了充分发掘用户和物品的历史交特征中隐含的个性化信息,我们在模型中引入注意力机制与图神经网络技术来增强特征的代表能力。本节将对注意力机制与图神经网络技术进行简单介绍。

3.2.1 注意力机制

近年来,注意力机制被广泛应用于以机器翻译为典型代表的序列建模任务中,并取得显著效果^[2,13,68]。其主要思想是考虑序列中的每一个部分的重要性,在此基础上求得整体序列的特征表示,从而达到使重要信息在整体的特征表示中发挥更大作用的目的。一般地,我们通过一个注意力函数来实现上述机制,该函数通常被描述为由一个查询 Q 及一组键值对 $(\mathbf{K}, \mathbf{V}) = (K_i, V_i)_{i=1}^l$ 到一个注意力值的映射:

$$\text{Attention}(Q, \mathbf{K}, \mathbf{V}) = \sum_{i=1}^l \text{Importance}(Q, K_i) \cdot V_i, \quad (3-1)$$

其中, Q, K_i, V_i 与注意力值 $\text{Attention}(Q, \mathbf{K}, \mathbf{V})$ 均为向量形式,且 Q 与 K_i 的维度相同, $\text{Importance}(Q, K_i)$ 通过特定函数 F 计算查询 Q 和键值的 K_i 之间的相似度,然后通过式 (A-4) 所定义的 Softmax 函数进行归一化得到对应的权重,即:

$$\text{Importance}(Q, K_i) = \text{Softmax}(F(Q, K_i)) = \frac{\exp(F(Q, K_i))}{\sum_{j=1}^l \exp(F(Q, K_j))}, \quad (3-2)$$

最后进行加权求和得到最终的注意力值。注意到最终的注意力值是键值对序列 $(K_i, V_i)_{i=1}^l$ 的特征表示,其中和查询 Q 更相似的部分被认为更重要,在最终的特征表示中拥有更大的权重。在实际应用中,通常有 $K_i = V_i$ 。

根据任务的不同,用于计算相似度的函数 F 往往会呈现不同的形式。常用的相似度函数有点积和神经网络两种类型。点积函数即直接计算查询 Q 与键值的 K_i 的点积作为其相似度;而神经网络函数通常将维度为 d_k 的查询 Q 与键值的 K_i 拼接成一个维度为 $2d_k$ 的向量作为神经网络的输入,然后将网络的输出作

为其相似度。在实践中，通常点积函数的时间和空间开销比较小，但有研究表明当 d_k 比较大时，神经网络函数的性能显著优于点积函数的性能^[76]。为了缓解点积函数在 d_k 增大时性能下降的问题，Vaswani 等人提出一种放缩点积函数^[2]，对点积以 $\frac{1}{\sqrt{d_k}}$ 的比例进行放缩，即：

$$F_{\text{scaled}}(Q, K_i) = \frac{Q^T K_i}{\sqrt{d_k}}, \quad (3-3)$$

该函数能够在一定程度上避免 Softmax 函数出现梯度消失现象。

在推荐系统领域，注意力机制通常被应用于基于大量稀疏特征进行预测以及表征用户历史行为序列等场景中^[17-21,32]，旨在刻画不同特征、不同行为对推荐任务的重要程度，进而更精细化地进行推荐以提升推荐精确度。

3.2.2 图神经网络

近年来，以 CNN 和 RNN 为代表的深度学习技术在图像、视频、文本等领域的大量应用中取得了突破性的进展，得益于其在抽取欧氏空间数据局部特征并将其组合以构建整体特征表示的强大能力。然而，实际应用中仍有许多需要处理非欧氏空间数据的场景，如：推荐系统、社交网络、通讯网络等。这些场景下，数据通常被建模为图 (graphs)，数据实体被建模为图的结点 (nodes)，实体之间的关系被建模为边 (edges)。图是一类相对复杂的数据结构，其形状多变而不规则，结点数目不固定，每个结点相邻的结点数目也不固定。此外，每个实体之间不是独立的，实体之间存在多种多样的关联。在这样的特性下，适用于规则欧氏空间数据的 CNN 和 RNN 不再适用，许多重要的操作（例如卷积）无法应用于这样的数据之上，并且数据实例互相独立的假设也不成立。因此，将深度学习拓展到图数据上的研究吸引了越来越多研究者的兴趣，适用于图结构的深度学习技术——图神经网络 (Graph Neural Network, GNN) 应运而生^[77-78]。

GNN 为每个结点维护一个状态向量，通过神经网络结构迭代地向相邻结点传递状态向量，然后聚合相邻结点的状态向量以更新结点的状态向量并计算得到输出结果。形式化地，我们用 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 来表示一个图，其中 \mathcal{V} 表示图的结点集合， \mathcal{E} 表示图的边集合。对于结点 v ，我们定义其邻居为所有与 v 相邻（通过边相连接）的结点集合 $\mathcal{N}_v = \{u \in \mathcal{V} | (u, v) \in \mathcal{E}\}$ 。结点 v 的特征表示为 \mathbf{x}_v ，状态

表示为 \mathbf{h}_v . 结点集合 $\mathcal{V}' \subseteq \mathcal{V}$ 中所有结点的特征表示为 $\mathbf{x}_{\mathcal{V}'}$, 状态表示为 $\mathbf{h}_{\mathcal{V}'}$. 我们定义图神经网络的迭代过程为:

$$\begin{aligned}\mathbf{h}_v^0 &= \mathbf{x}_v, \\ \mathbf{h}_v^t &= f(\mathbf{x}_v, \mathbf{h}_v^{t-1}, \mathbf{h}_{N_v}^{t-1}), \\ \mathbf{o}_v^t &= g(\mathbf{x}_v, \mathbf{h}_v^{t-1}),\end{aligned}\tag{3-4}$$

其中, \mathbf{h}_v^t 和 \mathbf{o}_v^t 分别表示第 t 次迭代结点 v 的状态和输出, 函数 f 和 g 分别表示聚合邻居结点更新状态的转移函数和计算当前输出的函数。

推荐系统中, 用户与物品之间的交互与联系常被建模为有向图或无向图。随着 GNN 的不断发展, 许多推荐模型引入 GNN 以构建表达能力更丰富的用户或物品的特征表示, 进而提升整体的推荐性能。GC-MC^[79] 使用图卷积神经网络 (graph convolution network, 简称 GCN)^[80-82] 直接捕获用户-物品交互图中隐含的用户与物品之间的联系。sRMGCNN^[83] 利用用户-用户图及物品-物品图信息, 通过 GCN 更新用户和物品的特征表示。NGCF^[84] 构建了基于 GNN 的嵌入向量传播层结构, 从而能够捕获到高阶连接关系中的协同信号。STAR-GCN^[85] 通过堆叠 GCN 作为编码器与解码器来学得高质量的用户和物品的特征表示。

尽管这类模型受益于 GNN 强大的结点表示能力, 但它们的主要关注点均为图内部信息的特征表示, 缺乏捕获推荐过程中的动态交互与推荐上下文之间关系的能力。

3.3 多向交互图注意力网络

为了应对现有推荐排序模型存在的缺少对物品历史交互特征的利用、缺少对历史交互特征置信度的考虑、对非热门物品不友好、缺乏捕获动态交互与推荐上下文之间关系的能力等问题, 我们提出了一个多向交互图注意力网络, 简称 MIGAT, 以充分利用动态用户-物品交互特征并提升推荐排序模型的性能。

首先, 我们将用户和物品及其交互关系建模为一个无向二部图, 我们称之为用户-物品交互图。在此基础上, 我们的模型 MIGAT 基于注意力机制去捕获多个维度下历史交互中的每一项交互对当前所要预测的用户及物品的重要度。综合考虑这些重要度信息, MIGAT 通过图注意力神经网络得到基于交互信息的图

结点特征表示, 通过加权平均得到自适应历史交互特征表示。此外, MIGAT 通过置信度嵌入向量来表示不同历史交互的置信度。基于上述设计思想, 在我们的模型架构中, 交互的重要度从多个维度进行衡量, 并且重要的交互在构建用于最终预测的特征表示时具有更大的影响, 这大大提升了模型的表达能力。

下文将从特征表示、模型架构、学习算法等方面对模型 MIGAT 进行详细阐述。

3.3.1 特征表示

用户-物品交互图是建模推荐系统中用户与物品之间交互关系的数据结构, 通常被表示为二部图 $\mathcal{G} = (\mathcal{V}_u, \mathcal{V}_i, \mathcal{E})$, 其中 \mathcal{V}_u 和 \mathcal{V}_i 分别表示用户结点集合和物品结点集合, \mathcal{E} 为边集合, 对于任意 $(u, i) \in \mathcal{E}$, 都有 $v_u \in \mathcal{V}_u, v_i \in \mathcal{V}_i$, 并且用户 u 和物品 i 之间发生过交互。

为了使我们的模型适用于推荐中动态交互的过程, 我们修改原始交互图的定义以满足动态设置。我们定义用户-物品交互图为无向二部图 $\mathcal{G} = (\mathcal{V}_u, \mathcal{V}_i, \mathcal{E})$. 用户部分 \mathcal{V}_u 包含所有用户结点, 物品部分 \mathcal{V}_i 包含所有物品结点。边集 \mathcal{E} 由形如 (v_u, v_i, o) 的所有边组成, 其中 v_u 是用户结点, v_i 是物品结点, 用户 u 和物品 i 之间发生过交互, 标签 o 表示交互顺序。最早发生的交互对应边的标签为 1, 此后发生的交互对应边的标签依次递增 1。我们将结点 v 的有序邻居定义为与 v 之间有边并按相应交互顺序排列的结点序列, 即

$$\begin{aligned} \mathcal{N}_v &= (v_1, v_2, \dots, v_L) \\ \text{s.t. } \forall 1 \leq l \leq L, (v, v_l, o_l) \in \mathcal{E} \text{ or } (v_l, v, o_l) \in \mathcal{E}, \\ &\forall 1 \leq j < k \leq L, o_j < o_k, \end{aligned} \quad (3-5)$$

其中 L 是 v 的交互总数。 \mathcal{N}_v 中的每个结点称为 v 的邻居。交互图 \mathcal{G} 随着时间而变化, 在 \mathcal{V}_u 或 \mathcal{V}_i 中不断插入新结点 (遇到新用户与新物品) 并将新发生的交互插入 \mathcal{E} 中。我们将 \mathcal{G}^t 表示为时间 t 时刻的交互图。

在 MIGAT 中, 我们使用以下四组类别特征: 用户属性、物品属性、用户邻居序列与物品邻居序列。其中, 用户邻居序列是用户的有序邻居中的每一项物品属性特征构成的序列, 物品邻居序列是物品有序邻居中的每一项用户属性构

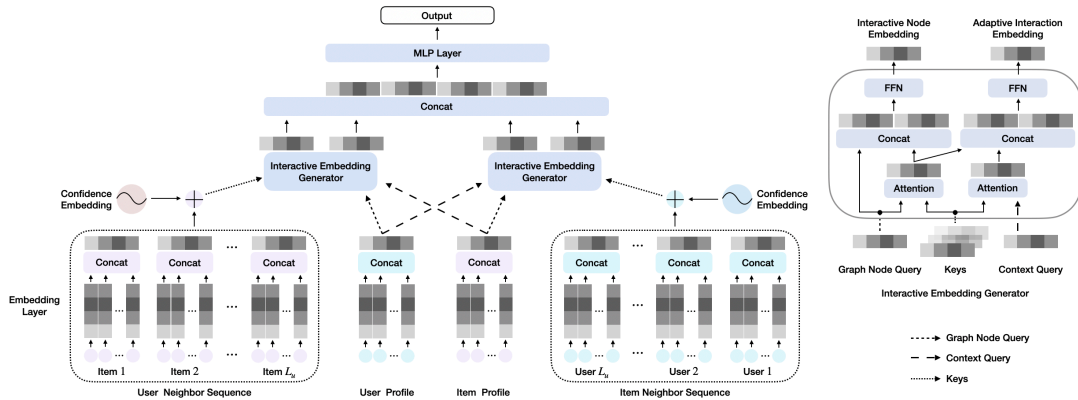


图 3-1: 模型架构。图的左半部分展示了我们提出的模型 MIGAT 的网络结构, 包括一个用来获取类别特征的嵌入表示的嵌入层, 一组用来区分不同交互的置信度嵌入向量, 一个用来生成表达能力丰富的特征表示的交互嵌入生成器, 以及最终进行预测推荐值的 MLP 层。图的右半部分展示了 MIGAT 中交互嵌入生成器的结构, 它基于注意力机制计算得到多个维度下的重要度信息, 综合考虑多种重要度信息生成表达能力丰富的特征表示。

成的序列, 这里有序邻居是指式 (3-5) 中定义的和结点有边并按相应交互顺序排列的结点序列。

3.3.2 模型架构

本节对 MIGAT 模型架构的细节进行阐述, 模型结构如图 3-1 所示。MIGAT 由四个部分组成: (1) 一个将稀疏特征转换为稠密嵌入向量的嵌入层 (embedding layer); (2) 一组置信度嵌入向量 (confidence embeddings), 用来区分序列中不同位置的交互邻居的置信度; (3) 一个交互嵌入生成器 (interactive embedding generator), 用来生成基于交互信息的图结点特征表示与自适应历史交互特征表示; (4) 一个多层感知机 (multilayer perceptron, MLP) 层, 用来预测用户对物品有偏好的概率。下文将详细介绍这四个部分。

1. 嵌入层

在嵌入层, 高维稀疏向量通过查看嵌入表的形式转化为低维稠密向量。具体地, 我们构建一个用户嵌入表 $\mathbf{E}_u = [\mathbf{e}_{u_1}, \mathbf{e}_{u_2}, \dots, \mathbf{e}_{u_{N_u}}] \in \mathbb{R}^{H_u \times N_u}$ 来表示所有的用户属性, 其中 H_u 是用户属性嵌入向量的维数, N_u 是用户属性所包含的所有类别特征的数量, $\mathbf{e}_{u_k} \in \mathbb{R}^{H_u}$ 是维数为 H_u 的嵌入向量。物品嵌入表 $\mathbf{E}_i \in \mathbb{R}^{H_i \times N_i}$ 通过类似的方式构建。

按照 2.3.1 节的方式, 我们可以分别得到用户属性、物品属性、用户邻居序列、物品邻居序列的嵌入向量表示 $\mathbf{e}_u \in \mathbb{R}^{n_u H_u}$, $\mathbf{e}_i \in \mathbb{R}^{n_i H_i}$, $\mathbf{e}_{u_{ns}} \in \mathbb{R}^{L_u \times n_i H_i}$,

$\mathbf{e}_{ins} \in \mathbb{R}^{L_i \times H_u}$, 其中 n_u (n_i) 是用户 (物品) 属性中所有类别特征的数量, L_u (L_i) 是用户 (物品) 邻居的数目。注意到用户邻居序列与物品属性共享同一个嵌入表 \mathbf{E}_i , 物品邻居序列与用户属性共享同一个嵌入表 \mathbf{E}_u , 这使得模型能够将图特征与上下文特征整合起来共同考虑。

2. 置信度嵌入向量

推荐系统中, 最近发生的交互通常比较早发生的交互更能反映用户偏好, 即拥有更高的置信度, 因而最近发生的交互应当在推荐过程中发挥更重要的作用。为了区分不同时间发生的交互的置信度, 我们在 MIGAT 中引入置信度嵌入向量结构。本文中, 我们使用如下公式对置信度嵌入向量进行初始化:

$$CE_{(l,i)} = \exp(l - L - 1) \cos((i - 1)\pi/H), \quad (3-6)$$

其中, l 表示交互发生的次序, i 表示当前单元在嵌入向量中的位置索引, L 表示序列所包含的交互总数, H 表示嵌入向量的维度。注意到式 (3-6) 由两部分组成: (1) 指数部分, 用来建模置信度随时间间隔增大而衰减的趋势; (2) 余弦部分, 用来从序列角度建模置信度信息。对于一个特定的交互, 它的置信度嵌入向量拟合了一个长度为 π 的余弦曲线。对于一个特定的嵌入向量位置, 所有这个位置上的嵌入向量单元的值随着交互发生时间的递减呈指数衰减。

注意到交互在我们的模型中被表示为有序邻居, 经过嵌入层我们得到了交互邻居嵌入序列。接下来, 相同维度的置信度嵌入向量与交互邻居嵌入序列相加作为后续使用注意力机制计算重要度信息时的输入特征。

置信度嵌入向量的设计与神经机器翻译 (Neural machine translation, NMT) 任务中经常使用的位置编码 (positional encodings)^[2,86] 有着类似的思想, 都利用了序列的顺序信息。使用正弦曲线或余弦曲线刻画序列信息的思想受到了离散傅里叶变换 (discrete Fourier transform, DFT) 理论的启发。在数字信号处理领域, DFT 经常被用来将时间信号转化为频率信号。因此, 我们选择半周期内的余弦曲线来初始化置信度嵌入向量以帮助我们的模型更轻松地捕获序列信息。此外, 我们使用指数因子来建模交互置信度衰减的现象, 而不仅仅是区分序列中的不同位置。

3. 交互嵌入生成器

MIGAT 的目的是有效刻画动态自适应的用户兴趣和物品吸引力进而提升推荐排序模型的性能。因此,不同于^[25,28]这些将嵌入向量直接输入到 MLP 网络结构进行预测的传统模型, MIGAT 设计了交互嵌入生成器结构来生成表达能力更丰富且更有效的嵌入向量,然后再输入到 MLP 层中。

如图 3-1 所示,交互嵌入生成器由两部分组成:(1)一个多向注意力层 (multi-directional attention layer),用来刻画结点与有序邻居之间的交互关系,以及历史交互序列与用户兴趣、物品吸引力以及推荐上下文之间的自适应关系;(2)一个聚合层 (integrate layer),用来生成图结点特征的最终表示,以及自适应历史交互序列嵌入特征表示。

在推荐系统中,用户-物品交互反映了用户兴趣和物品吸引力,可以被用来加强原有的用户/物品特征表示以及衡量其与推荐上下文之间的相关性。基于上述思想,我们设计了一个多向注意力层,来衡量每一项交互邻居对图结点及推荐上下文的重要度,并基于这些重要度进行加权求和以生成交互嵌入向量。形式化地,经过前文所述的嵌入层并与置信度嵌入向量相加,我们得到嵌入向量 $\mathbf{e}_u, \mathbf{e}_i, \mathbf{e}_{uns}$ 与 \mathbf{e}_{ins} 分别表示用户属性、物品属性、用户邻居序列与物品邻居序列。接下来,我们将这些嵌入向量作为多向注意力层的输入,计算注意力系数:

$$\begin{aligned} a_{ui}^{(l)} &= \text{softmax}(\text{FFN}_{ui}(\mathbf{e}_u, \mathbf{e}_{uns}^{(l)})), \\ a_{ua}^{(l)} &= \text{softmax}(\text{FFN}_{ua}(\mathbf{e}_i, \mathbf{e}_{uns}^{(l)})), \\ a_{ii}^{(l)} &= \text{softmax}(\text{FFN}_{ii}(\mathbf{e}_i, \mathbf{e}_{ins}^{(l)})), \\ a_{ia}^{(l)} &= \text{softmax}(\text{FFN}_{ia}(\mathbf{e}_u, \mathbf{e}_{ins}^{(l)})), \end{aligned} \quad (3-7)$$

其中, $\text{FFN}_{ui}(\cdot)$, $\text{FFN}_{ua}(\cdot)$, $\text{FFN}_{ii}(\cdot)$, 和 $\text{FFN}_{ia}(\cdot)$ 是四个结构相同但参数不同的独立的前馈神经网络 (feed-forward neural networks, FFNs), 每个 FFN 的输出都通过 softmax 函数进行规范化, softmax 函数的定义如下:

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}. \quad (3-8)$$

$a_{ui}^{(l)}$, $a_{ua}^{(l)}$, $a_{ii}^{(l)}$, $a_{ia}^{(l)}$ 分别被称为用户交互权重, 用户自适应权重, 物品交互权重, 以及物品自适应权重。注意到对于用户 (物品) 邻居序列, 其交互权重刻画了每个邻居对与之相邻的图结点的重要度, 其自适应权重刻画了每个邻居对推荐上下

文的重要程度。最后，我们基于上述注意力系数依下式计算交互嵌入表示与自适应嵌入表示：

$$\begin{aligned} \mathbf{h}_{ui} &= \sum_{l=1}^{L_u} a_{ui}^{(l)} \mathbf{e}_{uns}^{(l)}, & \mathbf{h}_{ua} &= \sum_{l=1}^{L_u} a_{ua}^{(l)} \mathbf{e}_{uns}^{(l)}, \\ \mathbf{h}_{ii} &= \sum_{l=1}^{L_i} a_{ii}^{(l)} \mathbf{e}_{ins}^{(l)}, & \mathbf{h}_{ia} &= \sum_{l=1}^{L_i} a_{ia}^{(l)} \mathbf{e}_{ins}^{(l)}, \end{aligned} \quad (3-9)$$

其中， L_u, L_i 分别表示用户和物品的邻居数目。

经过多向注意力层之后，交互嵌入向量 $\mathbf{h}_{ui}, \mathbf{h}_{ii}$ ，自适应嵌入向量 $\mathbf{h}_{ua}, \mathbf{h}_{ia}$ 与原始嵌入向量 $\mathbf{e}_u, \mathbf{e}_i$ 在聚合层进行整合。具体地，交互嵌入向量和原始嵌入向量通过 GNN 进行聚合以生成交互式结点嵌入向量。在我们的模型中，包含加权交互邻居特征的交互嵌入向量和原始的结点特征嵌入向量拼接起来，然后通过单层的前馈神经网络生成最终的交互式结点嵌入表示：

$$\begin{aligned} \mathbf{h}_u &= \text{LeakyReLU}(\mathbf{W}_u[\mathbf{e}_u \parallel \mathbf{h}_{ui}] + \mathbf{b}_u), \\ \mathbf{h}_i &= \text{LeakyReLU}(\mathbf{W}_i[\mathbf{e}_i \parallel \mathbf{h}_{ii}] + \mathbf{b}_i), \end{aligned} \quad (3-10)$$

其中， $\mathbf{W}_u \in \mathbb{R}^{n_i H_i \times 2n_i H_i}$ ， $\mathbf{W}_i \in \mathbb{R}^{n_u H_u \times (n_u + 1) H_u}$ 分别表示用来生成用户交互嵌入表示和物品交互嵌入表示的前馈神经网络的连接权重 (weight)， $\mathbf{b}_u \in \mathbb{R}^{n_i H_i}$ ， $\mathbf{b}_i \in \mathbb{R}^{n_u H_u}$ 分别表示前馈神经网络的偏置 (bias)。此外，全连接层的输出经激活函数 LeakyReLU^[87] 处理作为最终的输出。类似地，交互嵌入向量与自适应嵌入向量拼接起来，通过前馈神经网络生成自适应历史交互特征表示向量 \mathbf{h}'_u 和 \mathbf{h}'_i ：

$$\begin{aligned} \mathbf{h}'_u &= \text{LeakyReLU}(\mathbf{W}'_u[\mathbf{h}_{ui} \parallel \mathbf{h}_{ua}] + \mathbf{b}'_u), \\ \mathbf{h}'_i &= \text{LeakyReLU}(\mathbf{W}'_i[\mathbf{h}_{ii} \parallel \mathbf{h}_{ia}] + \mathbf{b}'_i). \end{aligned} \quad (3-11)$$

4. 多层感知机层

在多层感知机层，经交互嵌入生成器生成的交互结点嵌入表示 $(\mathbf{h}_u, \mathbf{h}_i)$ 与自适应历史交互特征表示 $(\mathbf{h}'_u, \mathbf{h}'_i)$ 拼接起来作为多层感知机的输入，然后由多层感知机计算得到预测的用户对物品有偏好的概率作为最终的输出。

3.3.3 学习算法

给定训练样本 (\mathbf{x}, y) , 我们的目标是当 $y = 1$ 时, 最大化预测概率 $\hat{y} = f(\mathbf{x})$, 当 $y = 0$ 时最小化预测概率 \hat{y} . 因此, 我们采用负对数似然函数作为我们的损失函数 (loss function), 其定义如下:

$$\mathcal{L} = -\frac{1}{n} \sum_{(\mathbf{x}, y) \in \mathcal{D}} (y \log f(\mathbf{x}) + (1 - y) \log(1 - f(\mathbf{x}))), \quad (3-12)$$

其中 \mathcal{D} 表示训练集。该函数也被称作二值交叉熵 (binary cross entropy, BCE) 损失函数。

基于 BCE 损失函数, 我们使用反向传播 (back-propagating, 简称 BP) 算法^[88] 对 MIGAT 模型中的参数进行学习。

3.4 实验与分析

在本节, 我们通过在四个公共数据集上进行实验以验证本章所提出模型 MIGAT 的有效性。首先, 我们介绍实验设置的细节, 然后呈现实验结果并对其进行分析。实验显示 MIGAT 在用户偏好预测任务上的表现优于最先进的 (state-of-the-art) 推荐排序模型。

3.4.1 实验设置

本节我们对实验使用的数据集、对比方法、实现细节与参数设定进行详细介绍, 然后给出用来衡量推荐排序模型效果的评价指标。

1. 数据集

在实验中, 我们使用了两个基准数据集 (benchmark datasets) 和两个比赛数据集 (grand challenge datasets) 进行实验以验证本章所提出模型的有效性。

- **Amazon**^①. Amazon 数据集是一个被广泛使用的基准数据集^[17,89-90], 它包含了亚马逊电商平台中的物品评论 (product reviews) 及元数据 (metadata) 信息。数据集中任意用户与物品都至少出现在 5 条评论中。我们的实验在其中两个

^①<http://jmcauley.ucsd.edu/data/amazon/>

子数据集 Books 和 Electronics 上进行。我们选择评论者的 ID 作为用户属性信息，物品的 ID 和类别作为物品属性特征，将用户评论过的物品作为用户邻居序列，将评论物品的用户作为物品邻居序列。此外，我们将总评分超过 3 分的样本划分为正例，评分不超过 3 分的样本划分为负例。

- **Byte-Recommend**^①. Byte-Recommend 数据集是一个大规模公开数据集，它包含了数十万用户和数千万短视频的信息。我们的实验在其中两个用于实际比赛的不同规模的子集上进行，我们分别将较小规模的记为 Byte-Rec-Small，较大规模的记为 Byte-Rec-Large。我们选择用户的 ID 及使用设备作为用户属性，短视频的 ID 及作者 ID 作为物品属性，将用户观看过的短视频作为用户邻居序列，将观看短视频的用户作为物品邻居序列。我们直接使用数据集中是否看完视频的指示标签作为数据集样本标签。

上述数据集的统计信息见表 3-1。其中用户密度和物品密度分别指数据集中用户的平均出现次数和物品的平均出现次数。注意到 Byte-Recommend 数据集的物品密度远小于 Amazon 数据集的物品密度，这意味着 Byte-Recommend 数据集中存在着更严重的长尾问题。

表 3-1: 数据集统计信息

数据集	样本数	用户数	物品数	用户密度	物品密度
Electronics	1,689,188	192,403	63,001	8.8	26.8
Books	8,898,041	603,668	367,982	14.7	24.2
Byte-Rec-Small	19,622,340	73,974	4,122,689	277.5	5.3
Byte-Rec-Large	275,855,531	636,281	27,342,248	433.5	10.1

我们将数据按时间顺序进行排序，然后按 8:1:1 的比例将数据集划分为训练集、验证集与测试集，其中验证集用来训练超参数 (hyper parameters)，最终性能的比较在测试集上进行。

2. 对比方法

我们将与以下模型进行对比以验证本章所提出模型 MIGAT 的性能：

- **FM**^[24]. FM 是经典的预测模型，它通过内积刻画成对特征之间的关系。
- **YoutubeNet**^[3]. YoutubeNet 是经典的遵循 Embedding&MLP 范式的深度学习模型。

^①<https://biendata.com/competition/icmechallenge2019/data/>

- **DeepFM**^[62]. DeepFM 由 FM 部分与深度部分构成，分别建模了低阶特征关系与高阶特征关系，并且两部分结构共用嵌入向量表示。
- **DIN**^[17]. DIN 是当前用户偏好预测任务最先进的模型之一，它使用注意力机制学习用户行为与候选物品之间的关系。我们在实验中将用户交互邻居作为用户行为，物品交互邻居作为上下文特征对模型进行训练。

对于 FM, 我们分别在使用交互邻居序列和不使用交互邻居序列（记为 FM-）的情况下进行实验；对于 YoutubeNet, 我们分别在使用物品交互邻居序列和不使用物品交互邻居序列（记为 YoutubeNet-）的情况下进行实验，用来验证使用交互邻居序列特征的有效性。在 FM 和 DeepFM 中，交互邻居序列被当做和其他特征无差别的系数特征。在 YoutubeNet 中，交互邻居序列通过平均池化 (average pooling) 操作转化为定长嵌入向量表示。

3. 实现细节与参数设定

我们的任务是在每个数据集上预测标签正负。我们只为每个用户或物品保留最近的 10 条交互（交互图中拥有最大标签的十条边），不足 10 条交互的情况则保留所有交互。为了实验的公平性，我们使用深度学习框架 Pytorch^[91] 实现所有模型，并使用 Adam 优化器^[92] 对其进行优化。Byte-Recommend 数据集的实验中嵌入向量维度设置为 64，Amazon 数据集的实验中嵌入向量维度设置为 128。对于 YoutubeNet, DeepFM, DIN 和 MIGAT，用于预测的多层感知机层设置为包含三层大小分别为 128, 64, 1 的隐藏层的结构。对比实验中，DIN 和 MIGAT 均使用三层全连接层计算注意力系数。训练批量大小设置为 4096。学习率设置范围为 $\{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$ ，每 1 到 2 个轮次 (epoch) 按衰减率进行衰减，衰减率设置范围为 $\{0.1, 0.2, 0.5, 1.0\}$ 。为了缓解过拟合问题，我们使用了 L_2 正则化及 Dropout 技术^[70]，正则化系数设置范围为 $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ ，Dropout 中神经元丢弃概率设置范围为 $\{0.0, 0.1, 0.2, 0.5\}$ ，并且 Dropout 操作仅应用于多层感知机层的输入数据。

所有实验均在不同随机种子下运行 5 次，下文汇报的结果为 5 次实验的平均结果。

4. 评价指标

我们在实验中使用 ROC 曲线下面积 (Area Under ROC Curve, AUC) 作为评价指标。AUC 衡量了模型的排序性能^[93]，被广泛应用于推荐系统任务的性能评

估中，其定义如下：

$$\text{AUC} = \frac{1}{m^+m^-} \sum_{\mathbf{x}^+ \in \mathcal{D}^+} \sum_{\mathbf{x}^- \in \mathcal{D}^-} R(f(\mathbf{x}^+), f(\mathbf{x}^-)), \quad (3-13)$$

其中 \mathcal{D}^+ 是大小为 m^+ 的所有正样本集合， \mathcal{D}^- 是大小为 m^- 的所有负样本集合， $f(\cdot)$ 是预测模型， $R: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ 是用来计算一对输入的排序得分的函数，其定义如下：

$$R(x_1, x_2) = \mathbb{I}(x_1 > x_2) + \frac{1}{2}\mathbb{I}(x_1 = x_2), \quad (3-14)$$

其中 $\mathbb{I}(\cdot)$ 是式 (A-1) 定义的指示函数。

3.4.2 对比实验

本节通过对比本章所提出的 MIGAT 模型与当前最先进的推荐排序模型在用户偏好预测任务上的性能，以及针对长尾物品的推荐性能，验证 MIGAT 模型的有效性，并分析取得性能提升的关键所在。首先，我们在表 3-2 中给出在全体测试集样本上的整体对比实验结果。

表 3-2: 四个数据集上整体对比实验的 AUC 结果

模型	Books	Electronics	Byte-Rec-Small	Byte-Rec-Large
FM-	0.6589	0.6280	0.6822	0.6038
FM	0.6753	0.6182	0.6979	0.6445
YoutubeNet-	0.7639	0.6994	0.7312	0.6965
YoutubeNet	0.7671	0.7008	0.7385	0.7142
DeepFM	0.7678	0.7004	0.7390	0.7119
DIN	0.7679	0.7009	0.7385	0.7124
MIGAT	0.7686	0.7013	0.7422	0.7161

从表 3-2 中我们可以得到以下结论：(1) YoutubeNet 在四个数据集上的表现均优于 YoutubeNet-，FM 在两个较大数据集上的表现优于 FM-，这验证了在推荐模型中引入用户-物品交互信息的必要性。(2) 所有深度模型的性能均优于 FM，这说明了仅使用内积刻画低阶特征关系不足以很好地进行用户偏好预测。DeepFM 在 Byte-Rec-Small 和 Books 数据集上的表现优于 YoutubeNet，但在 Electronics

和 Byte-Rec-Large 数据集上呈现劣势，这说明尽管模型考虑了较为复杂的特征关系，但仅对交互邻居序列进行和其他特征一样的处理仍不足以充分利用交互信息提升模型性能。DIN 在 Books 和 Electronics 数据集上显著提升了性能，这得益于其用来抽取用户兴趣表示的特殊结构的有效性。然而，在 Byte-Rec-Small 和 Byte-Rec-Large 数据集上 DIN 表现不佳，这说明使用用户历史交互信息仍不足以解决长尾物品推荐问题，这种情况下，物品的嵌入向量没能得到充分的训练进而表达能力不足，于是用户历史交互通过一系列物品的嵌入向量进行表示时也就不能很好的反映用户真正的偏好。(3) 我们提出的模型 MIGAT 在四个数据集上均取得了最好表现，尤其是在包含大量长尾物品的 Byte-Rec-Small 和 Byte-Rec-Large 数据集上表现更佳。MIGAT 在训练中学得交互邻居对图结点的重要度以及对推荐上下文的自适应关系，因此能够得到更有效的交互特征嵌入表示对用户偏好和物品吸引力进行表达。

为了进一步验证模型 MIGAT 在长尾物品推荐问题上的有效性，我们计算了长尾子集上的 AUC 值，这里长尾子集是指在训练集中拥有不超过 k 条交互的物品对应的测试样本集合。 k 被称作长尾阈值，在 Books 和 Electronics 数据集中我们分别设置为 10, 25, 50 进行实验，在 Byte-Rec-Small 和 Byte-Rec-Large 数据集中我们分别设置为 3, 5, 10 进行实验。 k 的设置基于物品分布情况，因此我们在长尾问题严重的 Byte-Rec-Small 和 Byte-Rec-Large 数据集上的实验设置了更小的 k 。实验结果如图 3-2 所示，注意到我们省略了 FM 的结果，因为其他模型的性能均显著超过 FM。

从图 3-2 中可以观察到，不同模型在每个数据集上的性能表现呈现相同的趋势。MIGAT 在大部分数据集上都有最优的性能，尤其是在长尾现象严重的数据集上性能提升尤为明显。这验证了我们的交互嵌入生成器结构所生成的嵌入向量表示对于解决长尾物品推荐问题的有效性。此外，MIGAT 在长尾阈值为 10 的情况下在数据集 Electronics 上的性能提升显著大于长尾阈值更大的情况，进一步证明了在物品拥有极少交互的情况下，MIGAT 具有生成表达能力足够强的特征表示的能力。

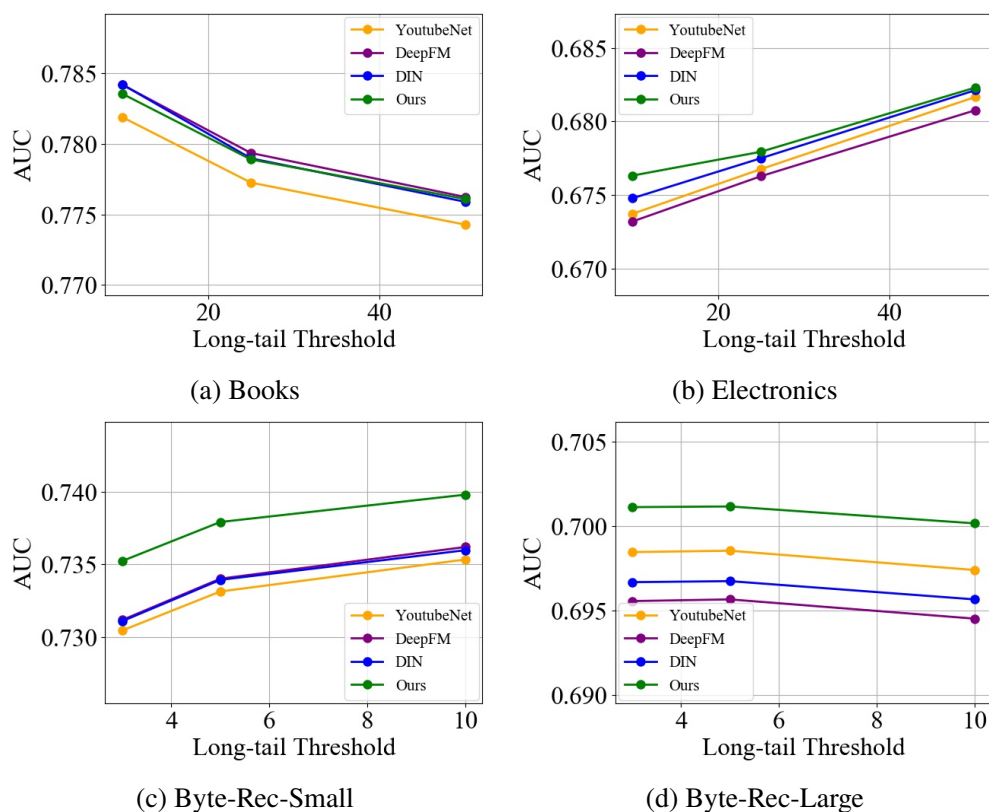


图 3-2: 不同长尾阈值下的性能对比 (以 AUC 度量)

3.4.3 消融实验

本节我们分别通过三组实验验证使用动态交互特征的有效性、置信度嵌入的有效性以及注意力函数对模型性能的影响。下面将依次展示这三组实验的结果并对其进行分析。

1. 使用动态交互特征的有效性

为了验证使用动态交互特征的有效性，我们在实验中将每个用户或物品的交互邻居序列替换为静态的交互邻居，即训练集中该用户或物品最近发生的 10 条交互的交互对象。实验结果如表 3-3 所示。从表中可以看到，使用静态交互特征后预测性能大幅下降。我们认为，导致该现象发生的原因是静态交互与动态用户兴趣或物品吸引力之间的不匹配。这验证了我们的模型引入动态交互特征进行用户偏好预测的必要性。

2. 置信度嵌入的有效性

为了研究置信度嵌入向量对预测性能的影响，我们在以下替换置信度嵌入向量形式的变种模型上进行实验：

表 3-3: 动态交互特征的效果 (以 AUC 度量)

模型	Books	Electronics	Byte-Rec-Small	Byte-Rec-Large
静态特征	0.7677	0.6960	0.7234	0.7023
动态特征	0.7686	0.7013	0.7422	0.7161

- (1) 移除置信度嵌入向量的变种, 记为 None;
- (2) 用^[2]中提出的位置编码替换置信度嵌入向量的变种, 记为 PE;
- (3) 用依式 (3-6) 计算得到的非可学习的置信度嵌入向量替换可学习的置信度嵌入向量的变种, 记为 FCE;
- (4) 用随机初始化的可学习的向量替换置信度嵌入向量的变种, 记为 RCE.

全体测试集上的实验结果如表 3-4 所示, 长尾设定下 Electronics 和 Byte-Rec-Small 数据集上的实验结果如图 3-3 所示。

表 3-4: 置信度嵌入向量的效果 (以 AUC 度量)

模型	Books	Electronics	Byte-Rec-Small	Byte-Rec-Large
NONE	0.7675	0.7002	0.7416	0.7147
FCE	0.7668	0.6928	0.7414	0.7152
RCE	0.7678	0.7002	0.7416	0.7153
PE	0.7587	0.6960	0.7399	0.7142
MIGAT	0.7686	0.7013	0.7422	0.7161

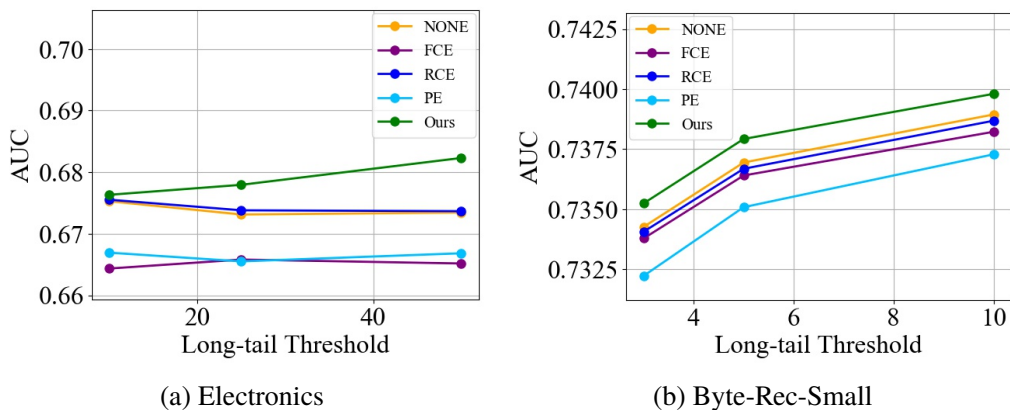


图 3-3: 置信度嵌入向量在长尾设定下的效果 (以 AUC 度量)

从实验结果中我们可以观察到, MIGAT 在所有数据集上表现最优, 这验证了我们提出的置信度嵌入向量的有效性。使用非可学习的置信度嵌入向量性能

出现明显下降，这意味着交互置信度的多样性可能无法通过一个统一的规则进行概况，需要根据实际进行调整。FCE 在大部分情况下的性能表现优于 PE，这说明我们设计的置信度嵌入向量比位置编码更适用于建模交互置信度信息。RCE 性能仅次于 MIGAT，这进一步验证了设计专门的交互置信度嵌入向量初始化方式并使得嵌入向量可学习的必要性。

3. 注意力函数对模型性能的影响

为了研究注意力函数对模型预测性能的影响，我们分别在使用点积函数、缩放点积函数^[2]和前馈神经网络作为注意力函数中的相似度计算函数的模型上进行实验。根据其使用的注意力函数，我们将上述模型分别记为 Dot-product, Dot-product-S, FFN- n , 其中 n 为所使用的前馈神经网络的层数。全体测试集上的实验结果如表 3-5 所示，长尾设定下 Electronics 和 Byte-Rec-Small 数据集上的实验结果如图 3-4 所示。

表 3-5: 使用不同注意力函数的模型性能 (以 AUC 度量)

模型	Books	Electronics	Byte-Rec-Small	Byte-Rec-Large
FFN-1	0.7674	0.7016	0.7418	0.7161
FFN-2	0.7677	0.7015	0.7419	0.7159
Dot-product	0.7685	0.7007	0.7419	0.7151
Dot-product-S	0.7666	0.7011	0.7417	0.7153
FFN-3	0.7686	0.7013	0.7422	0.7161

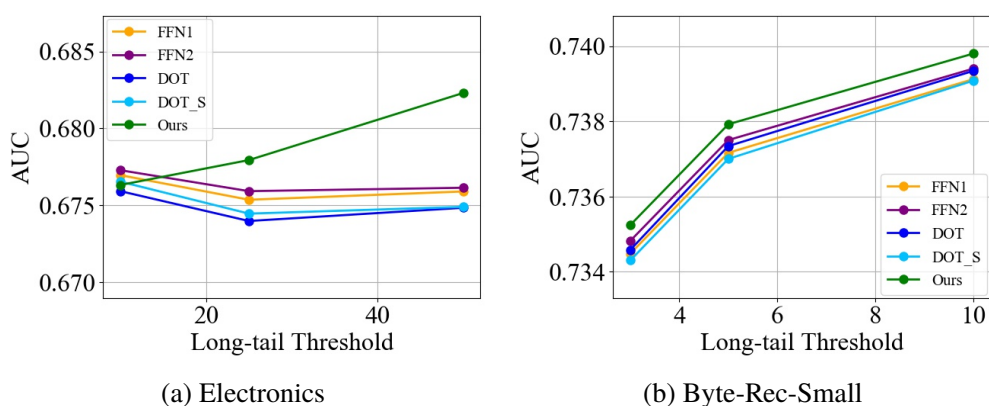


图 3-4: 长尾设定下使用不同注意力函数的模型性能 (以 AUC 度量)

从实验结果中我们可以观察到，使用前馈神经网络总体性能优于使用（缩放）点积函数作为注意力函数的相似度计算函数，并且FFN-3在除规模最小的

Electronics 数据集外的数据集上均取得最优性能。这意味着数据集中普遍存在图结点表示与上下文特征之间的高阶特征关系。此外,对于 Electronics 数据集,尽管FFN-3没有在测试集全集上取得最优性能,但在长尾设定下它仍然比使用其他注意力函数的模型有着更显著的提升。这验证了在存在长尾问题的情况下,模型在计算交互重要度时捕获高阶特征关系的必要性。

3.4.4 综合分析 with 讨论

通过综合分析表 3-2、表 3-4、表 3-5、图 3-2、图 3-3 和图 3-4,我们可以得到以下结论:(1)对于 Byte-Rec-Small 和 Byte-Rec-Large 数据集,替换置信度嵌入向量或注意力函数不影响模型在与其他对比模型的对比实验中取得最优性能,这显示了我们所提出模型在长尾设定下的优越性。(2)对于长尾现象不够严重的数据集 Books 和 Electronics,使用置信度嵌入向量以及选择合适的注意力函数对模型性能的提升至关重要。这说明了置信度嵌入向量及注意力函数对于从较长的交互序列中捕获用户偏好和物品吸引力的情况具有比较大的影响力。

3.5 本章小结

本章我们提出了一个基于注意力机制与图神经网络的用户偏好预测模型 MIGAT. MIGAT 通过注意力机制生成表达能力更丰富的自适应用户-物品交互特征表示,以及交互式的用户/物品特征表示。为了捕获动态的用户兴趣偏好以及物品吸引力,我们使用动态用户-物品交互图而非静态的交互图。模型 MIGAT 的关键在于从两个角度考虑每个交互邻居的重要度,即对图结点的重要度,以及对当前推荐上下文及推荐对象的重要度。对前者的考虑,使得模型能够生成表达能力更加丰富的用户/物品特征;综合考虑二者,用户兴趣偏好、物品吸引力以及推荐上下文之间的复杂关系得以被发掘,由此生成自适应的用户-物品交互特征表示。此外,我们设计了一种交互置信度向量结构及其初始化方式,以建模不同交互的置信度信息。四个数据集上的实验显示,基于上述思考设计的模型 MIGAT 在用户偏好预测任务上的性能优于其他先进的推荐排序模型,并且在长尾现象严重的数据集上性能提升更为明显。

第四章 基于模拟交互序列生成的冷启动推荐模型

上一章我们提出了一个多向交互图注意力网络推荐模型 MIGAT。尽管该模型能够充分利用用户和物品的历史交互信息构建表达能力更丰富的特征表示，并且实验验证了该模型在用户偏好预测任务上的有效性，但由于模型结构依赖于用户和物品的历史交互特征，在面对推荐系统中的新用户或新物品时模型结构将部分失效，从而在一定程度上限制了模型性能的提升。为此，本章提出一个基于元学习的伪交互序列生成模型 (Meta-Learned Pseudo Interaction Generator, 简称 MePIGen)，旨在推荐系统的冷启动阶段，通过在尽可能少的探索次数下训练更新，为新用户或新物品生成高质量的伪历史交互序列，帮助常规推荐模型提升用户偏好预测性能。

4.1 研究动机

推荐系统中，新的用户与物品不断出现，然后在系统中与其他物品或用户之间产生交互，逐渐成为老用户与老物品。一般地，我们称新用户或新物品刚进入系统至其积累一定量的交互之前的过程为冷启动阶段，称积累一定量的交互之后的过程为常规阶段。主流推荐模型几乎都建立在推荐系统的常规阶段下，因为常规阶段下每个用户与物品相关的信息已得到足够的积累，所以模型能够集中关注问题的核心与本质，基于丰富的数据得到令人满意的推荐效果。然而，大部分模型在冷启动阶段都表现不佳。由于缺少足够的历史交互信息，一方面大量基于交互信息进行预测的模型将无法正常工作；另一方面模型由于新用户或新物品的 ID 特征在推荐系统中出现频次过低，对应的特征表示无法得到充分的训练进而影响最终的预测效果。因此，如何提升模型在冷启动阶段的性能（冷启动问题）已成为推荐系统领域的一项挑战。

解决冷启动问题有以下两个关键点：(1) 在涉及到全新的用户或物品的样本时，模型拥有尽可能好的预测性能；(2) 模型在新用户与新物品逐渐积累交互的

过程中能够实现有效更新, 并尽快收敛达到常规阶段的性能表现。其中, (1) 主要关注的是冷启动的初始化问题, 它保证了针对新用户或新物品的推荐结果是相对有保障的, 避免新用户或新物品因较差的推荐结果而从系统中流失; (2) 主要关注的是冷启动的收敛问题, 它保证了模型达到令人满意的性能所需要的数据不会过多, 也即探索的次数不需要过多, 避免新用户或新物品在漫长的探索过程中失去对系统的信心。

为了解决冷启动问题, 传统方法引入辅助信息进行推荐, 例如引入丰富的用户属性和物品属性信息^[36-38,94], 社交平台信息^[39-40], 或异质信息关系网络^[41-42]。这类方法主要关注了冷启动的初始化问题, 它们从数据的角度出发, 基于引入的辅助信息预测新用户或新物品可能的偏好或受众, 因而可以与具有相同属性的老用户或老物品之间建立联系, 获得相对可信的推荐结果。这类方法会导致拥有相似属性的用户或物品的推荐结果总是相似的。但在实际中, 属性相同的用户或物品可能依然会具有非常不同的行为模式。这种情况下, 这类引入辅助信息进行推荐的模型的性能将严重受限。

近年来, 越来越多的推荐算法开始关注冷启动的收敛问题, 希望模型能够在仅拥有少量交互数据的情况下尽快收敛至一个令人满意的结果。该思想与元学习 (meta-learning) 的思想不谋而合。元学习通过从不同的学习任务 (learning tasks) 中总结出通用的规律 (先验知识), 然后基于该规律及少量涉及到新任务的数据进行训练, 达到快速适应新任务的目的^[74]。我们将每个用户或物品的推荐看作一个学习任务, 那么冷启动问题便可以表示为一个元学习问题。因此, 基于元学习的推荐模型逐渐成为当前推荐系统领域的研究热点^[43,55-57]。这些模型几乎都采用模型无关元学习 (model-agnostic meta-learning, MAML)^[1] 框架, 在冷启动阶段更新模型参数, 保证参数可以尽快收敛, 并且能够作为常规阶段的初始化参数在后续的推荐过程中提供有保障的性能。

另一方面, 用户与物品的历史交互信息往往在推荐中起到至关重要的作用, 但在冷启动场景下该信息无法获取。如果模型的设计中考虑到对历史交互信息的利用, 那么在冷启动的探索过程中逐渐积累的交互信息将会为模型的更新方向提供有力的帮助。此外, 这些交互信息还能够帮助构建历史交互信息的替代特征, 从而更方便地与常规阶段的模型之间建立联系。然而, 现有的针对冷启动问题设计的模型很少考虑交互信息的利用。

4.2 关键技术

为了保证冷启动阶段能够有效收敛,我们基于序列到序列模型 Transformer^[2]生成高质量的伪历史交互序列特征。本节将对序列到序列模型的基本架构及序列生成所使用的贪心搜索解码算法进行简单介绍。

4.2.1 序列到序列模型

序列到序列 (sequence to sequence, Seq2seq) 模型是一类根据给定序列生成另一个序列的模型,被广泛应用于机器翻译、自动摘要、对话系统等自然语言生成 (natural language generation, NLG) 任务中^[13,95-96]。如图 4-1 所示, Seq2seq 模

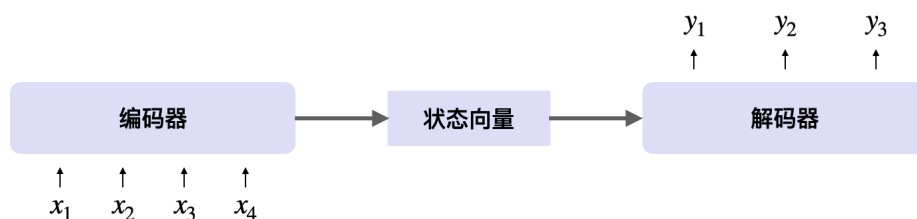
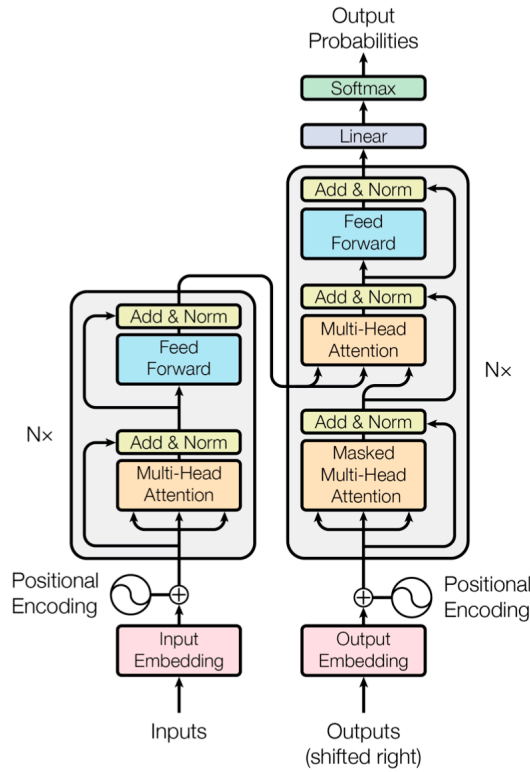


图 4-1: 编码器-解码器框架示意图

型通常采用 编码器-解码器 (encoder-decoder) 框架。其中, 编码器将输入序列编码成一个状态向量, 解码器将此状态向量作为输入最后解码得到输出序列。注意到, 模型的输入与输出是可变长度的序列, 并且二者的长度也不需要相同。

传统的 Seq2seq 模型都是基于 RNN 及其变种进行建模的, 因而存在无法并行化的问题, 在实际应用中严重影响了模型的运行效率。Vaswani 等人提出了一种完全基于注意力机制的语言模型 Transformer^[2], 避免了无法并行化的问题, 并在 NLG 任务上取得更优的性能表现。Transformer 的结构如图 4-2 所示。

Transformer 的编码器和解码器分别由 $N = 6$ 个相同的层 (layer) 堆叠而成。其中, 编码器由两个子层 (sub-layer) 构成, 分别是多头自注意力 (multi-head self-attention) 网络和全连接前馈神经网络 (fully connected feed-forward network)。其中, 多头注意力机制将输入的查询 Q , 键值对 K_i, V_i 平均分成 h 部分, 然后分别对每一部分向量进行线性变换并计算其注意力值, 最后将 h 个注意力值拼接

图 4-2: Transformer 模型结构^[2]

起来，通过线性变换得到最终的注意力值：

$$\text{MultiHead}(Q, K_i, V_i) = \parallel_{j=1}^h \left(\text{Attention} \left(QW_j^Q, KW_j^K, VW_j^V \right) \right) W^O, \quad (4-1)$$

其中， \parallel 表示拼接操作， W_j^Q, W_j^K, W_j^V, W^O 分别表示对查询、键、值及输出结果的映射函数。自注意力机制则要求 $Q = K_i = V_i$ 。多头自注意力网络中注意力函数使用如式 (3-3) 所示的缩放点积函数。全连接前馈神经网络分别对序列中每个位置进行相同的非线性变换，其包含两层全连接层并且两层之间通过激活函数 $\text{ReLU}^{[97]}$ 进行非线性变换：

$$\text{FFN}(\mathbf{x}) = W_2 \text{ReLU}(W_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2 = W_2 \max(0, W_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2, \quad (4-2)$$

其中 $(W_1), (W_2), \mathbf{b}_1, \mathbf{b}_2$ 分别为两层全连接层的参数。解码器由三个子层构成，除了与编码器相同的两个子层外，增加了一个多头注意力网络对编码器输出的状态向量进行处理，并且在多头自注意力网络层进行掩码 (mask) 操作，保证位置 i 的预测仅依赖于 i 之前位置的输出。此外，在编码器和解码器的每个子层后，都

依次进行残差连接 (residual connection)^[30] 和层规范化 (layer normalization)^[98] 操作。注意到模型还分别在编码器和解码器的输入序列对应的嵌入向量上增加了位置编码向量, 从而向模型中引入顺序信息, 其中位置编码函数如下:

$$\begin{aligned} PE_{(i,2j)} &= \sin\left(i/10000^{2j/d}\right), \\ PE_{(i,2j+1)} &= \cos\left(i/10000^{2j/d}\right), \end{aligned} \quad (4-3)$$

其中, i 表示在序列中的位置, j 表示在嵌入向量中的索引, d 表示嵌入向量的维度。

4.2.2 贪心搜索解码算法

对于序列生成任务, Seq2seq 模型依次计算生成序列的每一个位置上各个元素的条件概率分布, 然后通过解码算法得到最终的输出序列。以文本生成任务为例, Seq2seq 模型首先根据编码器状态向量计算词汇表中的每一个单词在生成序列的第 1 个位置上的概率分布, 然后依次基于前 $l-1$ 个位置的不同选择计算词汇表中的每一个单词在生成序列第 l 个位置上的概率分布, 最终得到生成序列每一个位置上各个单词的概率分布。解码算法根据各个位置上的概率分布按一定的策略选择最终输出的文本序列。注意到, 在上述过程中, 如果我们考察所有的单词组合构成的序列, 那么时间复杂度将随着序列长度的增加呈指数级增长。因此, 在实际应用中通常无法实现完全搜索, 也即从所有可能的序列中选择概率最大的作为输出。

实践中往往使用基于启发式搜索策略得到近似最优生成序列的解码算法。其中, 贪心搜索 (greedy search) 算法因其原理简单、运算高效的特点, 成为实践中使用最广泛的解码算法之一。贪心搜索算法在每个位置上均选择概率最大的一项, 将这些项依次排列构成的序列作为输出, 具体步骤如算法 4.1 所示。

4.3 基于元学习的伪交互序列生成模型

为了应对常规阶段推荐模型在冷启动阶段因无法获取交互特征而结构失效, 而冷启动模型没有充分考虑交互信息, 二者难以进行融合的问题, 我们提出了

Algorithm 4.1 贪心搜索算法

输入: 全体元素集合 \mathcal{A} , 最大序列长度 L , 状态向量 $\mathbf{h}_{\text{state}}$, 解码器模型 f_{dec} , 终结符 $a_{\text{eos}} \in \mathcal{A}$.

输出: 生成序列 S .

初始化 S 为空。

for $l = 1 \rightarrow L$ **do**

 对于 $\forall a \in \mathcal{A}$, 通过解码器模型计算当前概率 $\Pr[a|\mathbf{h}_{\text{state}}, S] = f_{\text{dec}}(a, \mathbf{h}_{\text{state}}, S)$.

 选择概率最大的元素 $a^* = \operatorname{argmax}_{a \in \mathcal{A}} \Pr[a|\mathbf{h}_{\text{state}}, S]$.

 将 a^* 添加到 S 末尾, 得到更新后的 $S \leftarrow S, a^*$.

if a^* 为终结符 a_{eos} **then**

 结束算法.

end if

end for

一个基于元学习的伪交互序列生成模型, 简称 MePIGen, 通过为新用户或新物品生成伪历史交互序列并快速调整以适应于新用户与新物品的预测任务, 减少冷启动阶段的探索次数, 弥补常规阶段推荐模型交互特征缺失的问题, 提升模型的预测性能。

我们基于元学习设计了冷启动推荐算法, 以每个用户或物品相关的数据样本作为一个元学习任务, 基于用户或物品已有的少量历史交互序列使用基于 Transformer 的结构生成一个伪历史交互序列, 替换常规推荐模型中对应的历史交互特征表示。基于元学习的算法保证了生成伪历史交互序列的结构中的参数可以通过少量样本微调之后快速适应于新任务, 从而减少所需要的探索次数, 尽快得到高质量的伪历史交互序列用于后续的预测。基于 Transformer 的结构则保证了伪历史交互序列的生成质量从始至终都是相对可信的。

下文将对模型的特征表示、整体架构以及元学习算法进行详细阐述。

4.3.1 特征表示

在 MePIGen 中, 我们使用用户属性、物品属性、用户历史交互序列、物品历史交互序列四组类别特征。其中, 用户历史交互序列与物品历史交互序列均为按时间顺序排列的最近发生的 L 条交互, 不足 L 条的则以指定的无效编号进行填充。每组类别特征均通过 2.1 节所述的 One-hot 编码或 Multi-hot 编码进行表示。

4.3.2 模型架构

本节对 MePIGen 模型架构的细节进行阐述。基于 MePIGen 的冷启动推荐模型架构如图 4-3 所示。

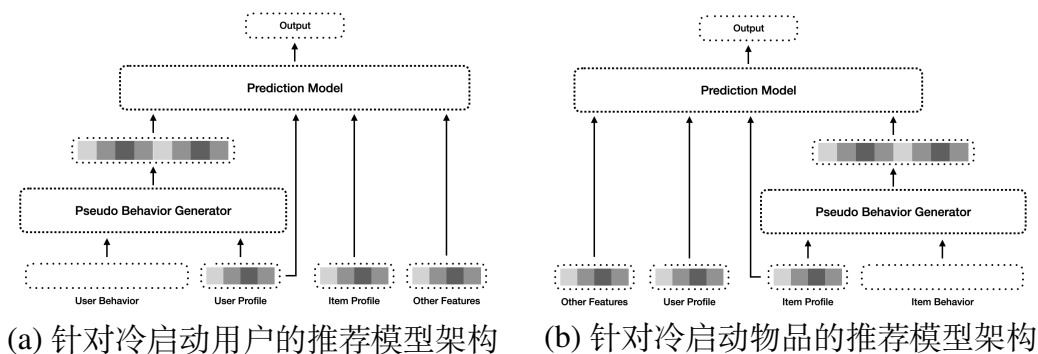


图 4-3: 基于 MePIGen 的冷启动推荐模型架构

我们在基于 Embedding&MLP 架构的推荐排序模型的基础上, 通过 MePIGen 模型生成伪历史交互序列, 替换冷启动用户或冷启动物品缺少的历史交互序列特征并用于后续神经网络的运算中, 从而实现针对冷启动用户与冷启动物品的推荐。

在嵌入层, 高维稀疏类别特征通过与 3.3.2 节类似的方式转化为稠密嵌入向量特征, 从而得到用户属性特征 \mathbf{e}_u , 物品属性特征 \mathbf{e}_i , 用户历史交互特征 \mathbf{e}_{I_u} , 物品历史交互特征 \mathbf{e}_{u_i} . 其中, 物品历史交互特征与用户属性共享用户嵌入表 \mathbf{E}_u , 用户历史交互特征与物品属性共享物品嵌入表 \mathbf{E}_I .

此后, 对于用户冷启动问题, MePIGen 模型基于用户属性特征与用户历史交互特征生成伪用户交互序列特征 \mathbf{h}_u , 对于物品冷启动问题, MePIGen 模型基于物品属性特征与物品历史交互特征生成伪物品交互序列特征 \mathbf{h}_i . 注意到, MePIGen 模型位于嵌入层之后, 特异化网络结构之前, 因此其输入与输出均为稠密嵌入向量表示的特征。伪交互特征将代替冷启动用户或冷启动物品的历史交互特征, 参与到后续的特征提取与预测过程中。结合 2.4.1 节所述, MePIGen 与嵌入层共同构成整个冷启动推荐模型的元学习器, 不包含嵌入层部分的推荐排序模型则是基学习器, 元学习器使用 MePIGen 为补充缺失的历史交互特征, 从而得以向基学习器提供初始化参数, 帮助基学习器快速适应于针对新用户或新物品的用户偏好预测任务。

其中, MePIGen 模型的结构如图 4-4 所示。具体地, MePIGen 由两部分组

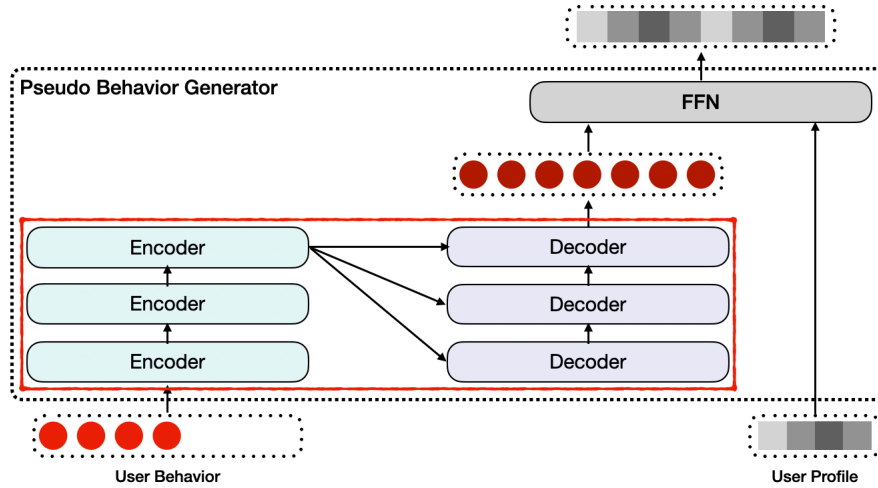


图 4-4: MePIGen 模型结构

成：(1) 一个序列生成层 (sequence generation layer)，用于根据真实历史交互序列去生成可能反映用户兴趣偏好或物品吸引力的伪交互序列；(2) 一个属性聚合层 (profile integrate layer)，用来向伪交互序列中聚合属性特征信息，增强伪交互序列与对应用户或物品的相关性。

为了生成高质量的交互序列特征，我们在序列生成层基于 Transformer 模型构建从真实的历史交互序列到伪交互序列的映射。Transformer 模型的结构细节已在 4.2.1 节中介绍，我们分别用 f_{enc} 和 f_{dec} 表示其编码器和解码器，则对于输入的嵌入向量表示的用户历史交互序列 \mathbf{e}_{I_u} ，编码器首先计算得到状态向量 $\mathbf{h}_{\text{state}_u} = f_{\text{enc}}(\mathbf{e}_{I_u})$ ，然后解码器基于 $\mathbf{h}_{\text{state}_u}$ 通过贪心搜索算法（算法 4.1）生成稀疏编码表示的用户伪交互序列 $S_u = i_1, i_2, \dots, i_{L_u}$ ，最后利用嵌入表 \mathbf{E}_I 将其转化为嵌入向量表示的用户伪交互序列 $\mathbf{e}_{S_u} = [\mathbf{e}_{i_1}, \mathbf{e}_{i_2}, \dots, \mathbf{e}_{i_{L_u}}]$ 。类似地，我们可以由物品历史交互序列 \mathbf{e}_{u_i} 生成物品伪交互序列 \mathbf{e}_{S_i} 。

注意到，序列生成层生成的伪交互序列仅与历史交互序列相关，目的是根据真实历史交互序列推测用户或物品的兴趣偏好或物品吸引力，在此基础上生成与历史交互模式相符的伪交互序列。然而，对于新用户与新物品，在其完全没有历史交互信息，或历史交互信息极少的阶段，序列生成层可利用的信息非常有限，我们需要进一步借助用户或物品的属性信息对伪交互序列特征进行调整以增强伪交互序列特征与当前用户或物品的相关性。

基于上述考虑，我们在序列生成层之后增加一个属性聚合层，通过包含一层全连接层的前馈神经网络，实现伪交互序列特征与属性信息的聚合。具体地，

我们将序列中每一项交互的嵌入向量表示与属性信息拼接起来，然后通过全连接层进行线性变换，并使用激活函数 ReLU 得到融合属性信息的交互特征：

$$\begin{aligned} \forall i' \in S_u, \quad \mathbf{h}_{u,i'} &= \text{ReLU}(\mathbf{W}_u(\mathbf{e}_{i'} \parallel \mathbf{e}_u) + \mathbf{b}_u), \\ \forall u' \in S_i, \quad \mathbf{h}_{i,u'} &= \text{ReLU}(\mathbf{W}_i(\mathbf{e}_{u'} \parallel \mathbf{e}_i) + \mathbf{b}_i), \end{aligned} \quad (4-4)$$

其中 \mathbf{W}_u 与 \mathbf{b}_u 分别表示对用户伪交互特征进行变换的全连接层参数， \mathbf{W}_i 与 \mathbf{b}_i 分别表示对物品伪交互特征进行变换的全连接层参数。最后，我们将每一项伪交互特征依顺序排列得到最终的伪交互序列特征：

$$\begin{aligned} \mathbf{h}_u &= [\mathbf{h}_{u,i_1}, \mathbf{h}_{u,i_2}, \dots, \mathbf{h}_{u,L_u}], \\ \mathbf{h}_i &= [\mathbf{h}_{i,u_1}, \mathbf{h}_{i,u_2}, \dots, \mathbf{h}_{i,L_i}]. \end{aligned} \quad (4-5)$$

得到伪交互序列特征之后，我们用其替换历史交互序列特征，参与后续推荐模型的运算过程。

4.3.3 元学习算法

我们在 2.4.2 节中定义了元学习框架下的冷启动问题，在此基础上，本节对我们基于 MAML 算法思想设计的冷启动推荐算法进行介绍。我们用 θ_p 表示作为基学习器的不含嵌入层的推荐模型的参数，用 θ_e 表示嵌入层的参数，用 θ_g 表示 MePIGen 模型的参数，则推荐排序模型、基学习器、元学习器、MePIGen 模型与整个冷启动推荐模型分别表示为 $f_{(\theta_p, \theta_e)}$, f_{θ_p} , $f_{(\theta_e, \theta_g)}$, f_{θ_g} 与 $f_{(\theta_p, \theta_e, \theta_g)}$ 。接下来，我们从用户冷启动问题的视角具体介绍我们设计的推荐算法。

首先，我们基于大量历史交互丰富的老用户数据对 $f_{(\theta_p, \theta_e)}$ 进行预训练。其中，历史交互丰富的老用户数据被定义为用户历史交互不小于 L_U 条的老用户数据样本。预训练的目的在于为基学习器学得相对可信的参数，从而在后续的元学习过程中，能够较为准确地反映元学习的学习效果。此后，我们分别基于老用户集合 \mathcal{U}^{old} 中每一个老用户 u 的前 $L_{\text{usplit}} < L_U$ 条交互数据与第 $L_{\text{usplit}} + 1$ 到第 L_U 条数据构建用户学习任务 \mathcal{T}_u 所使用的支持集 \mathcal{S}_u 于查询集 \mathcal{Q}_u 。然后，我们基于老用户任务集合 $\mathcal{T}_U^{\text{old}}$ 进行对 $f_{(\theta_e, \theta_g)}$ 进行训练。在每一轮训练中，模型都从 $\mathcal{T}_U^{\text{old}}$ 上的分布 $\rho(\mathcal{T}_u)$ 中采样一批学习任务 $\mathcal{T}_{u_1}, \mathcal{T}_{u_2}, \dots, \mathcal{T}_{u_N}$ ，对于其中的每个老用

户学习任务 \mathcal{T}_u , 模型在 \mathcal{S}_u 上计算二值交叉熵损失函数的值

$$\mathcal{L}(\theta_p, \theta_e, \theta_g) = -\frac{1}{L_{\text{usplit}}} \sum_{(\mathbf{x}, y) \in \mathcal{S}_u} (y \log f_{(\theta_p, \theta_e, \theta_g)}(\mathbf{x}) + (1 - y) \log(1 - f_{(\theta_p, \theta_e, \theta_g)}(\mathbf{x}))), \quad (4-6)$$

及其对基学习器参数的梯度 $\nabla_{\theta_p} \mathcal{L}(\theta_p, \theta_e, \theta_g)$ 。然后使用梯度下降法对参数 θ_p 进行局部更新得到

$$\theta_p^u = \theta_p - \alpha \nabla_{\theta_p} \mathcal{L}(\theta_p, \theta_e, \theta_g), \quad (4-7)$$

其中 α 为局部更新的学习率。接下来, 基于局部更新的参数, 模型在 \mathcal{Q}_u 上计算二值交叉熵损失函数的值

$$\mathcal{L}(\theta_p^u, \theta_e, \theta_g) = -\frac{1}{|\mathcal{Q}_u|} \sum_{(\mathbf{x}, y) \in \mathcal{Q}_u} (y \log f_{(\theta_p^u, \theta_e, \theta_g)}(\mathbf{x}) + (1 - y) \log(1 - f_{(\theta_p^u, \theta_e, \theta_g)}(\mathbf{x}))), \quad (4-8)$$

其中 $|\mathcal{Q}_u| = L_U - L_{\text{usplit}}$. 在完成本轮所有任务上的局部更新与计算后, 模型求得所有任务查询集上基于局部更新参数计算的损失函数之和 $\sum_{j=1}^N \mathcal{L}(\theta_p^{u_j}, \theta_e, \theta_g)$ 及其对元学习器参数 θ_e 与 θ_g 的梯度 $\nabla_{\theta_e} \sum_{j=1}^N \mathcal{L}(\theta_p^{u_j}, \theta_e, \theta_g)$ 与 $\nabla_{\theta_g} \sum_{j=1}^N \mathcal{L}(\theta_p^{u_j}, \theta_e, \theta_g)$, 然后使用梯度下降法对参数 θ_e 与 θ_g 进行全局更新得到

$$\begin{aligned} \theta_e &= \theta_e - \beta \nabla_{\theta_e} \sum_{j=1}^N \mathcal{L}(\theta_p^{u_j}, \theta_e, \theta_g), \\ \theta_g &= \theta_g - \beta \nabla_{\theta_g} \sum_{j=1}^N \mathcal{L}(\theta_p^{u_j}, \theta_e, \theta_g), \end{aligned} \quad (4-9)$$

其中 β 为全局更新的学习率。上述训练过程的算法步骤已总结至算法 4.2。

在测试阶段, 我们使用经过算法 4.2 训练得到的参数, 并且不再更新元学习器的参数 θ_e 与 θ_g , 仅利用其生成伪用户交互序列特征与其他特征的嵌入向量表示。对于每一个新用户 $u \notin \mathcal{U}^{\text{old}}$, 我们首先在不更新参数的情况下直接对查询集样本 $(\mathbf{x}, y) \in \mathcal{Q}_u$ 进行预测得到 $\hat{p}_{\text{cold}} = f_{(\theta_p, \theta_e, \theta_g)}(\mathbf{x})$, 即不经过探索直接对冷启动用户偏好进行预测得到的概率。然后在支持集样本 $(\mathbf{x}, y) \in \mathcal{S}_u$ 上根据式 (4-6) 计算得到损失函数的值 $\mathcal{L}(\theta_p, \theta_e, \theta_g)$ 及其对参数 θ_p 的梯度 $\nabla_{\theta_p} \mathcal{L}(\theta_p, \theta_e, \theta_g)$, 并根据式 (4-7) 对 θ_p 进行局部更新得到 θ_p^u . 此后, 我们使用局部更新后的参数对查询集样本 (\mathbf{x}, y) 再次预测, 得到 $\hat{p}_{\text{explore}} = f_{(\theta_p^u, \theta_e, \theta_g)}(\mathbf{x})$. 注意到, \hat{p}_{cold} 的准确性反映的

Algorithm 4.2 基于元学习的冷启动推荐模型训练算法

输入: 预训练数据集 $\mathcal{D}_U^{\text{old}}$, 老用户任务集合 $\mathcal{T}_U^{\text{old}}$, $\mathcal{T}_U^{\text{old}}$ 上的学习任务分布 $\rho(\mathcal{T}_u)$, 学习率超参数 α, β .

输出: 冷启动推荐模型的参数 $\theta_p, \theta_e, \theta_g$.

随机初始化 $\theta_p, \theta_e, \theta_g$.

基于 $\mathcal{D}_U^{\text{old}}$ 中的数据对 $f_{(\theta_p, \theta_e)}$ 进行预训练, 更新参数 θ_p 与 θ_e .

while 模型参数未收敛 **do**

从 $\rho(\mathcal{T}_u)$ 中采样一批用户学习任务 $\mathcal{T}_{u_1}, \mathcal{T}_{u_2}, \dots, \mathcal{T}_{u_N}$.

for $j = 1 \rightarrow N$ **do**

基于 \mathcal{S}_{u_j} 中的数据, 根据式 (4-6) 计算损失函数值 $\mathcal{L}(\theta_p, \theta_e, \theta_g)$.

计算梯度 $\nabla_{\theta_p} \mathcal{L}(\theta_p, \theta_e, \theta_g)$.

根据式 (4-7) 进行局部更新得到参数 $\theta_p^{u_j}$.

基于 \mathcal{Q}_{u_j} 中的数据, 根据式 (4-8) 计算损失函数值 $\mathcal{L}(\theta_p^{u_j}, \theta_e, \theta_g)$.

end for

计算梯度 $\nabla_{\theta_e} \sum_{j=1}^N \mathcal{L}(\theta_p^{u_j}, \theta_e, \theta_g)$ 与 $\nabla_{\theta_g} \sum_{j=1}^N \mathcal{L}(\theta_p^{u_j}, \theta_e, \theta_g)$.

根据式 (4-9) 更新参数 θ_e 与 θ_g .

end while

是对于全新的用户, 模型在冷启动阶段的初始化质量; \hat{p}_{explore} 的准确性反映的是经过一段时间的探索之后, 模型调整更新的有效程度。在下文的实验中, 我们将分别考虑 \hat{p}_{cold} 与 \hat{p}_{explore} 的质量以验证模型在解决用户冷启动问题上的有效性。

我们将用户相关数据与学习任务相对应替换为物品相关数据与学习任务, 即可得到针对物品冷启动问题的推荐算法, 故在此不再重复叙述。

4.4 实验与分析

在本节, 我们通过在三个公共数据集上进行实验以验证本章所提出的冷启动框架针对用户冷启动问题的有效性。首先, 我们介绍实验设置的细节, 然后呈现实验结果并对其进行分析。实验显示基于本章所提出的冷启动框架, 当前最先进的推荐排序模型针对新用户的推荐性能均得到了显著提升。

4.4.1 实验设置

我们将 3.4.1 节中介绍的对比方法 FM, YoutubeNet, DeepFM 和 DIN 作为基学习器, 将本章所提出的冷启动探索机制应用于其中进行实验, 以验证我们所提出的方法的有效性。评价指标使用 3.4.1 节中介绍的 AUC 评分指标。接下来, 我们对实验使用的数据集及其划分方式、实现细节与参数设定进行具体介

绍。

1. 数据集

在实验中，我们使用了三个广泛使用的基准数据集进行实验以验证本章所提出模型的有效性。

- **Movielens**^①. Movielens 数据集^[99] 包含了 IMDb^②网站中的电影评分数据。我们的实验在子数据集 MovieLens 1M 上进行。我们选择用户的 ID、年龄、性别、职业、邮编作为用户属性信息，选择电影的 ID、类别、发行年份作为物品属性信息，将用户打过的电影作为用户历史交互序列，将给电影打分的用户作为物品历史交互序列。此外，我们将总评分超过 3 分的样本划分为正例，评分不超过 3 分的样本划分为负例。
- **Book Crossing**^③. Book Crossing 数据集^[100] 包含了 Book-Crossing 社区中的图书评分数据。我们选择用户的 ID、年龄、城市、国家作为用户属性信息，选择书籍的 ISBN 号、出版年份、作者、出版商作为物品属性信息，将用户打过的书籍作为用户历史交互序列，将给书籍打分的用户作为物品历史交互序列。此外，我们将总评分超过 5 分的样本划分为正例，评分不超过 5 分的样本划分为负例。
- **Yelp**^④. Yelp 数据集包含了 Yelp 网站中的餐馆评分数据。我们选择用户 ID 作为用户属性信息，选择餐馆的 ID、城市、类型作为物品属性信息，将用户打过的餐馆作为用户历史交互序列，将给餐馆打分的用户作为物品历史交互序列。此外，我们将总评分超过 3 分的样本划分为正例，评分不超过 3 分的样本划分为负例。

上述数据集的统计信息见表 4-1。其中用户密度与物品密度分别指数据集中用户的平均出现次数与物品的平均出现次数。

为了验证本章所提出算法在用户冷启动任务上的性能，我们将数据集按用户进行分组，并根据每个用户拥有的交互数据样本数来划分新老用户。具体地，我们将交互数据样本数不小于 $L_U = 100$ 的用户划分为老用户，交互数据样本数在 $L_{usplit} = 20$ 到 L_U 之间的用户划分为新用户，其余用户不作考虑。设置新用户

^①<https://grouplens.org/datasets/movielens/1m/>

^②<https://www.imdb.com/>

^③<http://www2.informatik.uni-freiburg.de/cziegler/BX/>

^④<https://www.kaggle.com/yelp-dataset/yelp-dataset>

表 4-1: 数据集统计信息

数据集	样本数	用户数	物品数	用户密度	物品密度
Movielens	1,000,209	6,040	3,883	165.6	257.6
Book Crossing	1,149,780	278,858	271,379	4.1	4.2
Yelp	6,685,900	1,637,138	192,609	4.1	34.7

至少拥有 L_{usplit} 条交互数据样本是为了保证新用户学习任务包含足够的支持集样本与非空的查询集，从而能够根据查询集上的结果检验冷启动推荐模型的性能。对于每个用户的交互数据样本，我们按时间先后进行排序。我们将每个老用户的前 L_{usplit} 条样本划分为老用户学习任务的支持集，第 $L_{\text{usplit}} + 1$ 条至第 L_U 条数据划分为老用户学习任务的查询集；所有老用户第 L_U 条之后的样本构成预训练数据集 D_U^{old} 。我们将每个新用户的前 L_{usplit} 条样本划分为新用户学习任务的支持集，第 L_{usplit} 条之后的样本划分为新用户学习任务的查询集。上文提到的三个数据集的划分情况见表 4-2。

表 4-2: 数据集划分情况

数据集	预训练数据数	老用户数	新用户数	新用户查询集数据总数
Movielens	552,802	2,945	3,009	91,007
Book Crossing	397,572	1,574	4,534	103,476
Yelp	431,637	4,591	40251	736,066

注意到，用户支持集数据总数均为 用户数 $\times 20$ ，老用户查询集数据总数为老用户数 $\times 80$ ，可由新老用户的数目间接反映，故不再列于表格中。

2. 实现细节与参数设定

与第 3.4 章实验类似，我们的任务是在每个数据集上预测标签正负，所有模型均通过深度学习框架 Pytorch^[91] 进行实现，并使用 Adam 优化器^[92] 对模型进行优化。实验中，各个基学习器的结构与 3.4.1 节设置相同。各个模型的嵌入向量维度设置为 24，预训练批量大小设置为 1024，元学习批量大小设置为 16。预训练学习率设置范围为 $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$ ，元学习训练过程中局部更新的学习率设置范围为 $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ ，全局更新的学习率设置范围为 $\{10^{-5}, 10^{-4}, 10^{-3}\}$ ，基学习器在新用户支持集上训练的学习率设置范围为 $\{5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}\}$ 。各类学习率均按衰减率每 1 到 2 个轮次

进行衰减，衰减率设置为 0.1。预训练共进行 4 个轮次，元学习训练过程共进行 2 个轮次，元学习测试过程中在支持集上局部更新共进行 2 个轮次，基学习器在新用户支持集上训练共进行 2 个轮次。为了缓解过拟合问题，我们使用了 L_2 正则化及 Dropout 技术^[70]，正则化系数设置范围为 $\{10^{-4}, 10^{-3}, 10^{-2}\}$ ，Dropout 中神经元丢弃概率设置范围为 $\{0.0, 0.1, 0.2, 0.5\}$ ，并且 Dropout 操作仅应用于多层感知机层的输入数据。

所有实验均在不同随机种子下运行 5 次，下文汇报的结果为 5 次实验的平均结果。

4.4.2 对比实验

本节通过对比当前最先进的推荐排序模型在引入 MePIGen 与不引入 MePIGen 的情况下，针对冷启动用户的推荐性能，验证 MePIGen 模型的通用性与有效性，并详细分析取得性能提升的原因。表 4-3 给出了对比实验在新用户查询集上的 AUC 结果。其中，阶段一表示不在支持集上训练直接预测，阶段二表示经过支持集上训练并对参数进行更新后再进行预测，预测结果分别对应于 \hat{p}_{cold} 与 \hat{p}_{explore} 。从中我们可以观察到，在三个数据集上，引入 MePIGen 的模型在阶段一与阶段二的推荐性能均明显优于不引入 MePIGen 的模型，从而验证了 MePIGen 使用基于元学习的算法以及生成高质量伪交互序列的能力对于用户冷启动问题的有效性。

在同一数据集上，使用不同基学习器的模型性能表现各不相同，这反映的是作为基学习器的推荐排序模型自身的推荐性能差异，但 MePIGen 不局限于特定的基学习器，能够有效提升不同性能的基学习器的用户冷启动推荐性能，反映了 MePIGen 具有一定的通用性。对于我们实验中使用的每一种基学习器，尽管在阶段一 MePIGen 已经能够提升冷启动阶段的初始推荐性能，但在阶段二无论是否使用 MePIGen，模型的预测性能均明显优于阶段一的预测性能，并且使用 MePIGen 能够进一步提升基学习器的性能。因此，解决冷启动问题时，考虑适当的探索过程，允许模型根据新用户的少量交互数据进行调整，是非常有必要的，并且使用 MePIGen 更有利于模型在探索阶段提升性能。

从不同数据集上的结果可以观察到，预训练数据量较大的 Movielens 数据集上阶段一的预测性能明显优于其他两个数据集，这是由于预训练阶段较充

表 4-3: 三个数据集上整体对比实验的 AUC 结果

模型	Movielens		Book Crossing		Yelp	
	阶段一	阶段二	阶段一	阶段二	阶段一	阶段二
FM	0.6846	0.6966	0.5626	0.7086	0.6161	0.6623
MePIGen + FM	0.6928	0.7011	0.5676	0.7080	0.6246	0.6764
YoutubeNet	0.6944	0.7186	0.5700	0.7282	0.6328	0.7150
MePIGen + YoutubeNet	0.6971	0.7215	0.5702	0.7285	0.6413	0.7192
DeepFM	0.7032	0.7306	0.5706	0.7341	0.6388	0.7198
MePIGen + DeepFM	0.7046	0.7341	0.5724	0.7337	0.6506	0.7219
DIN	0.7053	0.7367	0.5739	0.7283	0.6399	0.7192
MePIGen + DIN	0.7073	0.7380	0.5748	0.7295	0.6500	0.7214

分的训练能够为基学习器学得相对可信的模型参数，从而基学习器本身具有更优的泛化性能，而元学习的学习效果也能得到更准确的反映；老用户数较小的 Book Crossing 数据集上 MePIGen 在阶段一对基学习器的性能提升幅度明显小于其他两个数据集，这可能是由于用于元学习训练的老用户学习任务数据量较小进而 MePIGen 得到的训练还不够充分。但无论在哪种情况下，MePIGen 均可以实现对基学习器推荐性能的提升。

4.4.3 消融实验

本节我们分别通过三组实验验证元学习算法的有效性、Transformer 层数对冷启动推荐性能的影响以及支持集大小对冷启动推荐性能的影响。下面将依次展示这三组实验的结果并对其进行分析。

1. 元学习算法的有效性

为了验证元学习算法的有效性，我们基于 MePIGen 的模型架构，分别使用监督学习与元学习方式对模型进行训练与预测。其中，监督学习训练过程同样分两个阶段进行，在第一个阶段使用老用户交互数据样本进行训练，在第二个阶段使用新用户支持集样本进行训练，两阶段训练之后分别进行预测，从而能够与元学习两阶段的预测结果进行公平的对比。Movielens 数据集上的实验结果如表 4-4 所示，其中 AUC 提升指标是指相对于不使用 MePIGen 的基学习器的

AUC 评分提升的百分数。

表 4-4: Movielens 数据集上使用不同学习算法对模型性能的影响

基学习器	学习算法	阶段一		阶段二	
		AUC	AUC 提升	AUC	AUC 提升
FM	监督学习	0.6816	-0.44%	0.6968	0.03%
	元学习	0.6928	1.20%	0.7011	0.65%
YoutubeNet	监督学习	0.6962	0.49%	0.7248	3.38%
	元学习	0.6971	0.62%	0.7215	2.91%
DeepFM	监督学习	0.7045	1.45%	0.7321	1.88%
	元学习	0.7046	1.47%	0.7341	2.16%
DIN	监督学习	0.7083	1.61%	0.7351	1.88%
	元学习	0.7073	1.46%	0.7380	2.29%

从表 4-4 中我们可以观察到, 对于不同的基学习器, 使用元学习训练得到的模型性能在大部分情况下均优于使用监督学习训练得到的模型, 并且使用元学习训练的模型在阶段二能够取得更大幅度的性能提升。注意到, 基学习器使用 FM 时, 监督学习训练得到的模型在阶段一的推荐性能弱于不使用 MePIGen 的基学习器, 在阶段二对基学习器冷启动推荐性能的提升也仅有 0.03%, 由此可见引入元学习算法对模型进行训练的必要性。此外, 使用监督学习在老用户交互数据上训练的过程可以视作预训练过程, 在新用户交互数据上训练的过程可以视作微调过程。由于预训练更关注当下的性能, 元学习更关注未来的性能, 因此在阶段一, 使用两种学习算法训练得到的模型性能差异相对较小, 甚至在基学习器为 DIN 的情况下, 使用监督学习训练得到的模型拥有更高的 AUC 评分。然而, 经过在新用户支持集数据上的训练, 使用元学习训练的模型均取得更优的性能, 这验证了元学习能够帮助模型更快地适应于针对新用户的预测任务。

2. Transformer 层数对冷启动推荐性能的影响

为了研究 Transformer 层数对冷启动性能的影响, 我们基于不同层数的 Transformer 进行实验。Movielens 数据集上的实验结果如图 4-5 所示。其中层数为 0 表示不使用 Transformer 结构, 仅使用一个全连接层构成的前馈神经网络作为伪交互序列生成器。

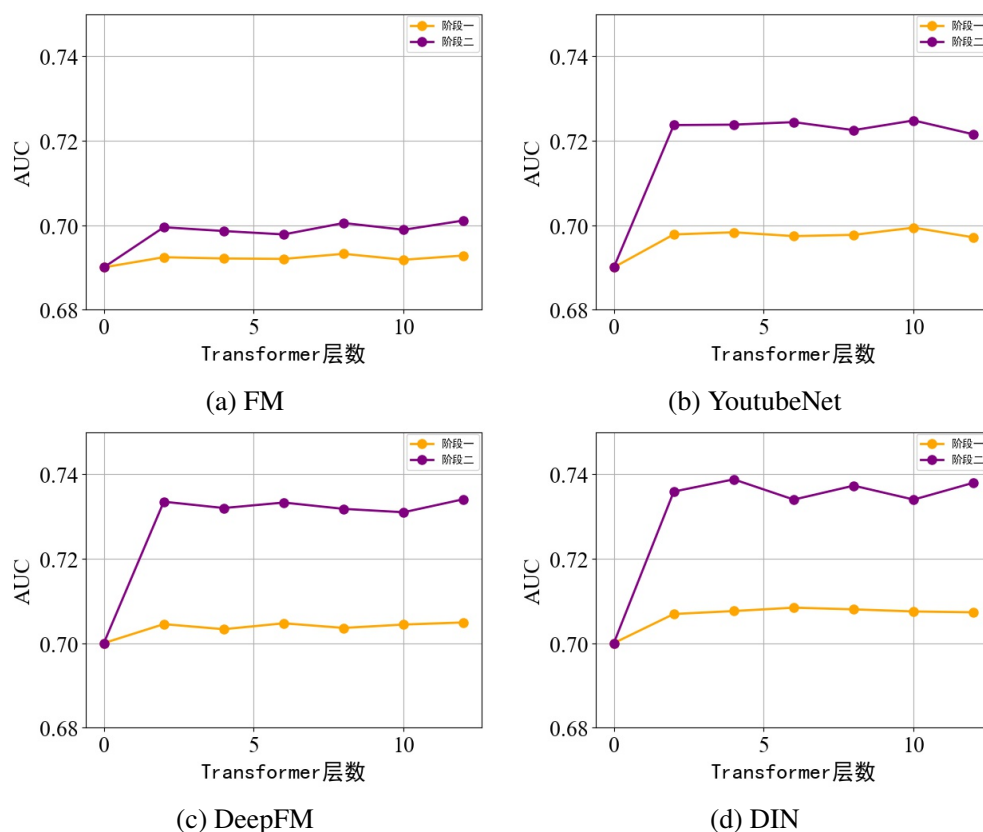


图 4-5: Movielens 数据集上不同层数的 Transformer 对冷启动推荐性能的影响 (以 AUC 度量)

从图 4-5 中我们可以观察到, 不使用 Transformer 结构的模型性能明显弱于使用 Transformer 结构的模型, 这意味着仅依赖于全连接层的生成模型不足以生成高质量的伪交互序列特征, 因此我们基于 Transformer 结构设计 MePIGen 是非常有必要的。使用不同层数 Transformer 结构的模型在阶段二的性能差异明显大于阶段一的性能差异, 这反映了不同模型对新任务的适应速度不同, 选择合适层数的模型能够加速模型对新用户学习任务的适应。此外, 增加层数并不会使得模型性能提升更加明显, 可能的原因是层数的增加使得模型变得更加复杂, 模型存在更大的过拟合风险, 限制了模型泛化性能的提升, 因此在实践中我们使用较少层数的模型即可。

3. 支持集大小对冷启动推荐性能的影响

为了验证支持集大小对冷启动推荐性能的影响, 我们重新划分用户学习任务的支持集与查询集, 将用户的前 L_{support} 条交互数据作为支持集, 第 80 到 100 条交互数据作为查询集, 并在 L_{support} 的不同取值下进行实验。实验结果如图 4-6 所示。

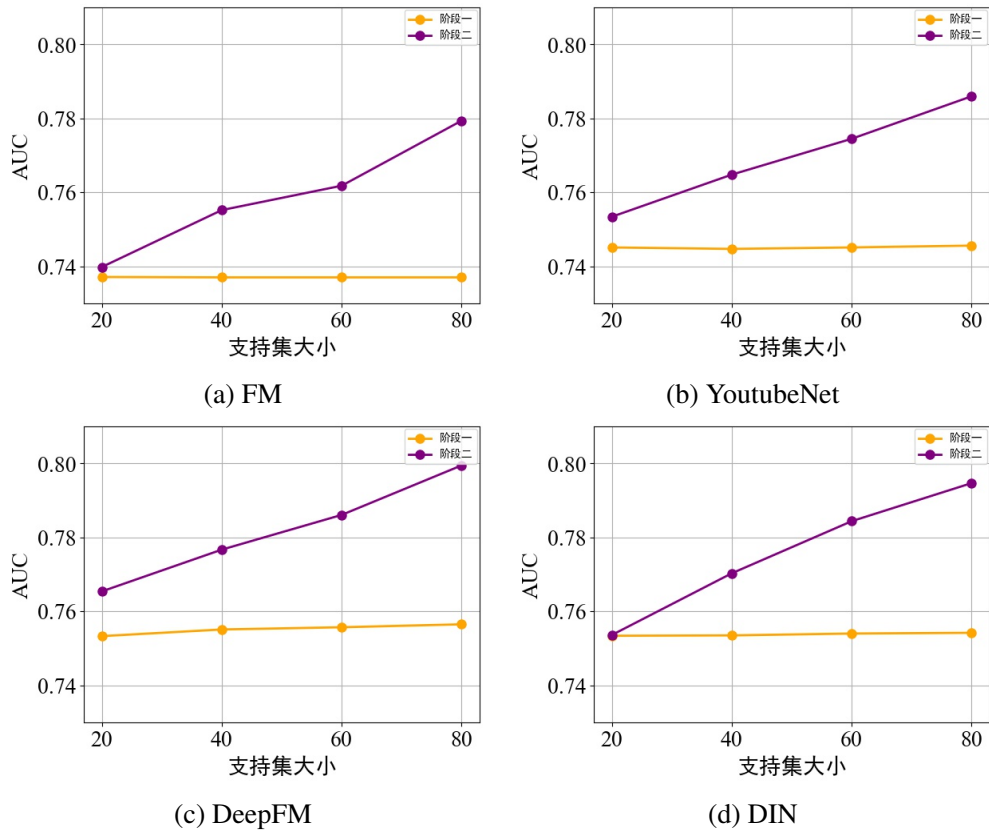


图 4-6: Movielens 数据集上支持集大小对冷启动推荐性能的影响 (以 AUC 度量)

从图 4-6 中可以看到, 随着支持集规模的增大, 模型在两阶段均取得更高的 AUC 评分, 这反映了模型在支持集上调整更新以适应新用户学习任务的过程。此外, 在支持集规模较小的情况下, 阶段二相对于阶段一的模型性能提升增长速度快于在支持集规模较大的情况, 这反映了元学习算法在适应新用户学习任务上的优越性, 也即能够在接触少量新任务样本的情况下对模型进行局部调整, 使得模型达到相对可观的性能。

4.4.4 综合讨论与分析

综合分析表 4-3、表 4-4、图 4-5 与图 4-6, 我们可以得到以下结论: (1) 从模型结构的角度出发, 仅使用全连接层构成的前馈神经网络作为 MePIGen, 无法有效提升冷启动推荐的性能, 因此我们基于 Transformer 结构设计 MePIGen 具有必要性, 这样能够更好地捕获到序列特征并生成高质量的序列特征, 为伪交互序列特征有效提升基学习器性能奠定基础。(2) 从学习方法的角度出发, 同样使用 MePIGen 生成伪交互序列特征替换真实的历史交互特征, 通过监督学习方

式训练得到的模型不足以保证在两阶段均提升冷启动推荐的性能，而通过元学习训练得到的模型可以更有效地提升性能，并且可以加速探索阶段的收敛过程。因此我们基于元学习设计算法是有必要的。

4.5 本章小结

本章我们提出了一个基于元学习与模拟交互序列生成的冷启动探索机制，基于元学习框架对伪历史记录生成器 MePIGen 进行训练，达到在尽可能少的探索次数下训练更新，为新用户或新物品生成高质量的伪历史交互序列的目的，帮助常规推荐模型提升针对新用户与新物品的用户偏好预测性能。为了提升序列的质量，我们基于 Transformer 结构设计 MePIGen 模型使其能够根据前序真实交互序列生成高质量的后续交互序列。基于元学习的框架保障了模型能够较快适应于针对新用户与新物品的推荐任务，在少量探索之后便可达到优良的推荐性能。三个数据集上的实验显示，基于上述思考设计的模型 MePIGen 在面向冷启动用户与冷启动物品的用户偏好预测任务上能够显著提升常规推荐排序模型的性能。

第五章 应用：图书推荐系统

为了验证本文所提出算法在实际推荐场景中的有效性，我们搭建了一个图书推荐系统并将本文所提出方法应用其中。下文将对图书推荐系统的背景、功能、整体架构、实现细节及推荐效果等进行详细介绍。

5.1 相关背景

在信息爆炸的时代背景下，推荐系统因具备帮助人们从海量数据中个性化地筛选信息的能力，逐渐在人们的生活中占据着越来越重要的地位，在电子商务、广告系统、社交媒体等领域都得到了广泛的应用。其中，以图书为推荐对象的图书推荐系统已在电子商务和社交媒体领域积累了一定的应用基础，无论是在线图书交易还是电子图书阅读场景中，均可见到图书推荐系统的身影。此外，在近几年兴起的互联网教育场景中，图书推荐系统也扮演着重要的角色。从教学内容的角度出发，传统课堂教学中教学内容及培养方案的设置往往缺乏灵活性，不能很好地做到因材施教，因此学生需要主动寻找适合自己的学习资料以进行更加个性化的学习。但实际情况中，学生寻找学习资料的过程中可能会遇到受限于信息检索的水平与途径而无法获取到所需资源，以及对教学内容缺乏充分的理解从而找不到针对性的书籍资料等问题。图书推荐系统通过针对性地为学生推荐适合自己的书籍资料，可以实现更加智能、便捷、个性化地教学。

一个性能优秀的推荐系统，无论对于是信息的需求者，还是对于信息的提供者而言，均具有重要意义。在实践中，推荐系统所依赖的数据往往都不够规范，因而理论研究中有效的模型，大部分都难以达到其理论上的性能效果。例如，推荐系统中普遍存在物品的长尾问题与冷启动问题，进而依赖于用户与物品历史交互行为的模型以及不支持辅助信息的模型性能将大打折扣。因此，为了进一步提升推荐系统的性能，需要结合实际应用场景从数据与推荐算法两方面着手，提取高质量的特征以充分反映用户的兴趣偏好与物品的吸引人群，并衡量二者是否匹配。

5.2 系统设计

我们结合在线教育背景，搭建了一个依托于智慧教育平台的图书推荐系统，并将本文第3章与第4章所提出推荐算法应用其中，实现从冷启动阶段到常规阶段的全流程推荐，形成一个完整的体系。

5.2.1 系统需求

图书推荐系统的主要目的是，基于学生的个人信息、图书的基本信息、以及学生的历史行为，帮助学生找到更适合自己的书籍资料，进而实现教学的个性化与智慧化。为了实现上述目的，我们设计的图书推荐系统存在以下需求：

- (1) 构建学生个人画像：基于学生的基本资料以及在系统中的行为模式构建用户画像，为推荐的个性化奠定基础。
- (2) 图书资料收集：基于图书的基础资料构建图书的基本特征，构建丰富的图书数据库作为推荐的素材库。
- (3) 良好的历史交互行为留存机制：保留最近的一部分学生与图书之间的历史交互行为记录，为刻画学生兴趣与图书吸引力特征打下良好的基础。
- (4) 精准而个性化的图书推荐：综合考虑学生的基本信息与系统中的历史行为，与图书数据库中的图书作比对，基于先进的推荐算法进行推荐，以达到精准而个性化的推荐效果。
- (5) 良好的冷启动推荐机制：系统中普遍存在新学生用户与新图书，为了保障系统的良好运作，需要提供良好的冷启动推荐机制，针对缺少历史交互信息的新学生用户与新图书进行推荐。

5.2.2 系统架构

基于5.2.1节所述的系统需求，我们设计了如图5-1所示的系统架构。我们的系统整体分为以下三部分：(1) 数据存储模块：负责数据的存储，主要由一个存储用户信息、图书信息、系统行为日志记录等数据的数据库构成。(2) 后端处理模块：负责数据的使用于管理，以及系统功能逻辑的实现等，具体包括数据的存取、数据的预处理与特征提取、推荐算法的实现、推荐结果的排序整合

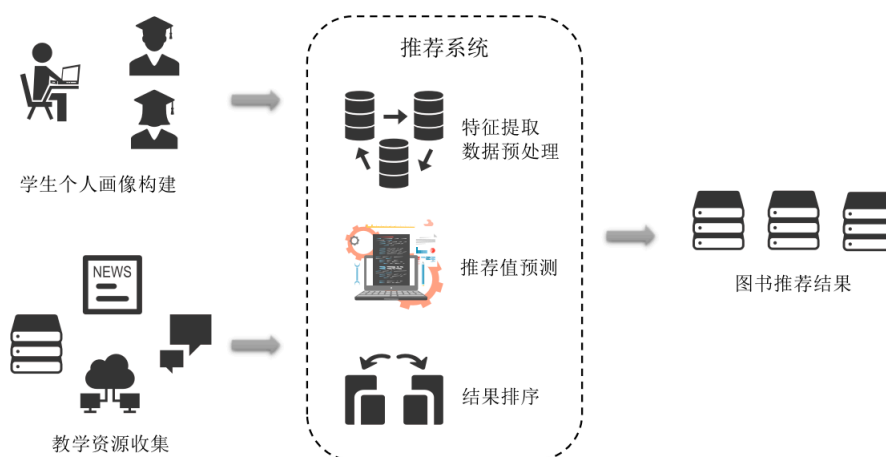


图 5-1: 图书推荐系统架构

等。(3) 前端交互模块：负责与用户进行交互，对用户点击等行为进行响应，并将后端运行结果展示给用户。三部分协同工作，维持着图书推荐系统的正常运作。其中数据存储模块与后端处理模块之间通过关系型数据库语言实现数据的存取，前端交互模块与后端处理模块之间通过调用相应接口函数进行协作。

5.3 系统实现

基于 5.2 节设计的系统架构，我们基于开源 Web 框架 Django 和前端框架 Bootstrap 对图书推荐系统进行具体实现，后端使用 python 语言进行实现，前端使用 javascript 语言进行实现。本系统已部署于装有英伟达公司所生产的 1080-TI 显卡的服务器上，能够通过校内 ip 地址进行访问与操作。下文将从数据采集与存取、数据预处理与特征提取、图书推荐几个方面进行介绍。

5.3.1 数据采集与存取

图书推荐功能涉及到用户信息、图书信息、历史行为几类数据的采集，采集方式如下：

用户信息数据通过学生在系统中填写的个人资料进行获取，包括学生的姓名、学号、年级、院系、籍贯、专业、兴趣爱好、职业规划等信息，其中学号作为用户标识符。

图书信息数据：通过教师自主上传参考资料与外部导入电商平台图书商品信息相结合的方式获取，包括图书的标题、封面图、ISBN 号、作者、出版

商、出版年份等信息，其中 ISBN 号作为图书标识符。

历史行为数据：通过学生在平台的操作日志进行获取，包括用户标识符、图书标识符、行为类型、时间等信息。

系统基于关系型数据库系统 MySQL 对采集到的数据进行存储与管理，后端处理模块通过使用 SQL 语言实现对数据进行存取与管理。

5.3.2 数据预处理与特征提取

由于数据的不规范性，我们首先需要对原始数据进行预处理。具体地，对于数据中的缺失值，我们统一用字符串“NULL”进行标记，对于数据中的不合法字符，我们也用“NULL”来进行替换；对于数据中相同信息的不同表示形式，例如籍贯信息中“广东省”与“广东”含义相同，我们通过分析数据并设置规则的方式对其进行合并。

数据经过预处理后，我们通过特征工程来提取基本特征。具体地，数据中的类别特征，如院系、专业、年级等，通过构建字典转换为数值型编码，进一步通过 One-hot 编码进行表示；数据中的文本特征，如图书名称、用户评论等，通过预训练模型 BERT^[101] 进行文本特征提取将原始文本转换为 768 维数值特征向量表示。

5.3.3 图书推荐

在得到初步提取的特征之后，我们基于第 3 章与第 4 章所提出的算法，实现图书推荐功能。与工业级推荐系统通常将推荐任务划分为召回和排序两阶段不同，我们实现的图书推荐系统省略了召回阶段，直接基于排序阶段进行实现，也即将数据库中的全体图书作为候选图书，而不再从中筛选出候选集。这是由于我们所搭建的图书推荐系统体量较小，即便是数据库中的图书全集，数据量也远小于工业界的推荐系统。因此我们可以简化上述流程，直接进入排序阶段进行推荐值预估与结果排序功能的实现。在推荐值预估过程，系统首先通过推荐排序算法分别预测学生用户对图书数据库中每本图书的偏好概率；在此基础上，对用户已经浏览过的图书对应的偏好概率乘以 0.8 作为最终的概率，以保证推荐内容在一定程度上的新颖程度；然后按照预测概率由大到小排序，选取

排在最前边的 10 本书推荐给用户。

推荐排序算法基于深度学习框架 PyTorch^[91] 进行开发。对于已经积累了一定历史交互的老用户与老图书，我们使用第 3 章所述的算法进行用户偏好预测；对于缺少历史交互的新用户与新图书，我们使用第 4 章所述的算法为用户或图书生成伪交互序列特征，然后再使用第 3 章所述的算法进行推荐。

5.4 效果展示

本节对我们搭建的智慧教务平台的基础页面以及图书推荐系统的推荐效果进行展示。

5.4.1 智慧教育平台

用户访问智慧教育平台之后会进入主页，登录后能够实现访问平台资源、完成基础教学活动、互动讨论、搜索查找、获取个性化推荐内容等功能。图 5-2(a)，图 5-2(c)，图 5-2(b) 与图 5-2(d) 分别展示了智慧教育平台的主页、用户页面、图书浏览页面与图书信息页面的前端效果。

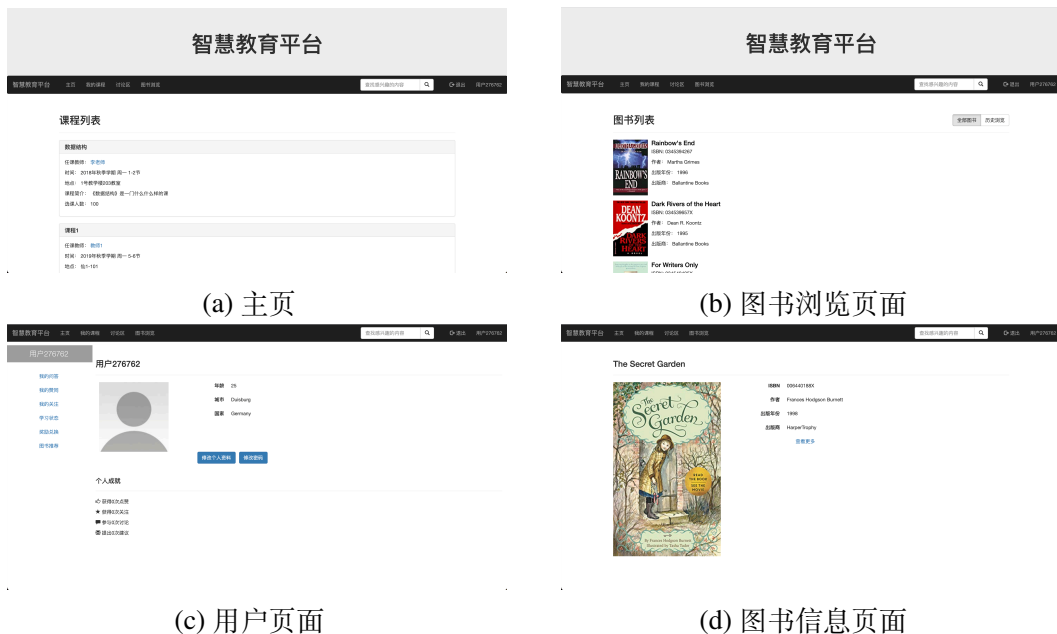


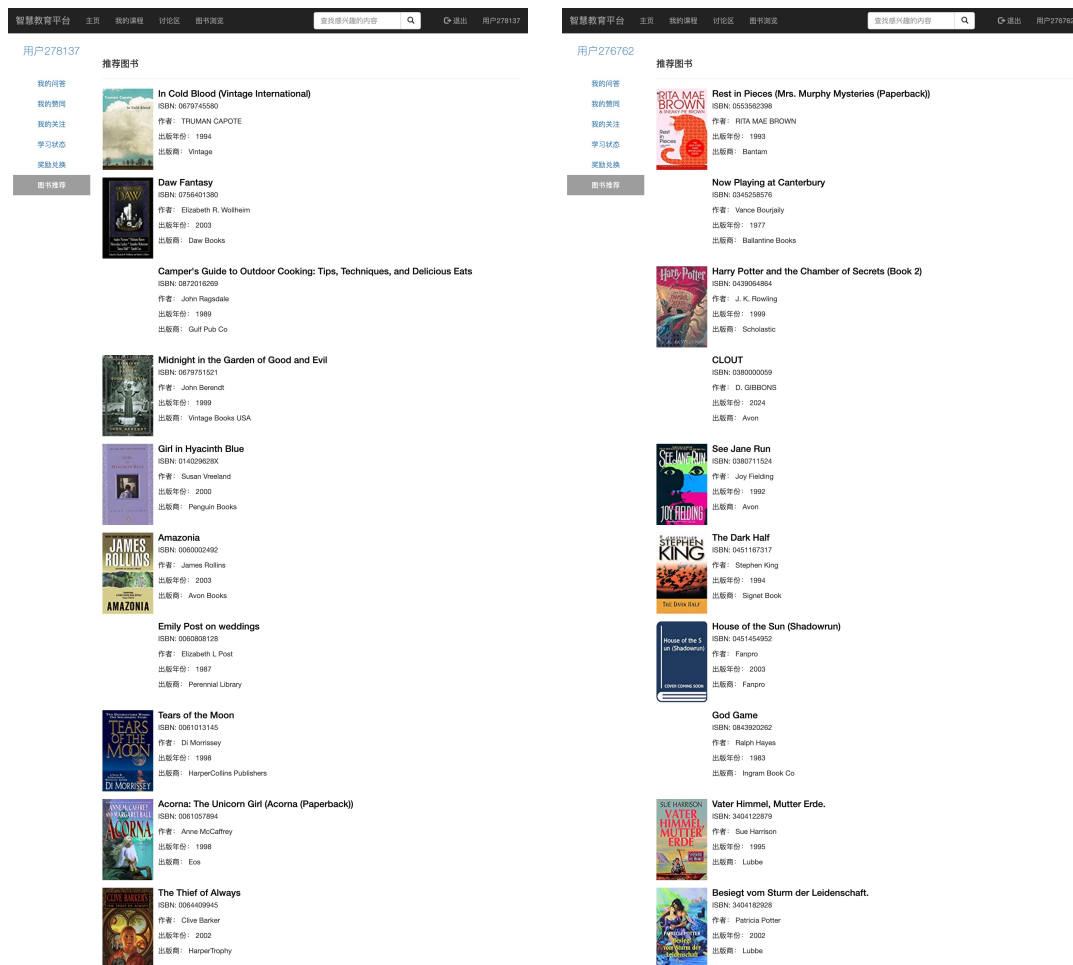
图 5-2: 智慧教育平台基础页面

在如图 5-2(a) 所示的主页中，我们能够看到当前开设的课程列表，并通过导航栏访问自己修读的课程列表、如图 5-2(b) 所示的图书浏览页面以及讨论区。

此外，导航栏的右上角会显示用户登录状态，未登录时点击按钮能够弹出登录对话框，登录后会显示用户名，点击用户名能够进入如图 5-2(c) 所示的用户页面。用户页面会显示用户的基本信息，通过左侧导航栏能够查看自己参与、点赞或关注的讨论与资源，以及获取个性化推荐信息。图书浏览页面会展示平台收录的图书基本信息列表，点击书名可以跳转到如图 5-2(d) 所示的图书信息页面。图书信息页面中展示了更具体的图书信息，并可点击“查看更多”链接访问亚马逊电商平台对应页面了解更多相关信息。

5.4.2 图书推荐系统

通过个人信息页左侧导航栏访问图书推荐模块，我们可以查看为用户个性化推荐的图书资源。我们基于 4.4.1 节中介绍的 Book Crossing 数据集进行推荐，图 5-3(a) 与图 5-3(b) 分别展示了针对老用户与新用户的推荐效果。



(a) 老用户推荐效果

(b) 新用户推荐效果

图 5-3: 图书推荐效果

对于每个用户,我们分别预测该用户对每一本图书的偏好概率,然后根据概率进行排序,选取概率最大的10本书作为推荐结果,并按顺序展示于推荐页面。对于老用户278137,我们使用第3章所提出算法为其进行推荐,得到如图5-3(a)所示的结果。结合老用户278137的历史浏览记录,我们发现推荐列表中有5本用户浏览过的图书及5本用户未浏览过的图书,在保证推荐效果的同时,能够增加用户接触未浏览图书的可能性。对于新用户276762,我们使用第4章所提出算法为其生成伪交互序列,然后再使用第3章所提出算法为其进行推荐,得到如图5-3(b)所示的结果。结合用户276762的初始20条浏览记录,我们能够发现推荐结果在经过一定时间探索后达到令人满意的效果,其中推荐列表中有2本用户浏览过的图书及8本用户未浏览过的图书。上述推荐效果进一步验证了我们所提出方法的有效性。

5.5 本章小结

本章介绍了我们搭建的图书推荐系统。该系统的主要功能是为学生用户个性化推荐图书资料,以实现智能化和个性化的在线教育功能。我们将本文所提出的两个推荐模型应用于该系统中,实现从用户冷启动阶段到常规阶段的全流程推荐功能,构成一个完整的推荐体系。实践表明,我们所提出的算法能提供良好的推荐性能,充分反映了本文所提出算法的实际应用价值。

第六章 总结与展望

本文围绕推荐排序算法进行研究，以历史交互序列信息为着手点，研究如何提升基于历史交互序列信息的推荐排序模型性能。在研究中，我们充分考虑到推荐系统中普遍存在的长尾问题，设计模型使得历史交互序列能够被有效利用，从而在历史交互信息不充分的情况下保证推荐模型仍然具有良好的推荐性能。对于更加困难的冷启动问题，我们通过为新用户或新物品生成替代历史交互序列信息的特征，并在少量探索后对模型参数进行调整更新，为使用历史交互序列信息的推荐模型提供了良好的冷启动性能保障。本文的具体研究内容与贡献如下：

- 本文从历史交互信息的有效利用角度出发，将用户与物品的交互建模为一个动态二部图，在此基础上提出一个多向交互图注意力网络推荐模型，简称 **MIGAT**。**MIGAT** 模型基于注意力机制分别考察历史交互序列中的每一项交互对其交互对象、推荐对象以及推荐上下文的重要度，同时考虑到交互发生时间的早晚对交互置信度的影响，设计置信度嵌入向量结构以区分不同交互的置信度。综合多个维度下的重要度衡量指标，**MIGAT** 模型基于图神经网络与前馈神经网络结构生成表达能力更丰富的用户特征表示、物品特征表示以及自适应的交互序列特征表示。经实验验证，该模型在多个数据集上表现优于当前先进的推荐排序模型，在长尾问题比较严重的情况下性能提升更加明显。
- 本文针对新用户或新物品缺少历史交互信息的问题，提出一个基于元学习的伪交互序列生成器模型，简称 **MePIGen**。该模型基于元学习算法对参数进行学习，使得模型经过少量几次对新用户与新物品的探索之后就能生成高质量的伪交互序列特征，伪交互序列特征将作为真实交互序列的替代特征用于推荐排序模型中，以提升推荐排序模型的预测性能。**MePIGen** 的结构基于 **Transformer** 而设计，将现有的历史交互序列与属性特征作为输入，得到输出的伪交互序列，保证生成伪交互序列的质量与效率。经实验验证，该模型在多个数据集上使得当前先进的推荐排序模型的冷启动推荐性能得到提升。

- 本文所提出的两个推荐模型应用于实际搭建的图书推荐系统中，获得良好的推荐性能。该图书推荐系统中，从针对新用户的冷启动阶段，到已经积累一定交互数据样本进入常规阶段，构成一个完整的推荐体系，充分展示了模型的实际应用价值。

在本文的研究基础上，可以从多个角度入手展开新的研究工作。对于第3章所提出的模型，可以考虑基于多层图卷积网络结构设计模型，发掘更复杂的数据实体间的关系，与更多数据实体之间建立联系，从而进一步缓解物品长尾问题所带来的影响。对于第4章所提出的伪交互序列生成器，可以借鉴自然语言处理领域的预训练语言模型中普遍使用的掩码机制进行训练，增强模型根据前序元素预测后序元素的能力，进一步提升伪历史交互序列的质量；此外，可以考虑从损失函数入手，结合伪历史交互序列与真实交互序列之间的相似度设置优化目标。

从长尾问题和冷启动问题出发，还可以考虑引入知识图谱相关技术，结合更丰富的信息及更先进的图算法来刻画用户的物品的偏好特征，从而达到更精准的推荐效果。

参考文献

- [1] FINN C, ABBEEL P, LEVINE S. Model-agnostic meta-learning for fast adaptation of deep networks[C]//Proceedings of Machine Learning Research: volume 70 Proceedings of the 34th International Conference on Machine Learning. : PMLR, 2017: 1126-1135.
- [2] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems 30. 2017: 5998-6008.
- [3] COVINGTON P, ADAMS J, SARGIN E. Deep neural networks for youtube recommendations[C]//Proceedings of the 10th ACM Conference on Recommender Systems. : ACM, 2016: 191-198.
- [4] SARWAR B M, KARYPIS G, KONSTAN J A, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th International World Wide Web Conference. : ACM, 2001: 285-295.
- [5] KOREN Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]//Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008. : ACM, 2008: 426-434.
- [6] LINDEN G, SMITH B, YORK J. Amazon.com recommendations: Item-to-item collaborative filtering[J]. IEEE Internet Comput., 2003, 7(1):76-80.
- [7] HE X, LIAO L, ZHANG H, et al. Neural collaborative filtering[C]//Proceedings of the 26th International Conference on World Wide Web. : ACM, 2017: 173-182.
- [8] 孙光福, 吴乐, 刘淇, 等. 基于时序行为的协同过滤推荐算法[J]. 软件学报, 2013, 11.
- [9] 冷亚军, 陆青, 梁昌勇. 协同过滤推荐技术综述[J]. 模式识别与人工智能, 2014, 27(8):720-734.
- [10] 邓爱林, 朱扬勇, 施伯乐. 基于项目评分预测的协同过滤推荐算法[J]. 软件学报, 2003, 14(9).

- [11] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]//Advances in Neural Information Processing Systems 25. 2012: 1106-1114.
- [12] CHAN W, JAITLEY N, LE Q V, et al. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition[C]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing. : IEEE, 2016: 4960-4964.
- [13] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[C]//The 3rd International Conference on Learning Representations. 2015.
- [14] 黄立威, 江碧涛, 吕守业, 等. 基于深度学习的推荐系统研究综述[J]. 计算机学报, 2018, 41(7):1619-1647.
- [15] 张敏, 丁弼原, 马为之, 等. 基于深度学习加强的混合推荐方法[J]. 2017.
- [16] MNIH V, HEES N, GRAVES A, et al. Recurrent models of visual attention [C]//Advances in Neural Information Processing Systems 27. 2014: 2204-2212.
- [17] ZHOU G, ZHU X, SONG C, et al. Deep interest network for click-through rate prediction[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. : ACM, 2018: 1059-1068.
- [18] ZHOU C, BAI J, SONG J, et al. Atrank: An attention-based user behavior modeling framework for recommendation[C]//Proceedings of the 32nd AAAI Conference on Artificial Intelligence. : AAAI Press, 2018: 4564-4571.
- [19] ZHOU G, MOU N, FAN Y, et al. Deep interest evolution network for click-through rate prediction[C]//Proceedings of the 33rd AAAI Conference on Artificial Intelligence. : AAAI Press, 2019: 5941-5948.
- [20] FENG Y, LV F, SHEN W, et al. Deep session interest network for click-through rate prediction[C]//Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2019: 2301-2307.
- [21] XU W, HE H, TAN M, et al. Deep interest with hierarchical attention network for click-through rate prediction[C]//Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. : ACM, 2020: 1905-1908.

- [22] ANDERSON C. The long tail: Why the future of business is selling less of more [M]. : Hachette Books, 2006.
- [23] YIN H, CUI B, LI J, et al. Challenging the long tail recommendation[J]. Proc. VLDB Endow., 2012, 5(9):896-907.
- [24] RENDLE S. Factorization machines[C]//Proceedings of the 10th International Conference on Data Mining. : IEEE, 2010: 995-1000.
- [25] CHENG H, KOC L, HARMSEN J, et al. Wide & deep learning for recommender systems[C]//Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. : ACM, 2016: 7-10.
- [26] QU Y, CAI H, REN K, et al. Product-based neural networks for user response prediction[C]//Proceedings of the 16th International Conference on Data Mining. : IEEE, 2016: 1149-1154.
- [27] ZHANG Y, DAI H, XU C, et al. Sequential click prediction for sponsored search with recurrent neural networks[C]//Proceedings of the 28th AAAI Conference on Artificial Intelligence. : AAAI Press, 2014: 1369-1375.
- [28] SHAN Y, HOENS T R, JIAO J, et al. Deep crossing: Web-scale modeling without manually crafted combinatorial features[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. : ACM, 2016: 255-262.
- [29] MCMAHAN H B, HOLT G, SCULLEY D, et al. Ad click prediction: a view from the trenches[C]//Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. : ACM, 2013: 1222-1230.
- [30] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition [C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. : IEEE Computer Society, 2016: 770-778.
- [31] HE X, CHUA T. Neural factorization machines for sparse predictive analytics [C]//Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017. : ACM, 2017: 355-364.
- [32] XIAO J, YE H, HE X, et al. Attentional factorization machines: Learning the weight of feature interactions via attention networks[C]//Proceedings of the

- Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. : ijcai.org, 2017: 3119-3125.
- [33] LIAN J, ZHOU X, ZHANG F, et al. xdeepfm: Combining explicit and implicit feature interactions for recommender systems[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. : ACM, 2018: 1754-1763.
- [34] YANG Y, XU B, SHEN S, et al. Operation-aware neural networks for user response prediction[J]. *Neural Networks*, 2020, 121:161-168.
- [35] CHUNG J, GÜLCEHRE cC, CHO K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J/OL]. *CoRR*, 2014, abs/1412.3555. <http://arxiv.org/abs/1412.3555>.
- [36] ZHANG M, TANG J, ZHANG X, et al. Addressing cold start in recommender systems: a semi-supervised co-training algorithm[C]//Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval. : ACM, 2014: 73-82.
- [37] SAVESKI M, MANTRACH A. Item cold-start recommendations: learning local collective embeddings[C]//Proceedings of the 8th ACM Conference on Recommender Systems. : ACM, 2014: 89-96.
- [38] LI X, SHE J. Collaborative variational autoencoder for recommender systems [C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. : ACM, 2017: 305-314.
- [39] LIN J, SUGIYAMA K, KAN M, et al. Addressing cold-start in app recommendation: latent user models constructed from twitter followers[C]//Proceedings of the 36th International ACM SIGIR conference on research and development in Information Retrieval. : ACM, 2013: 283-292.
- [40] ZHAO W X, LI S, HE Y, et al. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information[J]. *IEEE Trans. Knowl. Data Eng.*, 2016, 28(5):1147-1159.
- [41] SHI C, LI Y, ZHANG J, et al. A survey of heterogeneous information network analysis[J]. *IEEE Trans. Knowl. Data Eng.*, 2017, 29(1):17-37.
- [42] ZHANG Y, AI Q, CHEN X, et al. Joint representation learning for top-n recommendation with heterogeneous information sources[C]//Proceedings of the 2017

- ACM on Conference on Information and Knowledge Management. : ACM, 2017: 1449-1458.
- [43] LU Y, FANG Y, SHI C. Meta-learning on heterogeneous information networks for cold-start recommendation[C]//Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. : ACM, 2020: 1563-1573.
- [44] HWANG W, PARC J, KIM S, et al. "told you i didn't like it": Exploiting uninteresting items for effective collaborative filtering[C]//Proceedings of the 32nd IEEE International Conference on Data Engineering. : IEEE Computer Society, 2016: 349-360.
- [45] CHAE D, KANG J, KIM S, et al. Rating augmentation with generative adversarial networks towards accurate collaborative filtering[C]//Proceedings of the 28th International Conference on World Wide Web. : ACM, 2019: 2616-2622.
- [46] WANG Q, YIN H, WANG H, et al. Enhancing collaborative filtering with generative augmentation[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. : ACM, 2019: 548-556.
- [47] YOON J, JORDON J, VAN DER SCHAAR M. GAIN: missing data imputation using generative adversarial nets[C]//Proceedings of the 35th International Conference on Machine Learning: volume 80. : PMLR, 2018: 5675-5684.
- [48] YU J, GAO M, YIN H, et al. Generating reliable friends via adversarial training to improve social recommendation[C]//Proceedings of 19th IEEE International Conference on Data Mining. : IEEE, 2019: 768-777.
- [49] SUN C, LIU H, LIU M, et al. LARA: attribute-to-feature adversarial learning for new-item recommendation[C]//Proceedings of the 13th ACM International Conference on Web Search and Data Mining. : ACM, 2020.
- [50] CHAE D, KIM J, CHAU D H, et al. AR-CF: augmenting virtual users and items in collaborative filtering for addressing cold-start problems[C]//Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. : ACM, 2020: 1251-1260.
- [51] LI L, CHU W, LANGFORD J, et al. A contextual-bandit approach to personalized news article recommendation[C]//Proceedings of the 19th International Conference on World Wide Web. : ACM, 2010: 661-670.

- [52] TANG L, JIANG Y, LI L, et al. Personalized recommendation via parameter-free contextual bandits[C]//Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. : ACM, 2015: 323-332.
- [53] SHAH P, YANG M, ALLE S, et al. A practical exploration system for search advertising[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. : ACM, 2017: 1625-1631.
- [54] PAN F, CAI Q, TANG P, et al. Policy gradients for contextual recommendations [C]//Proceedings of the 28th International Conference on World Wide Web. : ACM, 2019: 1421-1431.
- [55] LEE H, IM J, JANG S, et al. Melu: Meta-learned user preference estimator for cold-start recommendation[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. : ACM, 2019: 1073-1082.
- [56] PAN F, LI S, AO X, et al. Warm up cold-start advertisements: Improving CTR predictions via learning to learn ID embeddings[C]//Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. : ACM, 2019: 695-704.
- [57] DONG M, YUAN F, YAO L, et al. MAMO: Memory-augmented meta-optimization for cold-start recommendation[C]//Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. : ACM, 2020: 688-697.
- [58] RICHARDSON M, DOMINOWSKA E, RAGNO R. Predicting clicks: estimating the click-through rate for new ads[C]//Proceedings of the 16th International Conference on World Wide Web. : ACM, 2007: 521-530.
- [59] HARSHMAN R A, et al. Foundations of the parafac procedure: Models and conditions for an " explanatory" multimodal factor analysis[J]. 1970:1-84.
- [60] JUAN Y, ZHUANG Y, CHIN W, et al. Field-aware factorization machines for CTR prediction[C]//Proceedings of the 10th ACM Conference on Recommender Systems. : ACM, 2016: 43-50.
- [61] ZHANG W, DU T, WANG J. Deep learning over multi-field categorical data - - A case study on user response prediction[C]//Lecture Notes in Computer Science:

- volume 9626 Proceedings of the 38th European Conference on IR Research. : Springer, 2016: 45-57.
- [62] GUO H, TANG R, YE Y, et al. Deepfm: A factorization-machine based neural network for CTR prediction[C]//Proceedings of the 26th International Joint Conference on Artificial Intelligence. : ijcai.org, 2017: 1725-1731.
- [63] LECUN Y, BOSER B E, DENKER J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural Comput., 1989, 1(4):541-551.
- [64] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Comput., 1997, 9(8):1735-1780.
- [65] LIU Q, YU F, WU S, et al. A convolutional click prediction model[C]// Proceedings of the 24th ACM International Conference on Information and Knowledge Management. : ACM, 2015: 1743-1746.
- [66] HARIRI N, MOBASHER B, BURKE R D. Context-aware music recommendation based on latent topic sequential patterns[C]//Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012. : ACM, 2012: 131-138.
- [67] WU S, REN W, YU C, et al. Personal recommendation using deep recurrent neural networks in netease[C]//Proceedings of the 32nd IEEE International Conference on Data Engineering. : IEEE Computer Society, 2016: 1218-1229.
- [68] GEHRING J, AULI M, GRANGIER D, et al. A convolutional encoder model for neural machine translation[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. : Association for Computational Linguistics, 2017: 123-135.
- [69] VOLKOV M, YU G W, POUTANEN T. Dropoutnet: Addressing cold start in recommender systems[C]//Advances in Neural Information Processing Systems 30. 2017: 4957-4966.
- [70] SRIVASTAVA N, HINTON G E, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. J. Mach. Learn. Res., 2014, 15(1):1929-1958.
- [71] ROY S, GUNTUKU S C. Latent factor representations for cold-start video recommendation[C]//Proceedings of the 10th ACM Conference on Recommender Systems. : ACM, 2016: 99-106.

- [72] CHOU S, YANG Y, JANG J R, et al. Addressing cold start for next-song recommendation[C]//Proceedings of the 10th ACM Conference on Recommender Systems. : ACM, 2016: 115-118.
- [73] WANG X, HE X, CAO Y, et al. KGAT: knowledge graph attention network for recommendation[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. : ACM, 2019: 950-958.
- [74] VILALTA R, DRISSI Y. A perspective view and survey of meta-learning[J]. *Artif. Intell. Rev.*, 2002, 18(2):77-95.
- [75] VARTAK M, THIAGARAJAN A, MIRANDA C, et al. A meta-learning perspective on cold-start recommendations for items[C]//Advances in Neural Information Processing Systems 30. 2017: 6904-6914.
- [76] BRITZ D, GOLDIE A, LUONG M, et al. Massive exploration of neural machine translation architectures[J/OL]. *CoRR*, 2017, abs/1703.03906. <http://arxiv.org/abs/1703.03906>.
- [77] SCARSELLI F, GORI M, TSOI A C, et al. The graph neural network model[J]. *IEEE Trans. Neural Networks*, 2009, 20(1):61-80.
- [78] 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.: volume 2. 2005: 729-734 vol. 2.
- [79] VAN DEN BERG R, KIPF T N, WELLING M. Graph convolutional matrix completion[J/OL]. *CoRR*, 2017, abs/1706.02263. <http://arxiv.org/abs/1706.02263>.
- [80] BRUNA J, ZAREMBA W, SZLAM A, et al. Spectral networks and locally connected networks on graphs[C/OL]//2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings. 2014. <http://arxiv.org/abs/1312.6203>.
- [81] DEFFERRARD M, BRESSON X, VANDERGHEYNST P. Convolutional neural networks on graphs with fast localized spectral filtering[C]//Advances in Neural Information Processing Systems 29. 2016: 3837-3845.
- [82] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks[C]//The 5th International Conference on Learning Representations. 2017.

- [83] MONTI F, BRONSTEIN M M, BRESSON X. Geometric matrix completion with recurrent multi-graph neural networks[C]//Advances in Neural Information Processing Systems 30. 2017: 3697-3707.
- [84] 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. : ACM, 2019: 165-174.
- [85] ZHANG J, SHI X, ZHAO S, et al. STAR-GCN: stacked and reconstructed graph convolutional networks for recommender systems[C]//Proceedings of the 28th International Joint Conference on Artificial Intelligence. : ijcai.org, 2019: 4264-4270.
- [86] GEHRING J, AULI M, GRANGIER D, et al. Convolutional sequence to sequence learning[C]//Proceedings of the 34th International Conference on Machine Learning. : PMLR, 2017: 1243-1252.
- [87] MAAS A L, HANNUN A Y, NG A Y. Rectifier nonlinearities improve neural network acoustic models[C]//Proceedings of the 30th International Conference on Machine Learning: volume 30. : Citeseer, 2013: 3.
- [88] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning representations by back-propagating errors[J]. nature, 1986, 323(6088):533-536.
- [89] HE R, MCAULEY J J. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering[C]//Proceedings of the 25th International Conference on World Wide Web. : ACM, 2016: 507-517.
- [90] MCAULEY J J, TARGETT C, SHI Q, et al. Image-based recommendations on styles and substitutes[C]//Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. : ACM, 2015: 43-52.
- [91] PASZKE A, GROSS S, MASSA F, et al. Pytorch: An imperative style, high-performance deep learning library[C]//Advances in Neural Information Processing Systems 32. 2019: 8024-8035.
- [92] KINGMA D P, BA J. Adam: A method for stochastic optimization[C]//The 3rd International Conference on Learning Representations. 2015.
- [93] FAWCETT T. An introduction to ROC analysis[J]. Pattern Recognition Letters, 2006, 27(8):861-874.

- [94] ZHU Y, LIN J, HE S, et al. Addressing the item cold-start problem by attribute-driven active learning[J]. *IEEE Trans. Knowl. Data Eng.*, 2020, 32(4):631-644.
- [95] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C]//*Advances in Neural Information Processing Systems* 27. 2014: 3104-3112.
- [96] CHO K, VAN MERRIENBOER B, GÜLCEHRE cC, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[C]//*Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. : ACL, 2014: 1724-1734.
- [97] NAIR V, HINTON G E. Rectified linear units improve restricted boltzmann machines[C]//*Proceedings of the 27th International Conference on Machine Learning*. : Omnipress, 2010: 807-814.
- [98] BA L J, KIROUS J R, HINTON G E. Layer normalization[J/OL]. *CoRR*, 2016, abs/1607.06450. <http://arxiv.org/abs/1607.06450>.
- [99] HARPER F M, KONSTAN J A. The movielens datasets: History and context [J]. *ACM Trans. Interact. Intell. Syst.*, 2015, 5(4):19:1-19:19.
- [100] ZIEGLER C, MCNEE S M, KONSTAN J A, et al. Improving recommendation lists through topic diversification[C]//*Proceedings of the 14th international conference on World Wide Web*. : ACM, 2005: 22-32.
- [101] DEVLIN J, CHANG M, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding[J/OL]. *CoRR*, 2018, abs/1810.04805. <http://arxiv.org/abs/1810.04805>.

附录 A 符号与函数说明

A.1 基础符号说明

本文使用形如 \mathbf{w} 的粗体小写字母来表示向量，使用形如 \mathbf{W} 的粗体大写字母来表示矩阵，使用形如 \mathcal{A} 的花体大写字母来表示集合，使用 \parallel 表示向量的拼接操作，使用 $\cdot^{(l)}$ 表示序列中第 l 个元素的权重或嵌入向量。

A.2 基础函数说明

下文将对本文所使用的基础函数进行说明。

1. **指示函数** $\mathbb{I}(\cdot)$. 对于任意条件变量 x , $\mathbb{I}(x)$ 的定义如下:

$$\mathbb{I}(x) = \begin{cases} 1 & \text{if } x \text{ is True,} \\ 0 & \text{if } x \text{ is False.} \end{cases} \quad (\text{A-1})$$

2. **Sigmoid 函数** $\text{Sigmoid}(\cdot)$. 对于任意输入 $x \in \mathbb{R}$, $\text{Sigmoid}(x)$ 的定义如下:

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}. \quad (\text{A-2})$$

对于任意向量形式的输入 $\mathbf{x} = [x_1, x_2, \dots, x_m] \in \mathbb{R}^m$, **Sigmoid** 函数的结果是 \mathbf{x} 每一维分别计算得到的结果构成的向量，即对于任意 $1 \leq i \leq m$, 均有

$$\text{Sigmoid}(\mathbf{x})_i = \text{Sigmoid}(x_i). \quad (\text{A-3})$$

3. **Softmax 函数** $\text{Softmax}(\cdot)$. 对于任意输入 $\mathbf{x} = [x_1, x_2, \dots, x_m] \in \mathbb{R}^m$, **Softmax**

函数对 \mathbf{x} 的每一维进行指数归一化得到结果向量，即对于任意 $1 \leq i \leq m$, 均有

$$\text{Softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^m e^{x_j}}. \quad (\text{A-4})$$

致 谢

三年的研究生时光转瞬即逝，回首往事，我在南京大学度过的时光历历在目。临别之际，对于陪伴我七年的校园，以及这些年校园时光中遇到的人与事，我的心中充满感慨，相信这将成为我人生历程中难以忘却的宝贵财富。此刻，我由衷地感谢在研究生生涯中给予我关心、支持和帮助的师长与同伴。

首先，我要感谢我的导师申富饶老师。申老师治学严谨，对科研有着独到的见解，鼓励我们从问题出发进行独立思考，做对实际生活有价值的研究工作。同时，申老师对于科研方向的选择给予我们极大的自由度，支持我们在各自感兴趣的方向开展研究。申老师的科研理念与对待科研的严谨态度对我产生了极大的影响，使我能够逐渐掌握独立进行科研的能力，并沿着自己感兴趣的方向进行研究，最终形成本文为三年科研经历做一个圆满总结。此外，申老师坚持每周单独与组内每一位同学进行面对面的讨论交流，并组织讨论班分享彼此的研究内容。申老师的辛苦付出，帮助我们在学习与科研中更快地找到不足，并不断取得进步。

其次，我要感谢南京大学的其他老师。赵健老师多次为我们分享论文写作的经验，并逐字逐句地帮我们检查修改待投稿的英文论文，对于我们英文论文写作给予了很大帮助。同时，赵老师与我们共同参与讨论班，为我们的研究工作提出许多有价值的意见与建议。此外，南京大学计算机系的老师为我们开设了丰富的专业课程，为我们进行深入的科研打下坚实的基础。

接着，我要感谢陪伴我度过研究生生涯的同学。感谢 RINC 研究组的同学，无论是在讨论班，还是私下的交流中，都热心为我的科研与学习提供了许多帮助。同时，友好融洽的实验室关系为我的生活带来了许多欢乐与温暖。感谢我的室友，在生活中给予我非常多关心与帮助。最后，特别感谢李明威同学，在研一带我参加推荐系统比赛，为我后续的科研工作奠定了基础。同时，李明威同学在三年的学习与生活中付出许多时间与精力给予我支持，并激励我不断突破自我，取得进步。

最后，我要感谢我的家人。父母为我提供了最坚强的后盾，良好的生活保障

及坚实的精神支柱使我能够将精力更多地投入到科研与学习之中，在面对困难时不畏惧，在面对压力时不崩溃，顺利完成学业。

简历与科研成果

基本信息

刘雅辉，女，汉族，1995年10月出生，内蒙古鄂尔多斯市人。

教育背景

2018年9月 – 2021年6月 南京大学计算机科学与技术系 硕士

2014年9月 – 2018年6月 南京大学计算机科学与技术系 本科

攻读硕士学位期间完成的学术成果

1. Yahui Liu, Furao Shen, Jian Zhao. "Pairwise Interactive Graph Attention Network for Context-Aware Recommendation." arXiv preprint arXiv:1911.07429 (2019).

已申请国家发明专利

1. 申富饶, 刘雅辉, 赵健, 于德。一种基于图神经网络和注意力机制的上下文推荐方法。专利申请号: 201911035702.2

攻读硕士学位期间参与的科研课题

1. 国家自然科学基金“基于深度感知增量式联想记忆神经网络的信息融合系统研究”(项目编号: 61876076, 课题年限: 2019年1月 – 2022年12月), 负责推荐系统相关问题的研究。

学位论文出版授权书

本人完全同意《中国优秀博硕士学位论文全文数据库出版章程》（以下简称“章程”），愿意将本人的学位论文提交“中国学术期刊（光盘版）电子杂志社”在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版，并同意编入《中国知识资源总库》，在《中国博硕士学位论文评价数据库》中使用和在互联网上传播，同意按“章程”规定享受相关权益。

作者签名：_____

_____年____月____日

论文题名	基于深度学习与历史交互序列建模的推荐排序算法研究				
研究生学号	MG1833050	所在院系	计算机科学与技术系	学位年度	2021
论文级别	<input checked="" type="checkbox"/> 学术学位硕士 <input type="checkbox"/> 专业学位硕士 <input type="checkbox"/> 学术学位博士 <input type="checkbox"/> 专业学位博士 (请在方框内画勾)				
作者 Email	liuyahui@smail.nju.edu.cn				
导师姓名	申富饶 教授				

论文涉密情况：

不保密

保密，保密期：_____年____月____日至_____年____月____日

注：请将该授权书填写后装订在学位论文最后一页（南大封面）。

