

# 高效神经网络推荐理论与方法研究

(申请清华大学工学博士学位论文)

培养单位： 计算机科学与技术系

学 科： 计算机科学与技术

研 究 生： 陈 冲

指导教师： 张 敏 副教授

二〇二二年五月



# **Efficient Neural Recommendation: Theories and Methods**

Dissertation Submitted to

**Tsinghua University**

in partial fulfillment of the requirement

for the degree of

**Doctor of Philosophy**

in

**Computer Science and Technology**

by

**Chen Chong**

Dissertation Supervisor: Associate Professor Zhang Min

**May, 2022**



## 学位论文公开评阅人和答辩委员会名单

### 公开评阅人名单

文继荣	教授	中国人民大学
李勇	副教授	清华大学

### 答辩委员会名单

主席	文继荣	教授	中国人民大学
委员	马少平	教授	清华大学
	朱军	教授	清华大学
	何向南	教授	中国科学技术大学
	李勇	副教授	清华大学
秘书	谢晓晖	助理研究员	清华大学



## 关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容；（3）根据《中华人民共和国学位条例暂行实施办法》及上级教育主管部门具体要求，向国家图书馆报送相应的学位论文。

本人保证遵守上述规定。

作者签名： 陈冲

导师签名： 张立

日期： 2022.3.21

日期： 2022.3.21



## 摘要

在大数据时代信息过载的背景下，个性化推荐系统被广泛的应用于互联网平台上，通过理解用户行为背后的深层兴趣帮助用户寻找所需的内容和信息，已成为支撑智能互联网的基础技术之一。近年来，深度学习和神经网络促进了推荐系统模型的进一步发展。然而，精准且高效的神经网络推荐系统仍是困扰学术界和产业界的重要问题之一。许多研究倾向于构建复杂的神经网络模型来提升推荐效果而忽视了模型的训练效率。同时，目前广泛采用的负采样训练策略很难使推荐模型学习到最优的状态。推荐系统多应用场景中的不同数据特点及近年来对于数据可遗忘的需求，也对模型表现和高效性带来了更加复杂的挑战。考虑到推荐系统的应用范围之广和影响之大，精准且高效的神经网络推荐研究具有其紧迫性和重要性。本文从高效非采样隐式数据学习、高效多场景推荐建模以及高效可遗忘推荐框架三个方面展开研究，主要贡献及创新点如下：

首先，本文研究如何从海量隐式数据中实现精准且高效的非采样学习。从模型训练的角度出发，本文通过严谨的数学推理，设计了一系列高效的、可用于神经网络训练的非采样学习算法，将从全样本数据中学习的时间复杂度降低了一个数量级。在现实数据集上的实验表明，所提出的算法相对于已有方法，推荐精准度提升 5% 以上，训练速度提升 5-30 倍。这一研究也是基础性的，本质上，从只有正向交互的数据中学习的机器学习任务和算法都可以使用这一解决方案。

第二，本文基于所设计的高效非采样学习算法，针对推荐系统不同应用场景的数据特点，研究如何高效且精细化建模结合异质交互行为（社交网络、多类型交互行为）与内容信息（特征信息、知识图谱）的神经网络推荐系统模型。所提出的模型在多个现实数据集上相比已有基准方法，在模型表现和训练效率上均取得了显著的提升效果。例如，在结合多行为数据的推荐任务上，推荐精准度提升 40% 以上，训练速度快 10 倍以上。

第三，本文在高效非采样推荐模型的基础上更进一步，针对数据可遗忘需求，研究了高效可遗忘推荐系统问题。即在收到数据遗忘的请求时（如用户主动要求删除数据，坏数据去除等），模型能够快速去除这些数据对于学习结果带来的影响。本文设计了一个通用的高效可遗忘推荐系统框架。现实数据集上的实验表明，所提出的方法可以在保证推荐精度的前提下实现高效的数据遗忘。

**关键词：**推荐系统；隐式数据；非采样学习；高效学习；机器遗忘

## Abstract

Under the background of information overload in the era of Big Data, personalized recommender system has been widely deployed in Web applications to help users seek desired information and items by learning users' potential preferences from behavioral data. It has become one of the most basic supportive techniques of web intelligence. In recent years, techniques like deep learning and neural networks have promoted the further development of recommendation models. However, efficient and effective neural recommendation is still an important issue for both the research community and practical application. Many deep learning studies only focus on obtaining better results but ignored the computational efficiency of reaching the reported accuracy. Meanwhile, the performances of existing recommendation methods are limited by the inherent weakness of the widely-used sampling-based learning strategy. The different data characteristics in various recommendation scenarios and the recent demand of erasable machine learning also bring more complex challenges to the effectiveness and efficiency of neural recommendation models. Considering the wide application range and great effect of recommender systems, the research of efficient and effective neural recommendation is urgent and critical. In this thesis, we conduct research in aspects of efficient non-sampling learning, efficient recommendation models for various scenarios, and efficient recommendation unlearning framework. The main contributions and innovations are as follows:

First, this work studies the basic problem of how to realize efficient and effective non-sampling learning from massive implicit data. Through rigorous mathematical analysis, we derive several new optimization methods, which resolves computational bottlenecks in optimization by leveraging the sparsity of implicit data. Experiments on real-world datasets show that the proposed methods outperform the state-of-the-art methods by more than 5% in accuracy and 5-30 times in training efficiency. This research is also a fundamental study of machine learning, which has the potential to benefit many tasks (not limited to recommendation) where only positive data is observed.

Second, based on the proposed efficient non-sampling learning methods, this work studies how to efficiently and finely-grained build neural recommendation models with social network, multi-behavior data, context information, and knowledge graph. The proposed models significantly better than state-of-the-art models in both recommendation

performance and training efficiency. E.g., on multi-behavior recommendation scenario, we achieve more than 40% better performance and 10 times faster training than existing methods.

Third, in terms of the recent demand of erasable machine learning, this work studies the problem of efficient recommendation unlearning. In many cases, a recommender system also needs to forget certain sensitive data and its complete lineage. From the perspective of privacy, several privacy regulations have recently been enacted, requiring systems to eliminate any impact of the data whose owner requests to forget. From the perspective of utility, if a system's utility is damaged by some bad data, the system needs to forget such data to regain utility. This work proposes a general machine unlearning framework tailored to recommendation tasks, which can not only achieve efficient unlearning but also outperform the state-of-the-art machine unlearning methods in terms of recommendation utility.

**Keywords:** Recommender System; Implicit Feedback; Non-Sampling Learning; Efficient Learning; Machine Unlearning

## 目 录

摘 要.....	I
Abstract.....	II
目 录.....	IV
插图和附表清单.....	VII
第 1 章 绪论 .....	1
1.1 选题背景与研究挑战.....	1
1.2 研究内容与贡献.....	5
1.3 本文结构安排.....	6
第 2 章 国内外研究现状 .....	8
2.1 推荐系统中的隐式数据学习.....	8
2.2 推荐系统多数据场景建模.....	9
2.2.1 协同过滤推荐方法简要回顾.....	9
2.2.2 结合异质交互行为的推荐方法.....	10
2.2.3 结合内容信息的推荐方法.....	11
2.3 数据遗忘研究方法.....	11
第 3 章 高效非采样隐式数据学习研究 .....	13
3.1 引言.....	13
3.2 预备知识.....	14
3.2.1 符号定义.....	14
3.2.2 非采样隐式数据学习.....	15
3.3 高效非采样学习.....	16
3.3.1 高效神经网络矩阵分解框架.....	16
3.3.2 基于用户的高效学习算法.....	17
3.3.3 基于物品的高效学习算法.....	19
3.3.4 交替训练的高效学习算法.....	20
3.3.5 讨论.....	21
3.4 实验.....	22
3.4.1 实验设置.....	22

---

3.4.2 性能评估.....	25
3.5 小结.....	30
<b>第 4 章 结合异质交互行为的高效推荐建模 .....</b>	<b>31</b>
4.1 引言.....	31
4.2 相关工作.....	32
4.2.1 结合社交行为的推荐.....	32
4.2.2 结合多类型交互行为的推荐.....	33
4.3 结合社交行为的高效推荐建模.....	34
4.3.1 问题定义.....	34
4.3.2 基于自适应迁移学习的高效社交推荐模型.....	35
4.3.3 实验.....	39
4.4 结合多类型交互行为的高效推荐建模.....	46
4.4.1 问题定义.....	46
4.4.2 基于因果推理的高效多行为推荐模型.....	47
4.4.3 基于异质图神经网络的高效多行为推荐模型.....	50
4.4.4 实验.....	54
4.5 小结.....	63
<b>第 5 章 结合内容信息的高效推荐建模 .....</b>	<b>64</b>
5.1 引言.....	64
5.2 相关工作.....	65
5.2.1 结合特征信息的推荐.....	65
5.2.2 结合知识图谱的推荐.....	66
5.3 结合特征信息的高效推荐建模.....	67
5.3.1 问题定义.....	67
5.3.2 预备知识.....	68
5.3.3 基于非采样因子分解机的高效特征增强推荐模型.....	68
5.3.4 实验.....	73
5.4 结合知识图谱的高效推荐建模.....	80
5.4.1 问题定义.....	80
5.4.2 基于非采样联合学习的高效知识增强推荐模型.....	81
5.4.3 实验.....	86
5.5 小结.....	94

---

第 6 章 高效可遗忘推荐系统框架研究 .....	95
6.1 引言 .....	95
6.2 预备知识 .....	96
6.2.1 问题定义 .....	97
6.3 高效可遗忘推荐框架 .....	97
6.3.1 方法概览 .....	97
6.3.2 数据平衡分割算法 .....	98
6.3.3 基于注意力机制的模型聚合 .....	100
6.3.4 训练 .....	102
6.4 实验 .....	103
6.4.1 实验设置 .....	103
6.4.2 性能评估 .....	105
6.5 小结 .....	109
第 7 章 总结与展望 .....	110
7.1 研究工作总结 .....	110
7.2 未来工作展望 .....	112
参考文献 .....	114
致 谢 .....	125
声 明 .....	126
个人简历、在学期间完成的相关学术成果 .....	127
指导教师学术评语 .....	130
答辩委员会决议书 .....	131

## 插图和附表清单

图 1.1	信息过载与个性化推荐 .....	1
图 1.2	本文研究工作的整体架构 .....	5
图 3.1	高效神经网络矩阵分解框架 (ENMF) 示意图 .....	16
图 3.2	基于用户的高效学习算法输入示意图 .....	17
图 3.3	基于物品的高效学习算法输入示意图 .....	20
图 3.4	GMF, ENMF-Original, ENMF-U, ENMF-I 和 ENMF-A 在不同向量维度下的训练耗时 .....	27
图 3.5	GMF, ConvNCF, ENMF-U, ENMF-I 和 ENMF-A 随训练轮数的收敛情况 .....	28
图 3.6	GMF, ConvNCF, ENMF-U, ENMF-I 和 ENMF-A 随向量维度变化的表现情况 .....	28
图 3.7	WMF, ENMF-U, ENMF-I 和 ENMF-A 随负样本权重变化的表现情况 .....	29
图 3.8	GMF, ConvNCF, ENMF-U, ENMF-I 和 ENMF-A 随网络留存率变化的表现情况 .....	30
图 4.1	用户在社交领域和物品领域的共有偏好案例示意图 .....	35
图 4.2	基于自适应迁移学习的高效社交推荐模型 (EATNN) 示意图 .....	36
图 4.3	EATNN 和 SAMN 随训练轮数的收敛情况 .....	43
图 4.4	EATNN 和其模型变体的表现情况 .....	45
图 4.5	用户多类型交互行为之间因果关系示意图 .....	47
图 4.6	基于因果推理的高效多行为推荐模型 (EHCF) 示意图 .....	48
图 4.7	基于异质图神经网络的高效多行为推荐模型 (GHCF) 示意图 .....	51
图 4.8	ENMF、NCF、NMTR、和 ECHF 模型随训练轮数的收敛情况 .....	58
图 4.9	NCF、ENMF、NMTR、EHCF 和 GHCF 模型在不同购买行为数量下的表现情况 .....	59
图 4.10	GHCF 及其变体的表现效果 .....	61
图 4.11	多任务学习系数对于 EHCF 模型的结果影响 .....	62
图 4.12	多任务学习系数对于 GHCF 模型的结果影响 .....	63
图 5.1	基于非采样因子分解机的高效特征增强推荐模型 (ENSFM) 示意图 .....	69
图 5.2	因子分解机中的特征交互示意图 .....	70
图 5.3	ENSFM 中辅助向量示意图 .....	71
图 5.4	ENSFM, CFM, NFM, DeepFM 和 FM 在不同向量维度下的训练耗时 .....	77

图 5.5	ENSFM, CFM, NFM, DeepFM 和 FM 随训练轮数的收敛情况 .....	78
图 5.6	ENSFM, CFM, NFM, DeepFM 和 FM 随向量维度变化的表现情况 ....	79
图 5.7	ENSFM, ENMF 和 CFM 随负样本权重变化的表现情况 .....	79
图 5.8	基于非采样联合学习的高效知识增强推荐模型 (JNSKR) 示意图 .....	81
图 5.9	JNSKR 与不同推荐模型在不同交互行为数量下的表现情况 .....	91
图 5.10	JNSKR 和它的两个变种的实验表现 .....	92
图 5.11	联合学习超参数 $\mu$ 敏感性分析 .....	93
图 6.1	高效可遗忘推荐框架 (RecEraser) 示意图 .....	98
图 6.2	RecEraser 中不同数据分割方式对推荐结果的影响 .....	107
图 6.3	RecEraser 中不同聚合方式对推荐结果的影响 .....	108
图 6.4	RecEraser 中不同子模型数量对结果的影响 .....	108
表 3.1	高效非采样学习方法中所用数学符号 .....	15
表 3.2	评估 ENMF 方法的数据集的基本统计信息 .....	22
表 3.3	ENMF 与对比方法的特点比较 .....	24
表 3.4	ENMF 与不同推荐方法的效果比较, ** 表示较最好的基准方法有显著提升, 显著性是 $p < 0.01$ 。“RI”表示 ENMF-A 相较对应方法的相对提升 ..	26
表 3.5	ENMF 与不同推荐模型的训练耗时情况, 其中 s/m/h 分别代表秒/分钟/小时	27
表 4.1	结合社交行为的高效推荐建模中所用数学符号 .....	34
表 4.2	评估 EATNN 模型的数据集统计信息 .....	39
表 4.3	EATNN 与对比方法的特点比较 .....	40
表 4.4	EATNN 与不同推荐方法的效果比较, ** 表示较最好的基准方法有显著提升, 显著性是 $p < 0.01$ 。“RI”表示 EATNN 相较对应方法的相对提升 ...	42
表 4.5	EATNN 与不同推荐模型的训练耗时情况, 其中 s/m/h 分别代表秒/分钟/小时 .....	43
表 4.6	Epinion 数据集冷启动条件下的实验结果, 25% 和 50% 代表训练集所占数据集比例 .....	44
表 4.7	自适应迁移学习权重展示 .....	46
表 4.8	结合多类型交互行为的高效推荐建模中所用数学符号 .....	47
表 4.9	贝贝和淘宝数据集的基本统计信息 .....	55
表 4.10	EHCF、GHCF 与基准方法的特点比较 .....	56

表 4.11	EHCF, GHCF 与不同基线推荐方法的效果比较, ** 表示较最好的基准方法有显著提升, 显著性是 $p < 0.01$ .....	57
表 4.12	EHCF 与不同推荐模型的训练耗时情况, 其中 s/m 分别代表秒/分钟....	58
表 4.13	EHCF 及其变体在淘宝数据集上的表现效果 .....	60
表 4.14	图卷积层数对 GHCF 模型结果的影响 .....	62
表 5.1	结合特征信息的高效推荐建模中所用数学符号 .....	67
表 5.2	评估 ENSFM 模型的数据集统计信息.....	73
表 5.3	ENSMF 与对比方法的特点比较 .....	74
表 5.4	ENSMF 与不同推荐方法的效果比较, ** 表示较最好的基准方法有显著提升, 显著性是 $p < 0.01$ 。“RI” 表示 ENSFM 相较对应方法的相对提升 ...	76
表 5.5	ENSMF 与不同推荐模型的训练耗时情况, 其中 s/m/h/d 分别代表秒/分钟/小时/天 .....	78
表 5.6	结合知识图谱的高效推荐建模中所用数学符号 .....	80
表 5.7	评估 JNSKR 模型的数据集统计信息.....	86
表 5.8	JNSKR 与对比方法的特点比较 .....	87
表 5.9	JNSKR 与不同推荐方法的效果比较, ** 表示较最好的基准方法有显著提升, 显著性是 $p < 0.01$ 。“RI” 表示 JNSKR 相较对应方法的相对提升 ....	89
表 5.10	JNSKR 与不同推荐模型的训练耗时情况, 其中 s/m/h 分别代表秒/分钟/小时.....	91
表 5.11	联合学习和交替学习两种训练策略在不同数据集上的结果比较, * 表示有统计显著的结果差异 $p < 0.05$ .....	93
表 6.1	高效可遗忘推荐系统研究中所用数学符号 .....	96
表 6.2	评估 RecEraser 的数据集统计信息 .....	103
表 6.3	RecEraser 与不同机器遗忘方法的效果比较, ** 表示较最好的基准方法有显著提升, 显著性是 $p < 0.01$ .....	105
表 6.4	RecEraser 的数据遗忘效率 .....	106



## 第1章 绪论

### 1.1 选题背景与研究挑战

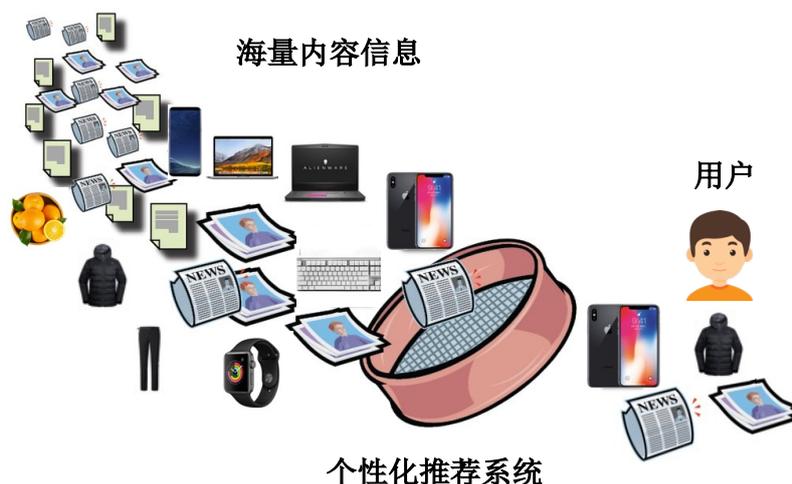


图 1.1 信息过载与个性化推荐

信息获取是人们感知、学习和理解世界的第一步，也是至关重要的一步。随着大数据时代的到来，互联网、通信技术及智能设备飞速发展，各种各样的内容与服务，如长短视频、商品、音乐、新闻等呈现出指数级上升趋势<sup>[1]</sup>。面对海量信息，人类有限的认知能力出现了严重的“信息过载”（Information Overload）问题，越来越难以从中获取自己感兴趣的部分。为了缓解信息过载，个性化推荐系统（Personalized Recommender Systems）<sup>[2-4]</sup>应运而生，通过理解用户历史行为背后的潜在兴趣与需求，自动对海量内容进行过滤，为用户提供精准的个性化服务。当前，推荐系统作为人工智能与机器学习商业化最成功的技术之一，已成为人们日常生活中不可或缺的一部分。从满足用户信息获取需求而快速成长的快手、抖音、今日头条等产品，到参与辅助用户决策过程的电子商务平台亚马逊、淘宝和京东等，推荐系统在方方面面得到了广泛的应用。推荐系统对于用户和商家平台均有重要的价值。首先，从用户的角度，推荐系统优化了信息获取的效率，拓宽了未知发现的视野，提升了用户体验。从商家平台的角度，推荐系统将商家生产的物品匹配给具有相应需求的用户，从而提高系统资源配置的效率，提升经济效益，进而推动国民经济的健康有序发展。

推荐系统的研究不但具有重要的实际应用价值，更有显著的科学研究意义。首先，推荐系统在技术上涉及多个人工智能和机器学习的重要分支，如深度学习（Deep Learning）、数据挖掘（Data Mining）、表示学习（Representation Learning）和自然语

言处理 (Natural Language Processing) 等, 有利于促进相关技术研究的广度和深度。其次, 在多类型信息处理和用户行为理解等不同场景下, 推荐系统涉及到经济学、社会学、传播学、心理学等学科的交叉综合, 有利于促进跨学科学术研究的共同发展。不仅如此, 推荐系统技术的基础性、重要性也有利于推动诸多应用平台的进一步智能化和个性化。

通常而言, 在推荐系统的构建中有三个重要组成部分需要考虑, 分别是“学习方法”, “推荐模型”, 以及“系统平台”。“学习方法”指从收集到的历史数据(如商品购买记录、视频和新闻的点击记录等)训练模型的方式, 其中又包括“目标函数”和“样本选择策略”<sup>[5]</sup>; “推荐模型”即是为具体应用场景所设计的推荐系统模型, 帮助用户寻找所需的信息(如商品、视频、新闻等内容或服务), 这些信息在本文中统一描述为“物品 (Item)”; “系统平台”指的是搭载推荐系统的框架和服务平台, 用于收集用户行为数据、筛选训练特征等。首先, 在“学习方法”上, 推荐系统通常使用的数据是隐式反馈数据 (Implicit Feedback Data), 往往只包含正反馈 (Positive-only Data)<sup>[6-7]</sup>。数据中用户兴趣的负反馈信号是缺失的。例如, 对于新闻平台中的点击行为, 如果用户点击了某个新闻, 那么系统将收集到正反馈(标签为“1”), 说明用户对于该新闻感兴趣。但是对于未点击的新闻, 原因可能是由于用户对该新闻不感兴趣, 或者是根本没有看到它, 此时所反映出的用户兴趣是不确定的<sup>[8-9]</sup>。从只有正反馈的数据中学习是广泛意义上大多数机器学习模型训练过程中的一个基本操作, 对于推荐系统而言同样十分重要。第二, 在“推荐模型”上, 深度学习和神经网络<sup>[10]</sup>促进了推荐系统模型的进一步发展。然而, 精准且高效的神经网络推荐系统仍是困扰学术界和产业界的重要问题之一。许多研究倾向于构建复杂的神经网络模型来提升推荐效果而忽视了模型的训练效率<sup>[11]</sup>。同时, 推荐系统多应用场景中包含的不同数据, 如社交网络<sup>[12]</sup>、多类型交互行为<sup>[13]</sup>、特征信息<sup>[14]</sup>、知识图谱<sup>[15]</sup>等也对模型表现和高效性带来了更加复杂的挑战。第三, 在“系统平台”上, 考虑到所收集的交互数据可能包含着敏感信息或是噪声点, 推荐系统在一些场景也需要快速遗忘某些数据以及从这些数据中学到的模型参数<sup>[16]</sup>。从隐私保护的角度, 最近颁布了一些隐私相关法规, 要求系统能够消除用户所要求遗忘的数据对模型的任何影响<sup>[17]</sup>。从模型表现的角度, 如果推荐系统的精准度被一些噪声数据影响<sup>[18]</sup>, 系统需要遗忘这些数据以提升推荐效果。因此, 高效可遗忘推荐系统框架的研究同样至关重要。

综上所述, 本文以实现精准且高效的个性化推荐系统为目标, 从“学习方法”、“推荐模型”、与“系统平台”三个不同的角度, 对**高效非采样隐式数据学习**、**高效多场景推荐建模**、以及**高效可遗忘推荐框架**三个问题展开研究。研究所面临的关键

科学挑战如下:

1. 推荐系统所常用的隐式数据中负反馈信息缺失, 且其对用户兴趣建模至关重要。如何精准且高效的获取负反馈信号一直是困扰推荐系统学术界和产业界的难题之一。

在隐式数据中, 用户的点击、购买等行为被记录为正反馈, 说明用户对于该物品感兴趣。而对于未记录的缺失数据, 真实的负反馈(用户不感兴趣)和无标签数据(用户没看到)是混合在一起的。一方面, 缺失数据中仅有一部分是用户真正不感兴趣的物品, 需要从一个庞大的数据集中高效的提取出这部分负反馈信息; 另一方面, 除了用户真实不感兴趣的物品之外, 剩余绝大多数物品并没有展示给用户, 对应的用户兴趣是不确定的, 如果出现误判, 则有可能在模型学习过程中引入错误的负样本(False Negative), 也就是缺失数据中未被观测到的正样本, 从而降低推荐模型的鲁棒性和表现效果<sup>[19]</sup>。为了从缺失数据中学习用户兴趣, 有两种常用的学习策略: (1) 负采样策略, 即通过采样的方式从缺失数据中选择出一部分作为负样本<sup>[6,20-21]</sup>; (2) 非采样策略, 又称全采样策略, 即将所有的缺失数据看做是负样本, 并根据其不确定性合理分配样本权重以缓解数据不平衡问题和降低误判风险<sup>[7-9,22]</sup>。这两种策略各有利弊: 负采样的优势是训练数据少, 效率较高, 但其有效性容易受到所采样的负样本质量的影响, 使得模型很难收敛到最优的状态。与之相反, 非采样学习考虑了所有的训练数据, 在模型训练效果上更有优势。但传统的非采样学习方法复杂度高, 计算效率较差。同时, 现有的提升非采样学习计算效率的方法<sup>[9,23]</sup>往往是基于交替最小二乘法(Alternating Least Squares)和EM算法(Expectation-Maximum), 不适用于目前广泛应用的基于梯度下降(Gradient Descent)的深度学习和神经网络模型。鉴于非采样学习在模型训练效果上的优势, 高效、可靠、且通用的非采样学习算法是一个极具研究前景和现实应用价值的课题, 也是机器学习领域一个重要的研究难题。

2. 现实应用场景中所包含的不同数据类型, 使得精准且高效的神经网络推荐系统建模极具挑战性。

在个性化推荐系统中, 收集到的用户行为数据往往是非常稀疏的。幸运的是, 在现实应用场景中广泛存在的额外信息可以作为交互行为数据的补充, 帮助推荐系统学习用户的兴趣偏好<sup>[4,24-25]</sup>。常见的额外信息包括社交网络<sup>[12,26]</sup>、多类型交互行为(如点击、加入购物车等)<sup>[13,27]</sup>、特征信息(如用户年龄、性别以及物品类别、价格等)<sup>[14,28]</sup>和知识图谱<sup>[15,29]</sup>。这些额外信息在不同程度和方面上反映了用户的偏好和物品的属性, 对于精准的推荐系统建模至关重要, 但所涉及到的个性化迁移学习<sup>[30]</sup>和多任务联合学习<sup>[31-32]</sup>的相关技术还未得到很好的研究。同时,

近年来深度学习和神经网络促进了推荐系统模型的进一步研究和发展。然而，当前学术界的研究更多的倾向于引入复杂和先进的神经网络如注意力机制 (Attention Mechanisms)<sup>[33-34]</sup>, 记忆网络 (Memory Networks)<sup>[12,35]</sup>, 卷积神经网络 (Convolutional Neural Networks)<sup>[36-37]</sup>, 循环神经网络 (Recurrent Neural Networks)<sup>[38]</sup>, 生成对抗网络 (Generative Adversarial Networks)<sup>[39-40]</sup>, 图神经网络 (Graph Neural Networks)<sup>[41-42]</sup> 等来提升推荐效果，在模型训练效率方面却关注较少。现有的神经网络推荐系统模型存在着两个重要的局限性：(1) 复杂的神经网络结构往往包含大量的模型参数<sup>[11]</sup>，即使使用基于负采样的学习策略，也需要昂贵的计算资源；(2) 广泛使用的负采样学习策略性能不稳定，其有效性容易受到采样质量的影响，使得模型很难收敛到最优的状态<sup>[43]</sup>。以上这些因素限制了神经网络推荐模型在大规模现实场景下的实用性，也为多场景下精准、高效的神经网络推荐建模带来了很大挑战。

3. 数据可遗忘性在保护用户隐私，增强模型稳定性等方面发挥重要作用，越来越受到人们的重视。现有方法难以在保证精准度的同时实现通用且高效率的可遗忘推荐系统。

推荐系统通过从训练数据中学习来为用户提供个性化的产品和服务。然而，在许多场景下，推荐系统也需要能够遗忘某些敏感数据以及从这些数据中学到的参数<sup>[16]</sup>。首先，最近一些关于数据隐私保护的法案颁布，如海外的 GDPR (General Data Protection Regulation)<sup>①</sup>和 CCPA (California Consumer Privacy Act)<sup>②</sup>赋予数据“被遗忘的权利”，“被遗忘的权利”要求系统能够消除数据所有者要求遗忘的数据及其带来的影响 (如从数据中学习到的用户偏好)。国内的《互联网信息服务算法推荐管理规定》<sup>③</sup>也对数据遗忘做了不同程度的规定。其次，在模型稳定性上，如果模型的表现效果被一些噪声数据所降低时，需要遗忘这些脏数据对于模型参数的影响。最直接的数据遗忘方法是用去除要删除数据后的训练集重新训练模型。然而在大规模现实数据场景下，重训模型需要巨大的计算开销。因此，高效的、可遗忘推荐系统有着重要的实际需求和研究价值。机器遗忘 (Machine Unlearning) 作为人工智能的新生领域，现有的方法仍处于刚起步阶段。目前关于数据遗忘的研究主要集中在计算机视觉和自然语言处理的相关任务上<sup>[44-46]</sup>，还未有在推荐系统上的研究工作。这些现有的机器遗忘方法缺乏对于推荐系统所使用的结构化历史交互数据的考量，直接应用于推荐任务上时会造成较大的精度损失。因此，如何在保证推荐精准度的同时实现通用且高效率的数据遗忘需要进一步的研究和探索。

① <https://gdpr-info.eu/>

② <https://oag.ca.gov/privacy/ccpa>

③ [http://www.cac.gov.cn/2022-01/04/c\\_1642894606364259.htm](http://www.cac.gov.cn/2022-01/04/c_1642894606364259.htm)

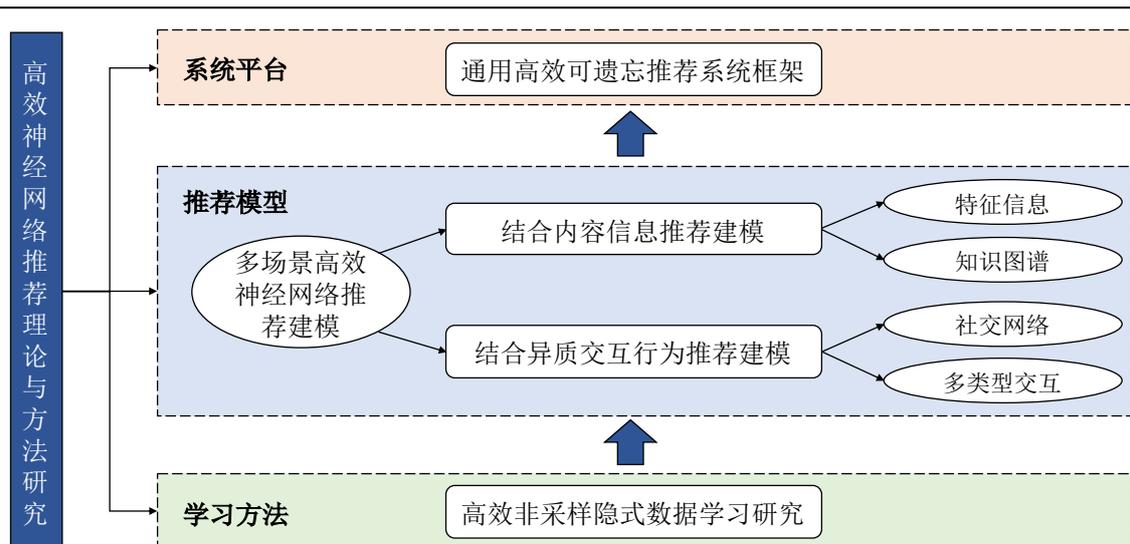


图 1.2 本文研究工作的整体架构

## 1.2 研究内容与贡献

以上研究挑战分别对应推荐系统中“学习方法”、“推荐模型”、和“系统平台”三个层次，且三个层次之间存在着交互依存的关系。本文的研究目标即是针对这些问题和挑战形成立体、综合的解决方案，从而为建立起精准、高效的个性化推荐系统提供理论和技术支撑。在“学习方法”上，本文研究了作为基础的高效非采样隐式数据学习算法；在“推荐模型”上，本文研究了多个应用场景下的精准且高效的神经网络推荐系统建模；在“系统平台”上，本文研究了通用的高效可遗忘推荐系统框架。本文研究工作的整体架构如图1.2所示。下面对这三部分研究工作进行具体介绍。

在推荐系统隐式数据学习方面，针对用户负反馈信息难以获取这一挑战，本文重点研究高效、可靠的非采样学习方法。现有隐式数据学习方法中，负采样策略稳定性差，难以将模型优化到最优的状态，而传统非采样学习方法复杂度较高，不适用于目前广泛使用的神经网络推荐系统模型。本文从学习理论出发，通过严谨的数学推理，将因子分解中互相独立的运算分离开来，新设计了一系列高效的、可用于神经网络训练的非采样学习方法。所提出来的方法将从全样本数据中学习的时间复杂度降低了一个数量级，并同时保留了非采样学习的高精准度优势。同时，这一研究也是基础性的，本质上，从只有正向交互的数据中学习的机器学习任务和模型都可以使用这一解决方案。

在多场景神经网络推荐系统建模方面，本文基于所设计的高效非采样学习算法，针对推荐系统在现实应用场景中包含的不同数据特点，研究如何高效且精细化建模结合异质（Heterogeneous）交互行为（社交网络、多类型交互行为）与内容信息（特征信息、知识图谱）的神经网络推荐模型，包括：（1）在结合社交网络的推

荐场景下, 本文提出了一个基于自适应迁移学习的高效社交推荐模型来联合学习用户社交行为和物品兴趣偏好, 实现社交行为的充分利用; (2) 在结合如点击、加入购物车、购买等多类型交互行为的推荐系统场景下, 本文提出了基于因果推理的高效多行为推荐模型和基于异质图神经网络的高效多行为推荐模型, 挖掘不同类型行为隐含的用户偏好, 实现对于用户不同交互行为的精准建模; (3) 在结合特征信息的推荐系统场景下, 本文首先通过理论分析建立了因子分解机 (Factorization Machines)<sup>[47]</sup>和矩阵分解 (Matrix Factorization)<sup>[48]</sup>之间的转换关系, 进而提出了一个高效非采样分解机模型来全面考虑用户偏好与特征信息之间的关联, 提升推荐效果; (4) 在结合知识图谱的推荐系统场景下, 本文首先通过理论推导将在第一部分所提出的非采样学习算法扩展到三元知识图谱表示学习任务, 然后提出了一个基于非采样联合学习的高效知识增强推荐模型。该模型可以很好地融合知识图谱中的特征信息, 帮助对用户偏好进行充分刻画。以上所提出的推荐模型在多个现实数据集上相比已有基准方法在模型表现和训练效率上均取得了显著的提升效果。

在可遗忘推荐系统框架方面, 针对数据可遗忘需求, 本文提出了一个通用的高效可遗忘推荐系统框架。其主要思想是将训练数据划分为多个子集, 为每个子集训练一个子模型, 通过模型融合的方式得到最终的推荐结果。当收到特定数据遗忘的请求时, 只需要将相应的子模型重新训练即可。为了保留数据之间的结构化信息, 本文设计了一系列新的数据划分算法, 根据数据之间的相似性将其划分为平衡的组。然后, 考虑到不同的子模型对最终预测的贡献不同, 本文进一步设计了一种自适应聚合方法来提高全局模型的表现。现实数据集上的实验表明, 所提出的方法可以在保证推荐精度的前提下实现高效的数据遗忘。同时, 该框架是跟模型无关的, 广泛适用于各种推荐系统模型和应用场景。

### 1.3 本文结构安排

本文分为七章进行组织, 具体的结构安排如下:

第1章为绪论, 概述了本文的选题背景与意义, 指出了精准、高效的神经网络推荐系统在“学习方法”、“推荐模型”、“系统平台”三个层面所面临的挑战, 并且阐述了本文的研究内容与取得的成果。

第2章为相关工作, 主要介绍国内外在推荐系统隐式数据学习、推荐系统多数数据场景建模、数据遗忘研究方法领域的研究现状与不足。

第3章研究了如何从海量隐式数据中实现精准且高效的非采样学习。通过严谨的数学推理, 设计了一系列高效的、可用于神经网络训练的非采样学习算法, 显著

降低了从全样本数据中学习的时间复杂度。在现实数据集上的实验表明，相对于已有基准方法推荐精准度提升 5% 以上，训练速度提升 5-30 倍。本章也对负采样策略和非采样策略在推荐系统上的应用进行了分析。本章相关成果发表于 CCF A 类期刊论文 TOIS 2020<sup>[49]</sup> 和 TOIS 2022<sup>[50]</sup>。

第4章研究了结合异质交互行为的高效神经网络推荐模型。在现实数据集上的实验表明，在包含社交网络的推荐场景，本文提出的基于自适应迁移学习的高效社交推荐模型相对于已有基准方法推荐精准度提升 4% 以上，训练速度提升 7 倍以上；在包含多类型交互行为的推荐场景，本文提出了基于因果推理的高效多行为推荐模型和基于异质图神经网络的高效多行为推荐模型，相对于已有基准方法推荐精准度提升 40% 以上，训练速度提升 10 倍以上。本章相关成果发表于 CCF A 类会议论文 SIGIR 2019<sup>[51]</sup>、AAAI 2020<sup>[52]</sup>、和 AAAI 2021<sup>[53]</sup>。

第5章研究了结合内容信息的高效神经网络推荐模型。在现实数据集上的实验表明，在包含特征信息的推荐场景，本文提出的高效非采样分解机模型相对于已有基准方法推荐精准度提升 9% 以上，训练速度提升 5 倍以上；在包含知识图谱的推荐场景，本文提出的基于非采样联合学习的高效知识增强推荐模型相对于已有基准方法推荐精准度提升 5% 以上，训练速度提升 20 倍以上。本章相关成果发表于 CCF A 类会议论文 TheWebConf (旧称 WWW) 2020<sup>[54]</sup>和 SIGIR 2020<sup>[55]</sup>。

第6章研究了高效可遗忘推荐系统框架。现实数据集上的实验表明，所提出的方法可以在保证推荐精度的前提下实现高效的数据遗忘。本章相关成果发表于 CCF A 类会议论文 TheWebConf 2022<sup>[56]</sup>。

第7章为本文总结并对未来研究工作进行了展望。

本文作者为全部所提到工作的第一作者（不含共一）。所有工作均已开源<sup>①</sup>。

---

① <https://github.com/chenchongthu>

## 第2章 国内外研究现状

个性化推荐系统是学术界和产业界的重要研究及应用课题。在学术研究方面,推荐系统是近年来的研究热点,成果广泛发表在人工智能、数据挖掘和信息检索领域的国际顶级学术会议 AAI, TheWebConf, WSDM, KDD, SIGIR 和期刊 TOIS、TKDE 等。在产业应用上,推荐系统做为当前人工智能商业化最成功的技术之一,相关技术也受到谷歌、微软、阿里巴巴、字节跳动等科技公司的积极关注和迭代更新。本节从推荐系统中的隐式数据学习、多数据场景建模、数据遗忘研究方法三个方面阐述国内外研究现状与不足。

### 2.1 推荐系统中的隐式数据学习

从隐式数据中学习是广泛意义上大多数机器学习任务中的一个基本操作,如知识图谱构建<sup>[57]</sup>、图表示学习<sup>[58]</sup>、推荐系统<sup>[6]</sup>等,其关键是如何从未观测到的缺失数据中获取负反馈信息。针对这一问题,目前有两类广泛使用的学习策略:(1) 负采样策略<sup>[6]</sup>和 (2) 非采样策略<sup>[7]</sup>。

负采样策略通过采样的方式从缺失数据(未观测到的用户—物品交互记录)中选出一部分作为负样本。其优势在于方便并且具备较好的通用性,无需特别设计即可用于不同结构的推荐系统模型。然而由于采样的不确定性,负采样策略的有效性容易受到所采负样本质量的影响,使得模型收敛速度缓慢并且很难收敛到最优的状态<sup>[21,43]</sup>。因此,近年来的一些负采样的相关研究主要围绕在如何提高负样本质量。现有的负采样策略根据不同的研究路线可以分为以下四种:(1) 基于固定分布的负采样<sup>[6,59]</sup>,是目前最常用的负采样方法,以相同的概率分布,或者是基于物品流行度的概率分布等方式从缺失数据中采样负例。其采样效率较高,但质量不稳定;(2) 基于困难样本的负采样(Hard Negative Sampling)<sup>[5,60]</sup>。这种方法倾向于过采样较困难的候选样本。“困难样本”指当前模型尚无法准确识别的样本,因此其潜在的可学习的信息量也相应的较高。面临的问题在于困难样例的识别涉及模型和数据的动态变化,容易增加错误的负样本(False Negative,即缺失数据中的潜在正样本)参与的风险;(3) 基于生成对抗模型的负采样<sup>[40,61-62]</sup>,使用对抗学习<sup>[63]</sup>的方式,让生成器产生越来越困难的负样例,从而提升鉴别器(推荐模型)对于负样本的鉴别能力,进而提升推荐系统的表现。这类方法的有效性已经在现有工作中得到了验证,面临的问题则是由于采样过程通常需要遍历所有缺失数据,计算效率较低;(4) 基于图的负采样<sup>[64-66]</sup>,利用图结构信息从缺失数据中进行采

样。如通过计算图节点的度排序、在图上随机游走、以及将多个样例结合等方式选择更有信息量的负样例。通常被应用于基于图神经网络的推荐系统模型，并且展示出了不错的效果。面临的问题是由于该采样策略通常是基于特定图模型设计的，对于不同模型的泛化性较差。

非采样策略<sup>[7-9]</sup>，又称全采样学习策略，将所有的缺失数据看做是负样本，并根据其不确定性合理分配样本权重以缓解数据不平衡问题和降低误判风险。样本权重可以被设置为固定值<sup>[7]</sup>、与物品流行度相关的自适应值<sup>[8-9]</sup>或是根据概率图模型所计算的估计值<sup>[22]</sup>等。这类方法的优势在于可以使用到全部的训练数据，模型的收敛效果通常相比于负采样策略更好。所面临的问题则是传统的非采样学习方法复杂度高，在大规模数据场景下计算效率较差。虽然近几年出现了一些研究工作来缓解非采样学习的效率问题，但这些方法往往是基于交替最小二乘法和EM算法<sup>[7,23]</sup>，或针对传统推荐系统模型所设计的<sup>[43,67]</sup>，不适用于目前广泛应用的基于梯度下降的深度学习和神经网络模型。综上所述，非采样学习虽然在模型优化效果上具有优势，但现有方法较高的复杂度使其难以在神经网络推荐模型中得以应用。隐式数据中通用、高效的非采样学习是一个亟待研究的重要课题。

## 2.2 推荐系统多数据场景建模

### 2.2.1 协同过滤推荐方法简要回顾

协同过滤 (Collaborative Filtering) 是个性化推荐系统的关键技术，其主要思想是通过学习用户与物品之间的历史交互行为来挖掘用户偏好，实现对于用户未来行为的预测<sup>[48,68]</sup>。传统协同过滤方法直接通过计算用户或物品之间交互的相似度来学习向量表示，包括基于用户的协同过滤<sup>[69]</sup>、基于物品的协同过滤<sup>[70]</sup>、以及随后的一些矩阵分解方法<sup>[6-7,20,71-72]</sup>。近年来，随着深度学习的发展，基于神经网络的协同过滤技术逐渐成为推荐模型构建的主流。通过与注意力机制<sup>[4,73]</sup>、卷积神经网络<sup>[36]</sup>、循环神经网络<sup>[38]</sup>、记忆网络<sup>[74]</sup>、生成对抗网络<sup>[40]</sup>、图神经网络<sup>[41]</sup>等相结合，在用户物品的稠密向量表示 (Embedding) 和交互函数计算<sup>[75]</sup>等方面优化或替代了传统的隐空间因子模型 (Latent Factor Model) 方法。然而，由于仅使用用户和物品的历史交互数据进行训练，这些协同过滤方法在应对稀疏数据和冷启动问题时面临了极大的挑战。

在推荐系统现实应用场景中，除了与业务 KPI (Key Performance Indicator) 直接相关的用户—物品交互数据外，也广泛存在着许多额外信息可以帮助学习用户的兴趣偏好，包括异质交互行为信息如社交网络、多类型交互，以及内容信息如用户和物品特征、知识图谱等。下面对于这些结合额外信息的推荐系统模型进行介

绍。

### 2.2.2 结合异质交互行为的推荐方法

社交网络在当前互联网平台中十分常见，因此结合社交行为的推荐系统在现实生活中有着广泛的应用场景。现有的结合社交的推荐系统<sup>[12,26,30]</sup>主要基于以下两种假设：(1) 用户和其社交好友的偏好是相似的，即用户兴趣受到社交好友的影响<sup>[76]</sup>；(2) 用户对于社交领域中好友的偏好与其对于物品交互领域中的物品偏好是相似的，即用户对于不同领域之间的偏好具有一定程度的共性<sup>[30]</sup>。基于第一种假设，一些研究工作<sup>[76-78]</sup>在用户的偏好学习过程中引入来自于社交好友的相似性约束，通过建模社交好友的影响来提升推荐系统的效果。基于第二种假设，一些研究将预测社交领域的用户—用户关系视为预测用户—物品交互之外的另一个任务，通过挖掘用户对于社交好友的偏好来增强其对物品偏好的学习<sup>[30,79-80]</sup>。这一类工作经常使用到的技术是迁移学习 (Transfer Learning)<sup>[81-82]</sup>，其核心思想是将共有知识从源领域迁移到目标领域。如前所述，社交网络同样包含用户的偏好信息，这就为迁移学习提供了基础。然而，现有的迁移学习方法通常是静态的，不同领域之间的知识共享程度无法针对不同的用户动态改变。在真实互联网平台中，不同的用户对于跨领域之间偏好的共享程度是不同的，如有的用户在两个领域之间具有非常相似的表现，而有的用户则会展现出自己不一样的一面。现有方法对于结合社交网络的推荐仍停留在较为粗犷的地步，缺乏对结合用户社交行为的精细化研究。

另一方面，在实际互联网平台上可以收集到用户对于物品的多种交互信息<sup>[13]</sup>。例如在电商平台上的点击、收藏、加入购物车、购买等行为。对于这些多类型交互数据的使用，现有的研究可以被分为两类：(1) 将对不同类型交互数据的刻画建模到对目标行为（如购买）的预测任务中<sup>[83-84]</sup>。这一类方法的核心思想是不同类型的交互行为反应了用户对物品不同程度的兴趣偏好，对于目标行为的学习起到的作用有强弱之分。其不足是仅优化一个预测任务，模型表达能力有限；(2) 基于多任务学习的方式，对不同的交互行为分别进行预测，并通过参数共享协同学习<sup>[13,80]</sup>。这一类方法的不足在于现有方法通常是基于负采样技术进行优化，而负采样策略在多任务形式下其不稳定性会成倍增加，且某些样本会在不同行为的预测时被同时作为正样本和负样本，造成了优化矛盾，使得模型整体更难收敛到最优的状态。此外，现有方法对于结合多类型交互行为的推荐未能建模不同行为之间的因果推断关系，缺乏对于不同行为之间关系的显式刻画。

### 2.2.3 结合内容信息的推荐方法

结合内容信息的推荐旨在利用用户和物品的特征信息或是所具有的知识图谱信息来增强物品推荐的效果<sup>[14,47,85]</sup>。

对于特征信息的使用, 现有的研究主要是通过不同方式建模特征之间的交互关系, 可分为以下两种<sup>[37,86]</sup>: (1) 通过专家手动筛选组合特征, 以显式的方式进行特征交叉, 然后将交叉特征传入预测模型如逻辑回归模型<sup>[14]</sup>或神经网络<sup>[87]</sup>。显然这种方式需要大量的人工成本、专业工程技术和领域知识, 缺乏在不同领域应用的灵活性; (2) 通过自动组合特征的方式进行特征交叉。代表性的方法包括基于自动机器学习 (Auto Machine Learning)<sup>[75]</sup>和基于因子分解机的方式<sup>[28,47]</sup>。其中因子分解机由于其灵活性和通用性得到了广泛的应用和发展。然而, 目前关于因子分解机的研究通常侧重于对于高阶特征交互的挖掘, 如一些引入非线性层<sup>[28,88]</sup>、注意力机制<sup>[34]</sup>、和卷积神经网络<sup>[37,89]</sup>的方法。在推荐系统优化过程中因子分解机仍需结合负采样策略进行学习<sup>[90-91]</sup>, 这使得其表现容易收到负采样缺点的影响, 无法达到最佳的效果。同时, 高阶神经网络的引入也使得基于深度学习的因子分解机模型复杂度更高, 限制了在大规模现实数据场景下的应用。

在推荐任务上, 知识图谱的引入能够帮助建立起物品之间的联系, 从而更精准地刻画物品和用户特征。现有的结合知识图谱的推荐方法主要可以分为以下两类: (1) 直接利用知识图谱的网络结构, 通过随机游走等方式来利用图谱上实体间的联系来改进现有推荐方法<sup>[92-95]</sup>。这类方法主要优势在于保留了知识图谱上的结构信息, 可以很方便地生成可解释的推荐理由, 并且提升推荐算法的表现。其不足之处在于效率和效果往往受到随机游走的不确定性限制, 且依赖于人工定义游走的路径规则; (2) 从知识图谱中学习用户或物品的额外特征, 然后把它结合到推荐模型中<sup>[15,29,96]</sup>。这类方法的主要优势在于能够从知识图谱中直接获得物品和用户的表示, 可以方便的与现有的神经网络推荐方法相结合。不足之处是现有工作大都采用了图谱表示学习和推荐任务分别训练的策略, 如交替训练每个任务 (Alternative Learning)<sup>[96]</sup>或是训练完成一个任务后再训练另一个 (One-by-one Learning)<sup>[15]</sup>, 这降低了每个任务之间的协同性。同时, 现有工作广泛依赖于负采样学习, 对算法的优化比较有限, 面临效率问题以及负采样的不稳定问题。

## 2.3 数据遗忘研究方法

数据遗忘问题又称机器遗忘, 即在收到数据去除的请求时 (如用户主动要求删除数据, 坏数据去除等), 模型如何高效、可靠地去除这些敏感数据对于学习结果的影响<sup>[16,45]</sup>。针对这一问题, 目前有两个方向上的研究: (1) 统计意义上的遗忘

(Statistical Unlearning) 和 (2) 绝对遗忘 (Exact Unlearning)。统计意义上的遗忘放宽了对于数据遗忘的要求，只需保证在一定的误差范围内无法区分数据遗忘后的模型和未用该数据训练的模型<sup>[97-101]</sup>。这一类方法通常是基于梯度反推来快速消除特定数据对于模型参数的影响，目前在线性模型如线性回归中展示了不错的效果。与绝对遗忘相比，统计意义上的遗忘计算效率更高，不需要对模型重新训练。然而，该类方法对于数据遗忘的保证是统计意义上的，不适用于一些受法律严格管制的机器学习应用。同时，目前基于梯度反推的遗忘技术局限于浅层和线性模型，在深度学习和复杂神经网络模型上的研究还比较匮乏。绝对遗忘旨在确保从模型中完全删除特定数据及其影响<sup>[16-17]</sup>。最简单的绝对遗忘方法是用去除要删除数据后的训练集重新训练模型，但是会带来巨大的计算开销。为了解决效率问题，近年来一种有代表性的方式<sup>[16]</sup>是将训练数据划分为多个子集，为每个子集训练一个子模型，通过模型融合的方式得到最终的结果。当收到特定数据遗忘的请求时，只需要将相应的子模型重新训练即可。然而现有的研究主要集中在计算机视觉和自然语言处理的相关任务上，缺乏对于推荐系统所使用的结构化历史交互数据的考量，直接应用于推荐任务上时会造成较大的精度损失。因此，如何在推荐系统任务上保证精准度的同时实现通用且高效率的数据遗忘需要进一步的研究和探索。

## 第3章 高效非采样隐式数据学习研究

### 3.1 引言

从隐式数据中学习是广泛意义上大多数机器学习任务中的一个基本操作<sup>[6,57-58,102]</sup>。在推荐系统任务中,用户对物品的历史交互被记录为隐式数据,如新闻点击记录,视频浏览历史等。这些隐式数据中往往只包含正反馈,而负反馈信号是缺失的。例如,对于新闻平台中的点击行为,如果用户点击了某个新闻,那么系统将收集到正反馈(标签为“1”),说明用户对于该新闻感兴趣。但是对于用户未点击的新闻,原因可能是由于用户对该新闻不感兴趣,或者是根本没有看到它<sup>[6,7]</sup>,此时系统中对于用户和该新闻的记录是缺失的。因此,在缺失的数据中,真实的负反馈(用户不感兴趣)和无标签数据(用户没看到)是混合在一起的。如何正确的从缺失数据中学习用户兴趣,对于推荐系统学习至关重要。针对这一问题,目前有两类广泛使用的学习策略:(1)负采样策略,即通过采样的方式从缺失数据中选择出一部分作为负样本<sup>[6,20-21]</sup>; (2)非采样策略,又称全采样策略,即将所有的缺失数据看做是负样本,并根据其不确定性合理分配样本权重以缓解数据不平衡问题和降低误判风险<sup>[7-9,22]</sup>。这两种策略各有利弊:负采样的优势是训练数据少,效率较高,但其有效性容易受到所采样的负样本质量的影响,使得模型很难收敛到最优的状态。与之相反,非采样学习考虑了所有的训练数据,在模型训练效果上更有优势。但传统的非采样学习方法复杂度高,计算效率较差。

同时,近年来深度学习和神经网络促进了推荐系统模型的进一步研究和发展。然而,当前学术界的研究更多的倾向于引入复杂和先进的神经网络如注意力机制(Attention Mechanisms)<sup>[33-34]</sup>,记忆网络(Memory Networks)<sup>[12,35]</sup>,卷积神经网络(Convolutional Neural Networks)<sup>[36-37]</sup>,循环神经网络(Recurrent Neural Networks)<sup>[38]</sup>,生成对抗网络(Generative Adversarial Networks)<sup>[39-40]</sup>,图神经网络(Graph Neural Networks)<sup>[41-42]</sup>等来提升推荐效果,在模型训练效率方面却关注较少。现有的神经网络推荐系统模型存在着两个重要的局限性:(1)复杂的神经网络结构往往包含大量的模型参数<sup>[11,103]</sup>,即使使用基于负采样的学习策略,也需要昂贵的计算资源;(2)广泛使用的负采样学习策略性能不稳定,其有效性容易受到采样质量的影响,使得模型很难收敛到最优的状态<sup>[43]</sup>。

为了缓解由于负采样学习策略带来的效果损失,本文提出使用非采样学习策略来优化神经网络推荐系统。然而,传统非采样学习方法由于需要遍历全部用户—物品交互数据,复杂度较高,直接应用于神经网络模型中将显著影响模型的计算

效率，降低模型在大规模数据场景下的实用性。虽然近几年出现了一些研究工作来缓解非采样学习的效率问题，但这些方法往往是基于交替最小二乘法和 EM 算法<sup>[23]</sup>，或针对传统推荐系统模型所设计的<sup>[43,67]</sup>，不适用于目前广泛应用的基于梯度下降的深度学习和神经网络模型。

综上所述，隐式数据中通用、高效的非采样学习是一个亟待研究的重要问题。针对该问题，本文从学习理论出发，通过严谨的数学推理重构目前常用的平方损失函数<sup>[7]</sup>，将因子分解中互相独立的运算分离开来，新设计了一系列高效的、可用于神经网络训练的非采样学习方法。所提出来的方法将从全样本数据中学习的时间复杂度降低了一个数量级，并同时保留了非采样学习的高精度优势。进一步地，基于一个简单的神经网络矩阵分解模型，本章将所提出的高效非采样学习算法在推荐系统任务上实例化，并命名为高效神经网络矩阵分解模型 (ENMF, Efficient Neural Matrix Factorization)。在现实数据集上的实验表明，所提出的方法相对于已有方法推荐精准度提升 5% 以上，训练速度提升 5-30 倍。同时，这一研究也是基础性的，本质上，从只有正向交互的数据中学习的机器学习任务和模型都可以使用这一解决方案。

本章的结构安排如下，第3.2节介绍预备知识；第3.3节介绍所提出的高效非采样隐式数据学习方法，包括神经网络矩阵分解框架以及基于用户、基于物品、和交替训练的高效学习算法；第3.4节展示了实验结果；最后，第3.5节对本章进行小结。

## 3.2 预备知识

本节首先介绍所要研究的问题，包括本章使用的数学符号以及传统非采样隐式数据学习方法。

### 3.2.1 符号定义

表3.1描述了本章中使用的数学符号和相应描述。假设数据集中有  $M$  个用户和  $N$  个物品，用户集合记为  $\mathbf{U}$ ，物品集合记为  $\mathbf{V}$ 。我们使用索引  $u$  表示用户，使用  $v$  表示物品。用户—物品交互矩阵记为  $\mathbf{Y} = [y_{uv}]_{M \times N} \in \{0, 1\}$ ，表示  $u$  是否与物品  $v$  有过交互。我们使用  $\mathcal{Y}$  表示  $\mathbf{Y}$  中观察到的正向用户—物品交互。 $\mathbf{p}_u$  是用户  $u$  的隐语意表示向量， $\mathbf{q}_v$  是物品  $v$  的隐语意表示向量， $c_{uv}$  表示训练样例  $y_{uv}$  的权重。更多详细符号定义将在下一节具体使用过程中予以介绍。

表 3.1 高效非采样学习方法中所用数学符号

符号	描述
$U, V$	分别代表用户集合和物品集合
$B$	一批用户或物品
$Y$	用户—物品交互
$\mathcal{Y}$	正向用户—物品交互
$\mathbf{p}_u, \mathbf{q}_v$	分别代表用户 $u$ 和物品 $v$ 的隐语义表示向量
$\mathbf{h}$	预测层向量
$c_{uv}$	训练样例 $y_{uv}$ 的权重
$d$	向量维度
$\Theta$	神经网络参数集合

### 3.2.2 非采样隐式数据学习

在非采样隐式数据设置中，用户与物品的交互记录  $Y$  定义如下：

$$y_{uv} = \begin{cases} 1, & \text{用户 } u \text{ 和物品 } v \text{ 有过交互} \\ 0, & \text{其他} \end{cases} \quad (3.1)$$

可以看到，非采样学习将所有的缺失数据看做是负样本。由于用户和物品之间的交互缺失既可能是用户不感兴趣，也可能是用户没有看到，导致这种负样例假设存在误判的可能，因此需要对不确定性高的负样本相应的分配较小的训练权重。广泛使用的非采样损失函数是加权回归<sup>[7]</sup>，形式如下：

$$\mathcal{L}(\Theta) = \sum_{u \in U} \sum_{v \in V} c_{uv} (y_{uv} - \hat{y}_{uv})^2 \quad (3.2)$$

其中  $\hat{y}_{uv}$  表示推荐模型预测的用户  $u$  对于物品  $v$  的分值。该损失函数的特点是对每一个训练样例给予一个相关联的权重  $c_{uv}$ 。样本权重可以被设置为固定值<sup>[7]</sup>、与物品流行度相关的自适应值<sup>[8-9]</sup>或是根据概率图模型所计算的估计值<sup>[22]</sup>等。从该函数的形式可以分析得出其每一轮训练过程复杂度为  $O(MNd)$ ，其中  $M$  为用户的数量， $N$  为物品的数量， $d$  是向量长度。在推荐系统中， $MN$  即用户数量乘物品数量可以达到上亿级别<sup>[104]</sup>，因此直接使用该损失函数难以在复杂结构的模型中实现高效学习。这正是本章研究想要解决的问题。

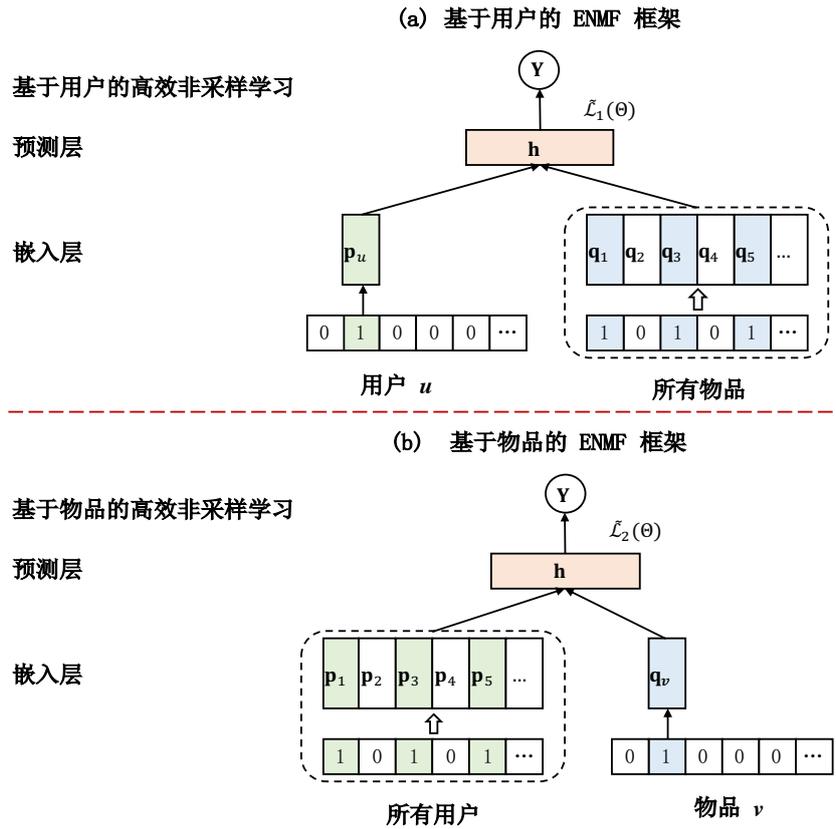


图 3.1 高效神经网络矩阵分解框架 (ENMF) 示意图

### 3.3 高效非采样学习

本节介绍所提出的高效非采样学习算法，我们首先给出一个基于非采样学习的神经网络矩阵分解框架，然后介绍在该框架下的基于用户、基于物品、及交替训练的高效学习算法。

#### 3.3.1 高效神经网络矩阵分解框架

图3.1展示了本文提出的非采样高效神经矩阵分解 (ENMF) 框架。ENMF 的结构遵循现有工作神经网络协同过滤 (NCF, Neural Collaborative Filtering)<sup>[20]</sup>的设计，但有两个主要区别：首先，在输入层，与 NCF 中将用户—物品对  $(u, v)$  作为输入不同，ENMF 使用用户和他/她的所有物品交互（基于用户）或物品和它所有的用户交互（基于物品）作为输入。该输入设置使得 ENMF 可以从整体训练数据中学习。其次在学习方法上，ENMF 没有使用基于采样的策略来训练，而是使用本章接下来所提出的高效非采样方法从全样本空间中训练模型。

在设计框架中，用户和物品首先通过嵌入层转换为密集向量表示。然后，通

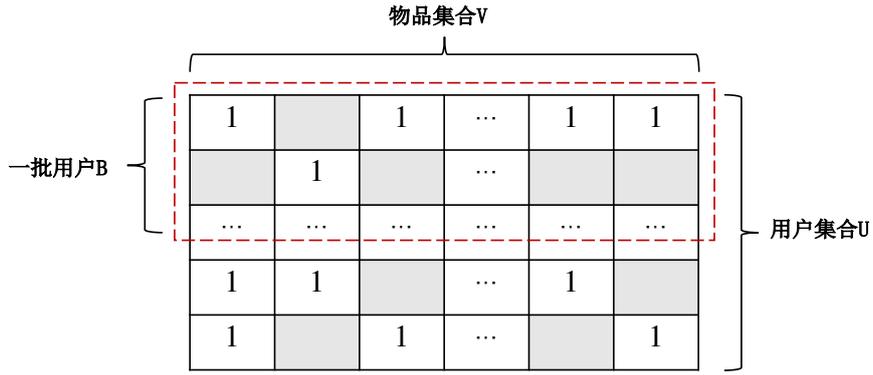


图 3.2 基于用户的高效学习算法输入示意图

过泛化矩阵分解预测用户对于物品的偏好程度，计算方法如下所示：

$$\begin{aligned}\hat{y}_{uv} &= \mathbf{h}^T(\mathbf{p}_u \odot \mathbf{q}_v) \\ &= \sum_{i=1}^d h_i p_{u,i} q_{v,i}\end{aligned}\quad (3.3)$$

其中  $\mathbf{p}_u \in \mathbb{R}^d$  和  $\mathbf{q}_v \in \mathbb{R}^d$  表示用户  $u$  和物品  $v$  的隐语义嵌入向量； $\odot$  符号表示哈达玛积 (Hadamard Product)，即对于两个向量进行逐项相乘； $\mathbf{h}$  表示预测层，为向量空间的不同维度分配相应的权重。

### 3.3.2 基于用户的高效学习算法

本文所设计的高效非采样学习策略有三种形式，这使得其在面对不同的应用场景时更具有灵活性。我们首先介绍基于用户的方法。

为了使公式 (3.2) 所示的损失函数适合学习神经网络模型，我们首先将其调整为批 (Batch) 处理的形式，其中训练所用的批是基于用户选取的。图3.2展示了基于用户的高效学习算法输入示例。在此设置下，对一批用户的损失函数计算如下：

$$\begin{aligned}\mathcal{L}_1(\Theta) &= \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}} c_{uv} (y_{uv} - \hat{y}_{uv})^2 \\ &= \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}} c_{uv} (y_{uv}^2 - 2y_{uv}\hat{y}_{uv} + \hat{y}_{uv}^2)\end{aligned}\quad (3.4)$$

可以看出，计算该损失函数的时间复杂度为  $O(|\mathbf{B}|Nd)$ ，这意味着直接使用该损失函数通常会需要昂贵的计算资源，使得模型难以应用于实际系统中。

在隐式数据中，由于  $\mathbf{Y} = [y_{uv}]_{M \times N} \in \{0, 1\}$ ，信息是确定的且在训练过程中不会发生变化。因此可以用一个常量替换它以简化上公式：

$$\mathcal{L}_1(\Theta) = \text{const} - 2 \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}^+} c_{uv}^+ \hat{y}_{uv} + \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}} c_{uv} \hat{y}_{uv}^2 \quad (3.5)$$

其中将  $y_{uv}$  消除,  $const$  表示跟模型参数  $\theta$  无关的常量值。然后, 通过将缺失数据的损失函数计算值表示为全部数据的损失与正样本数据的损失之间的残差, 可以将上公式进行如下推导:

$$\begin{aligned}
 \mathcal{L}_1(\theta) &= const - 2 \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}^+} c_{uv}^+ \hat{y}_{uv} + \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}^+} c_{uv}^+ \hat{y}_{uv}^2 + \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}^-} c_{uv}^- \hat{y}_{uv}^2 \\
 &= const - 2 \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}^+} c_{uv}^+ \hat{y}_{uv} + \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}^+} c_{uv}^+ \hat{y}_{uv}^2 + \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}} c_{uv}^- \hat{y}_{uv}^2 - \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}^+} c_{uv}^- \hat{y}_{uv}^2 \\
 &= const + \underbrace{\sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}^+} ((c_{uv}^+ - c_{uv}^-) \hat{y}_{uv}^2 - 2c_{uv}^+ \hat{y}_{uv})}_{\mathcal{L}_1^P(\theta)} + \underbrace{\sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}} c_{uv}^- \hat{y}_{uv}^2}_{\mathcal{L}_1^A(\theta)}
 \end{aligned} \tag{3.6}$$

其中  $\mathcal{L}_1^P(\theta)$  记录了对于所有正样本交互需要计算的内容,  $\mathcal{L}_1^A(\theta)$  表示需要对所有数据进行的计算。因为真实场景下隐式数据中正样本数量相对稀疏, 因此该损失函数的主要计算瓶颈在于  $\mathcal{L}_1^A(\theta)$  部分。接下来, 我们将展示如何通过数学解耦将相互独立的运算分离开来降低计算复杂度。

$\hat{y}_{uv}$  的计算如公式 (3.3) 所示, 将其代入  $\hat{y}_{uv}^2$ , 并且对向量拆项得到:

$$\begin{aligned}
 \hat{y}_{uv}^2 &= \sum_{i=1}^d h_i p_{u,i} q_{v,i} \sum_{j=1}^d h_j p_{u,j} q_{v,j} \\
 &= \sum_{i=1}^d \sum_{j=1}^d (h_i h_j) (p_{u,i} p_{u,j}) (q_{v,i} q_{v,j})
 \end{aligned} \tag{3.7}$$

通过将公式 (3.7) 的结果代入  $\mathcal{L}_1^A(\theta)$ , 并且将训练权重  $c_{uv}^-$  设置为固定常数<sup>[7]</sup>或者是跟物品流行度相关的值  $c_v^-$ <sup>[9]</sup>时,  $p_{u,i}$  和  $q_{v,i}$  的计算过程可以被完全拆开单独计算。在此基础上,  $\sum_{v \in \mathbf{V}} c_v^- q_{v,i} q_{v,j}$  和  $\sum_{u \in \mathbf{B}} p_{u,i} p_{u,j}$  的计算也是独立的。此时,  $\mathcal{L}_1^A(\theta)$  的计算如下:

$$\mathcal{L}_1^A(\theta) = \sum_{i=1}^d \sum_{j=1}^d \left( (h_i h_j) \left( \sum_{u \in \mathbf{B}} p_{u,i} p_{u,j} \right) \left( \sum_{v \in \mathbf{V}} c_v^- q_{v,i} q_{v,j} \right) \right) \tag{3.8}$$

对于  $\mathcal{L}_1^A(\theta)$  中运算单元的重新组合是提升算法效率的关键。改进前的计算复杂度为  $O(|\mathbf{B}|Nd)$ , 改进后的计算复杂度则为  $O((|\mathbf{B}| + N)d^2)$ , 因此较原有的计算方式将会有很大的效率提升。

通过将公式 (3.8) 代入公式 (3.6), 并且去除与梯度计算无关的参数项  $const$ ,

**算法 3.1** 基于用户的高效非采样学习算法

**Require:** 训练集  $\{\mathbf{Y}, \mathbf{U}, \mathbf{V}\}$ ; 训练样例权重  $c$ ; 学习率  $\eta$ ; 向量长度  $d$

**Ensure:** 神经网络模型参数  $\Theta$

- 1: 随机初始化模型参数  $\Theta$
- 2: **while** 结束条件尚未满足 **do**
- 3:   选出一批训练用户及相应训练样例  $\{\mathbf{Y}_B, \mathbf{B}, \mathbf{V}\}$
- 4:   计算损失函数  $\tilde{\mathcal{L}}_1(\Theta)$  (公式 (3.9))
- 5:   更新模型参数
- 6: **end while**
- 7: **return**  $\Theta$

最终的基于用户的高效非采样损失函数为:

$$\begin{aligned} \tilde{\mathcal{L}}_1(\Theta) = & \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}^+} ((c_v^+ - c_v^-) \hat{y}_{uv}^2 - 2c_v^+ \hat{y}_{uv}) \\ & + \sum_{i=1}^d \sum_{j=1}^d \left( (h_i, h_j) \left( \sum_{u \in \mathbf{B}} p_{u,i} p_{u,j} \right) \left( \sum_{v \in \mathbf{V}} c_v^- q_{v,i} q_{v,j} \right) \right) \end{aligned} \quad (3.9)$$

其中如前所述,  $c_{uv}$  被简化为  $c_v$ 。值得注意的是, 因为在推导过程中没有引入近似操作, 公式 (3.9) 所示的高效学习损失函数严格等同于原始公式 (3.4) 所示的损失函数。具体的基于用户的高效非采样学习算法训练流程参见算法3.1。

形式化地, 我们可以对上述推导过程给出如下定理:

**定理 3.1:** 对于双塔推荐模型, 如果用户  $u$  对物品  $v$  的预测函数符合公式 (3.3) 的形式 (预测层为线性), 则对其通过非采样学习进行优化时, 使用公式 (3.9) 与公式 (3.4) 的梯度计算结果是完全等价的。

此定理也将被广泛应用于接下来的章节中我们所设计的高效神经网络推荐模型。

### 3.3.3 基于物品的高效学习算法

本小节介绍基于物品的高效非采样学习算法, 其中训练所用的批数据是基于物品选取的。图3.3展示了输入示例。在此设置下, 对一批物品的损失函数计算如下:

$$\mathcal{L}_2(\Theta) = \sum_{u \in \mathbf{U}} \sum_{v \in \mathbf{B}} c_{uv} (y_{uv} - \hat{y}_{uv})^2 \quad (3.10)$$

其中  $\mathbf{B}$  表示一批物品。

基于用户的方法和基于物品的方法之间的主要区别在于批次是基于用户还是基于物品选取。两种形式的推导过程相似, 为了避免重复, 本文省略了基于物品的

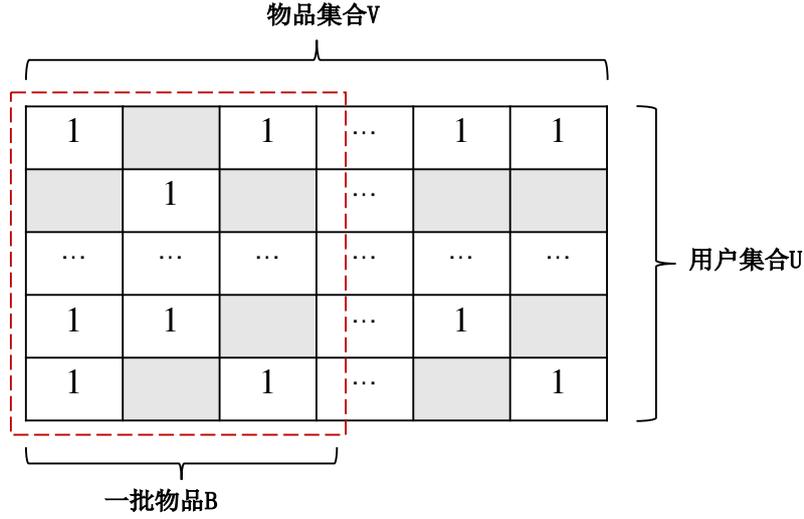


图 3.3 基于物品的高效学习算法输入示意图

### 算法 3.2 基于物品的高效非采样学习算法

**Require:** 训练集  $\{\mathbf{Y}, \mathbf{U}, \mathbf{V}\}$ ; 训练样例权重  $c$ ; 学习率  $\eta$ ; 向量长度  $d$

**Ensure:** 神经网络模型参数  $\Theta$

- 1: 随机初始化模型参数  $\Theta$
- 2: **while** 结束条件尚未满足 **do**
- 3: 选出一批训练物品及相应训练样例  $\{\mathbf{Y}_B, \mathbf{B}, \mathbf{U}\}$
- 4: 计算损失函数  $\tilde{\mathcal{L}}_2(\Theta)$  (公式 (3.11))
- 5: 更新模型参数
- 6: **end while**
- 7: **return**  $\Theta$

方法的详细推导过程，仅展示以下最终结果：

$$\begin{aligned} \tilde{\mathcal{L}}_2(\Theta) = & \sum_{u \in \mathbf{U}^+} \sum_{v \in \mathbf{B}} ((c_v^+ - c_v^-) \hat{y}_{uv}^2 - 2c_v^+ \hat{y}_{uv}) \\ & + \sum_{i=1}^d \sum_{j=1}^d \left( (h_i h_j) \left( \sum_{u \in \mathbf{U}} p_{u,i} p_{u,j} \right) \left( \sum_{v \in \mathbf{B}} c_v^- q_{v,i} q_{v,j} \right) \right) \end{aligned} \quad (3.11)$$

具体的基于物品的高效非采样学习算法训练流程参见算法3.2。

#### 3.3.4 交替训练的高效学习算法

到目前为止，本文已经介绍了两种高效非采样学习算法，分别基于用户和基于物品构建训练样例。尽管这两方法在使用中都能取得良好的效果，但它们仍然可能遇到一个潜在的问题，即用户和物品之间的学习速度不平衡。以基于用户的方法为例，其每个训练样例是单个用户与所有的物品。这将导致每一轮训练中物品梯度更新的次数远高于用户。因此，物品可能会过训练，而此时用户处于欠训练的状态。虽然现有工作已经提出了一些技巧，如 Adagrad<sup>[105]</sup>之类的自适应学习率

**算法 3.3** 交替训练的高效非采样学习算法**Require:** 训练集  $\{\mathbf{Y}, \mathbf{U}, \mathbf{V}\}$ ; 训练样例权重  $c$ ; 学习率  $\eta$ ; 向量长度  $d$ **Ensure:** 神经网络模型参数  $\Theta$ 

```

1: 随机初始化模型参数  $\Theta$ 
2: while 结束条件尚未满足 do
3:   for 基于用户的训练步骤 do
4:     选出一批训练用户及相应训练样例  $\{\mathbf{Y}_B, \mathbf{B}, \mathbf{V}\}$ 
5:     计算损失函数  $\tilde{\mathcal{L}}_1(\Theta)$  (公式 (3.9))
6:     更新模型参数
7:   end for
8:   for 基于物品的训练步骤 do
9:     选出一批训练物品及相应训练样例  $\{\mathbf{Y}_B, \mathbf{B}, \mathbf{U}\}$ 
10:    计算损失函数  $\tilde{\mathcal{L}}_2(\Theta)$  (公式 (3.11))
11:    更新模型参数
12:  end for
13: end while
14: return  $\Theta$ 

```

参数更新算法来缓解这个问题，但我们仍然希望可以从根本上解决问题。

面对基于用户的方法和基于物品的方法，一个自然的想法便是如何融合这两种形式，以便它们相互加强，从而缓解学习不平衡问题并更好地建模用户—物品交互数据。受到传统协同过滤方法中交替最小二乘法<sup>[7,9]</sup>思想的启发，本文进一步提出了一种交替训练的高效学习算法。在每一轮训练中，第一步使用基于用户的损失函数，第二步使用基于物品的损失函数，交替对模型参数进行更新训练。算法3.3中展示了交替训练的高效学习算法训练流程。

### 3.3.5 讨论

以上我们详细介绍了所提出的高效非采样学习算法，本节中对其时间复杂度、局限性、及训练方法进行讨论。

对于基于用户的方法，更新一批训练数据的时间复杂度为  $O((|\mathbf{B}| + |\mathbf{V}|)d^2 + |\mathcal{Y}_B|d)$ ，其中  $\mathcal{Y}_B$  是每批训练数据中实际存在的用户—物品正向交互。因此每一轮训练的时间复杂度为  $O((|\mathbf{U}| + \frac{|\mathbf{U}||\mathbf{V}|}{|\mathbf{B}|})d^2 + |\mathcal{Y}|d)$ 。与之相似，对于基于物品的方法，每一轮训练的时间复杂度为  $O((|\mathbf{V}| + \frac{|\mathbf{U}||\mathbf{V}|}{|\mathbf{B}|})d^2 + |\mathcal{Y}|d)$ 。对于交替训练的方法，其时间复杂度为上述二者之和。但是值得一提的是，由于交替训练的方法往往使得模型收敛所需的训练轮数更少，因此在实际使用中整体所需的训练时间通常优于基于用户和基于物品的方法。如果不使用本文所提出的高效非采样学习算法，计算原始损失函数的时间复杂度为  $O(|\mathbf{U}||\mathbf{V}|d)$ 。考虑到实际场景中  $|\mathcal{Y}| \ll |\mathbf{U}||\mathbf{V}|$  且  $d \ll |\mathbf{B}|$ ，优化后的高效非采样学习算法有了数量级的效率提升。同时，因为在推

表 3.2 评估 ENMF 方法的数据集的基本统计信息

数据集	用户数量	物品数量	用户—物品交互数量	稠密度
<i>Ciao</i>	7,267	11,211	157,995	0.19%
<i>Epinion</i>	20,608	23,585	454,002	0.09%
<i>MovieLens</i>	6,940	3,706	1,000,209	4.47%

导过程中没有引入近似操作，本文所提出的高效损失函数在梯度计算上严格等同于原始损失函数。

当然，本文所提出的高效非采样算法也存在一定的局限性，主要表现在损失函数推导的过程中限制了最终输出层只能为线性预测层，而不能是非线性层。即便如此，由于目前基于线性预测的双塔模型在实际推荐系统建模中占据了主流<sup>[41-42,106]</sup>，我们的方法仍具有广阔的应用前景和扩展空间。未来研究中我们也将继续尝试将该系列学习方法推广到预测层为非线性的情况。

对于所提出的高效非采样学习算法，可以用随机梯度下降（SGD, Stochastic Gradient Descent）进行参数更新，由于现代深度学习工具包如 TensorFlow<sup>①</sup>、PyTorch<sup>②</sup>等可以为参数更新自动计算梯度与相应的更新值，因此这里不再对梯度计算做过多赘述。

### 3.4 实验

本实验所用代码已开源<sup>③</sup>。

#### 3.4.1 实验设置

##### 3.4.1.1 数据集介绍

我们选择三个被广泛使用的推荐系统公开数据集进行实验，分别是 *Ciao*<sup>④</sup>，*Epinion*<sup>⑤</sup> 和 *MovieLens*<sup>⑥</sup>。对三个数据集的简单介绍如下：

- *Ciao*：此数据集包含用户对所购买物品的评分。由于本文重点关注隐式数据，因此具体的评分被转换为 0 或 1 的值，表示用户是否对某个物品进行了评分。
- *Epinions*：Epinions 是一个广泛使用的推荐数据集。它同样收集了用户对于

① <https://www.tensorflow.org>

② <https://pytorch.org>

③ <https://github.com/chenchongthu/ENMF>

④ <http://www.jiliang.xyz/trust.html>

⑤ <https://alchemy.cs.washington.edu/data/epinions/>

⑥ <https://grouplens.org/datasets/movielens/1m/>

对所购买物品的评分。在我们的实验中，相应的评分也被转换为隐式数据。

- **Movielens**: Movielens 是一个电影评分数据集，广泛被用于评估推荐系统算法的性能。在我们的实验中，我们选择了包含一百万个评分的数据集版本，其中每个用户至少有 20 个评分。这是我们实验中所用最大的数据集。

三个数据集的基本统计信息见表3.2。

### 3.4.1.2 基准方法

为了评估所提出的高效非采样算法及 ENMF 框架的性能，我们比较了以下方法：

- **Most Popular (MP)**: 最简单的非个性化推荐方法，根据物品的流行度对物品进行排序。
- **ItemKNN**<sup>[70]</sup>: 基于物品交互记录的相似度建模隐式数据的协同过滤方法。
- **BPR**<sup>[6]</sup>: 贝叶斯个性化排序 (Bayesian Personalized Ranking)，通过优化正负样本的偏序损失来训练矩阵分解类的推荐方法。
- **WMF**<sup>[7]</sup>: 传统的基于矩阵分解和非采样学习的推荐模型，其中训练样例权重设置为固定值。
- **ExpoMF**<sup>[8]</sup>: 同样基于矩阵分解和非采样学习的推荐模型，其中训练样例权重通过对物品的曝光 (Exposure) 概率进行建模。
- **GMF**<sup>[20]</sup>: 泛化矩阵分解模型，其模型框架与本文的 ENMF 模型完全一致，区别在于使用传统负采样学习策略进行参数优化。
- **NCF**<sup>[20]</sup>: 神经网络协同过滤模型，是一个有代表性的基于神经网络的推荐模型，通过将矩阵分解与多层感知机相结合对隐式反馈数据进行建模。
- **ConvNCF**<sup>[36]</sup>: 卷积神经网络协同过滤方法，通过卷积神经网络对用户和物品向量的外积矩阵进行优化，是目前较为领先的面向隐式反馈数据的推荐模型。

由于本文提出了三种不同的高效非采样学习算法，与 ENMF 推荐框架相结合可产生三个推荐模型实例，分别记为基于用户的 ENMF-U，基于物品的 ENMF-I，以及交替训练的 ENMF-A。表3.3列出了本文所提出的 ENMF 与基准方法的特点比较。

### 3.4.1.3 评价方式

在模型评价过程中，我们将使用不同方法为每个用户对所有的候选物品进行排序（除了训练集和验证集已知的有正向交互的物品）<sup>[107]</sup>。同时，我们采用了在推荐系统评价中被广泛使用的留一法来进行方法评价<sup>[13,20]</sup>。对于每个用户对于物

表 3.3 ENMF 与对比方法的特点比较

模型特点	MP	ItemKNN	BPR	WMF	ExpoMF	GMF	NCF	ConvNCF	ENMF
基于隐式数据推荐	√	√	√	√	√	√	√	√	√
神经网络模型	\	\	\	\	\	√	√	√	√
非采样学习	\	\	\	√	√	\	\	\	√

品的交互记录，我们保留最后一个交互作为测试数据，倒数第二个作为验证数据，其他的所有交互用做训练。在评价指标上，我们采用命中率（HR, Hit Ration）和归一化的累积增益（NDCG, Normalized Discounted Cumulative Gain）。HR@K 主要评价所预测的正例是否在推荐列表中的前 K 个，而 NDCG@K 则还会考虑所预测的正例在推荐列表中的具体位置，越靠前则得分越高。两种评价指标的计算公式如下：

$$HR@K = \frac{1}{|\mathbf{U}|} \sum_u I(|\mathbf{Re}_u \cap \mathbf{T}_u|)$$

$$NDCG@K = \frac{1}{Z} DCG@N = \frac{1}{Z} \frac{1}{|\mathbf{U}|} \sum_u \sum_{i=1}^K \frac{2^{I(|\{re_u^i\} \cap \mathbf{T}_u|)} - 1}{\log_2(i+1)}$$
(3.12)

其中  $I(x)$  是一个函数,当  $x > 0$  时其值为 1,  $x < 0$  时其值为 0;  $\mathbf{Re}_u = \{re_u^1, re_u^2, \dots, re_u^K\}$  代表模型给用户  $u$  的推荐列表,  $K$  是推荐列表长度;  $\mathbf{T}_u$  表示用户  $u$  在测试集中的物品;  $Z$  表示正则化项, 其值为累计增益 (DCG@K) 的最优值。

实验过程中我们对所有方法使用不同的随机数种子重复运行 5 次, 然后汇报各个方法取得的平均表现。

#### 3.4.1.4 超参数设置

所有基准算法的参数设定都是根据对应论文的初始化要求设置的, 且也都在验证集上进行了调参以确保其达到最优的效果。我们对超参数的主要搜索空间包括: 学习率的搜索范围为 [0.005, 0.01, 0.02, 0.05], 为了避免过拟合, 网络留存率 (Dropout) 的搜索范围为 [0.0, 0.1, ..., 0.9], 特征向量长度  $d$  的搜索范围为 [8, 16, 32, 64], 训练样本权重  $c_0$  的搜索范围为 [0.005, 0.01, 0.05, 0.1, 0.2, 0.5, 1]。最终, 根据预实验结果, 每个批的大小被设置成 512, 学习率被设置成 0.05, 特征向量的长度被设置成 64。对于我们的 ENMF 方法, 网络留存率  $\rho$  对于 Ciao 设置为 0.3, 对于 Epinion 设置为 0.5, 对于 Movielens 设置为 0.7。对于训练样本权重, 正样本权重设为 1, 负样本权重  $c_0$  对 Ciao 和 Epinion 设定为 0.05, Movielens 设定为 0.5。这些超参数的影响将在下一小节中进一步探讨。

### 3.4.2 性能评估

#### 3.4.2.1 整体表现分析

本文所提出的方法 (ENMF-U, ENMF-I, ENMF-A) 与其他基准方法在三个数据集上的推荐表现见表3.4, 为了对比不同长度推荐列表的效果, 我们汇报了推荐列表长度为 50、100 和 200 的结果。从实验结果中我们有以下发现:

首先我们可以看到, 使用非采样学习策略的方法通常比基于负采样的方法表现更好。例如, 在表3.4中, WMF 和 ExpoMF 的性能优于 BPR; 我们的 ENMF 表现优于 BPR、GMF、NCF 和 ConvNCF。这一结果与已有研究工作中对非采样学习和负采样训练方法的比较结论是一致的<sup>[9,43,67]</sup>, 即: 非采样学习方法能够取得更优秀的模型表现。

其次, 实验结果表明神经网络方法总体上优于传统的协同过滤方法, 验证了神经网络在推荐系统任务上对于传统模型的优越性。然而, 在实验中, 我们也同时发现神经网络方法 NCF 和 ConvNCF 需要仔细调参以避免模型崩溃。与它们相比, 我们的 ENMF 更稳定, 可以很容易调参以获得最佳性能。

最后, 我们提出的 ENMF 方法在三个数据集上均展示出了最好的效果, 显著优于所有的基线方法 ( $p$  值  $<0.01$ )。即便是与 ConvNCF (一种最近提出的、表达能力非常强的深度学习模型) 相比, 我们的 ENMF-A 在三个数据集上也分别取得了 5.90%、4.87% 和 6.30% 的相对提升。作为基于浅层神经网络的 ENMF 参数量显著少于 ConvNCF, 但是因为非采样训练方法的应用能够让 ENMF 在整个数据集上找到最优解。此外, 与具有相同网络结构但使用负采样进行模型学习的 GMF 相比, ENMF-A 在三个数据集上的表现分别提升 9.36%、5.52% 和 10.30%。这进一步验证了我们的方法的有效性, 也展示了用更好的学习算法改进浅层推荐模型的潜力。

#### 3.4.2.2 训练效率分析

很多深度学习方法仅关注模型是否能够提供更好的结果, 但是忽略了模型实际应用中所面临的训练效率挑战, 因而也很少汇报模型的训练时间<sup>[11]</sup>。实际上如果模型训练或者输出的时间花费过大, 很可能会限制它在真实场景下的应用。本节中, 我们进行实验探索所提出的高效非采样学习算法的训练效率。所有实验均在同一台服务器上运行的, 其配置为: 英特尔 Xeon9 核心 CPU (2.4GHz), 使用了一张 NVIDIA GeForce GTX TITAN X GPU。

我们首先通过实验探索所设计的高效学习算法带来的实际加速效果。我们对以下具有相同模型结构, 但使用不同学习算法的推荐模型进行了比较, 分别是 GMF, ENMF-Original, ENMF-U, ENMF-I 和 ENMF-A。其中 ENMF-Original 表示使用原

表 3.4 ENMF 与不同推荐方法的效果比较, \*\* 表示较最好的基准方法有显著提升, 显著性是  $p < 0.01$ 。“RI”表示 ENMF-A 相较对应方法的相对提升

<i>Ciao</i>	HR@50	HR@100	HR@200	NDCG@50	NDCG@100	NDCG@200	RI
MP	0.1047	0.1384	0.1776	0.0396	0.0452	0.0506	+67.96%
ItemKNN	0.1453	0.1884	0.2468	0.0497	0.0581	0.0668	+26.14%
BPR	0.1531	0.1930	0.2558	0.0517	0.0598	0.0685	+21.91%
WMF	0.1587	0.2011	0.2608	0.0562	0.0631	0.0714	+16.40%
ExpoMF	0.1602	0.1994	0.2613	0.0569	0.0626	0.0709	+16.41%
GMF	0.1668	0.2103	0.2674	0.0633	0.0687	0.0752	+9.36%
NCF	0.1651	0.2108	0.2712	0.0629	0.0695	0.0764	+8.84%
ConvNCF	0.1682	0.2237	0.2741	0.0641	0.0714	0.0787	+5.90%
ENMF-U	<b>0.1750**</b>	<b>0.2296**</b>	<b>0.2945**</b>	<b>0.0651**</b>	<b>0.0741**</b>	<b>0.0830**</b>	-
ENMF-I	<b>0.1749**</b>	<b>0.2311**</b>	<b>0.2946**</b>	<b>0.0643*</b>	<b>0.0734**</b>	<b>0.0823**</b>	-
ENMF-A	<b>0.1757**</b>	<b>0.2331**</b>	<b>0.3015**</b>	<b>0.0662**</b>	<b>0.0753**</b>	<b>0.0850**</b>	-
<i>Epinion</i>	HR@50	HR@100	HR@200	NDCG@50	NDCG@100	NDCG@200	RI
MP	0.0661	0.1068	0.1659	0.0234	0.0299	0.0382	+153.96%
ItemKNN	0.1312	0.2082	0.2929	0.0455	0.0563	0.0682	+34.41%
BPR	0.1708	0.2338	0.3007	0.0548	0.0646	0.0747	+17.04%
WMF	0.1765	0.2384	0.3158	0.0605	0.0685	0.0789	+11.07%
ExpoMF	0.1784	0.2368	0.3064	0.0602	0.0691	0.0781	+11.70%
GMF	0.1811	0.2513	0.3388	0.0613	0.0739	0.0845	+5.52%
NCF	0.1816	0.2534	0.3442	0.0621	0.0750	0.0869	+4.08%
ConvNCF	0.1833	0.2510	0.3418	0.0617	0.0742	0.0851	+4.87%
ENMF-U	<b>0.1893**</b>	<b>0.2647**</b>	<b>0.3523**</b>	<b>0.0639**</b>	<b>0.0761**</b>	<b>0.0883**</b>	-
ENMF-I	<b>0.1888**</b>	<b>0.2667**</b>	<b>0.3534**</b>	<b>0.0634**</b>	<b>0.0759**</b>	<b>0.0884**</b>	-
ENMF-A	<b>0.1911**</b>	<b>0.2688**</b>	<b>0.3546**</b>	<b>0.0648**</b>	<b>0.0773**</b>	<b>0.0893**</b>	-
<i>MovieLens</i>	HR@50	HR@100	HR@200	NDCG@50	NDCG@100	NDCG@200	RI
MP	0.1842	0.2099	0.3382	0.0441	0.0481	0.0659	+109.01%
ItemKNN	0.2101	0.2889	0.3918	0.0598	0.0724	0.0867	+59.18%
BPR	0.2637	0.4048	0.5710	0.0757	0.0986	0.1217	+17.59%
WMF	0.2924	0.4378	0.6040	0.0909	0.1073	0.1324	+6.47%
ExpoMF	0.2904	0.4368	0.5927	0.0865	0.1100	0.1346	+7.11%
GMF	0.2847	0.4226	0.5847	0.0821	0.1086	0.1289	+10.30%
NCF	0.2902	0.4316	0.6023	0.0837	0.1097	0.1324	+8.02%
ConvNCF	0.2943	0.4403	0.6017	0.0872	0.1112	0.1333	+6.30%
ENMF-U	<b>0.3117**</b>	<b>0.4574**</b>	<b>0.6092**</b>	<b>0.0962**</b>	<b>0.1198**</b>	<b>0.1410**</b>	-
ENMF-I	<b>0.3105**</b>	<b>0.4576**</b>	<b>0.6107**</b>	<b>0.0956**</b>	<b>0.1194**</b>	<b>0.1398**</b>	-
ENMF-A	<b>0.3124**</b>	<b>0.4581**</b>	<b>0.6139**</b>	<b>0.0968**</b>	<b>0.1202**</b>	<b>0.1419**</b>	-

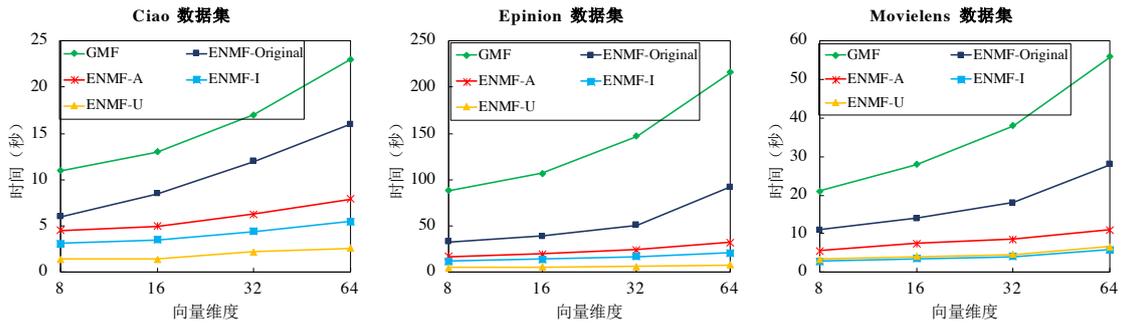


图 3.4 GMF, ENMF-Original, ENMF-U, ENMF-I 和 ENMF-A 在不同向量维度下的训练耗时

表 3.5 ENMF 与不同推荐模型的训练耗时情况, 其中 s/m/h 分别代表秒/分钟/小时

模型	Ciao			Epinion			Movielens		
	每轮时长	训练轮数	总训练时长	每轮时长	训练轮数	总训练时长	每轮时长	训练轮数	总训练时长
GMF	23s	300	115m	216s	500	30h	56s	500	7h
NCF	34s	300	170m	305s	500	42h	91s	500	12h
ConvNCF	88s	300	440m	510s	500	70h	246s	500	34h
ENMF-Original	16s	300	80m	65s	200	216m	28s	300	140m
ENMF-U	2.6s	300	13m	8s	200	27m	6.7s	300	34m
ENMF-I	5.5s	300	28m	21s	200	70m	5.8s	300	29m
ENMF-A	8s	150	20m	32s	100	53m	11s	50	9m

始非采样损失函数 (公式 (3.2)) 进行计算的方法。以上方法在不同向量维度  $d$  下每一轮的训练耗时如图3.4所示。从图中, 我们可以明显地观察到 ENMF-U、ENMF-I 和 ENMF-A 的训练耗时比相同向量维度下的 GMF 和 ENMF-Original 快很多。例如, 在 Epinion 数据集上, GMF 和 ENMF Original 分别需要 216 秒和 92 秒来训练  $d=64$  的模型, 而我们的方法 ENMF-U、ENMF-I 和 ENMF-A 分别只需要 8 秒、21 秒和 32 秒。由于这五种方法具有相同的神经网络结构, 我们可以将加速归因于所设计的高效非采样学习算法。与原始的非采样损失函数和负采样策略相比, 我们的方法在计算效率上更有优势。

进一步地, 我们比较了 ENMF 和三个有代表性的神经网络推荐模型 GMF, NCF, ConvNCF 的整体训练效率, 实验结果如表3.5所示。从表中可以看到, 本文所提出的高效非采样学习算法结合 ENMF 较其他推荐算法有数量级上的训练效率提升。以 Epinion 数据集为例, 基准模型完成训练需要 30 小时以上, 而我们的方法分别只需要 27 分钟、70 分钟和 53 分钟即可达到最佳性能, 加速超过 30 倍。在真实的应用场景中, 模型训练时间也是需要考虑的一个重要因素。我们提出的高效非采样方法在训练效率方面显示出明显的优势, 表明了算法具有很强的实用性。

我们也通过实验展示了神经网络模型 GMF、ConvNCF 和 ENMF 方法的学习

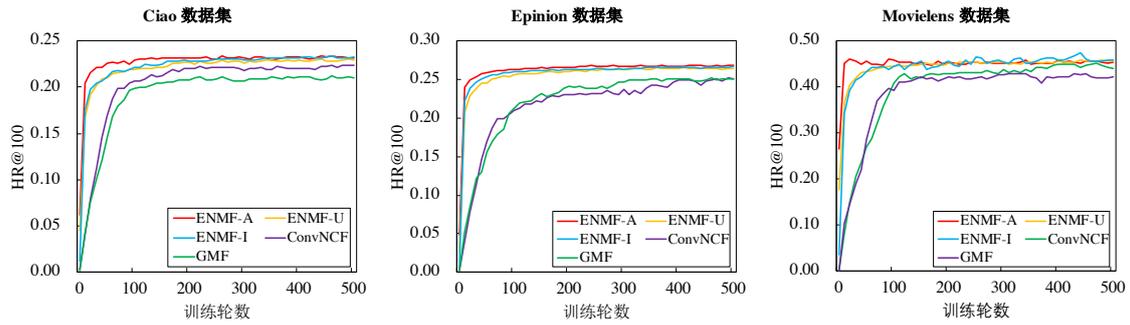


图 3.5 GMF, ConvNCF, ENMF-U, ENMF-I 和 ENMF-A 随训练轮数的收敛情况

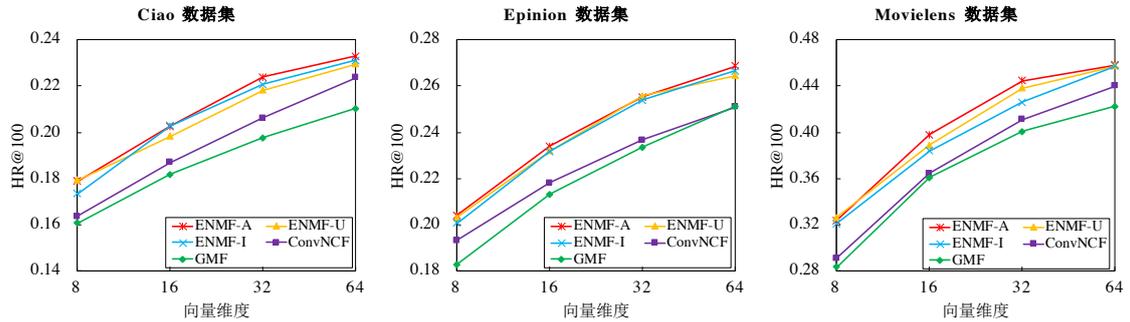


图 3.6 GMF, ConvNCF, ENMF-U, ENMF-I 和 ENMF-A 随向量维度变化的表现情况

收敛过程，实验结果如图3.5所示。为了方便，图3.5中以  $HR@100$  评价指标为例进行展示，其余评价指标上的结果与之类似。从图中，我们有如下观察：首先，我们的方法 ENMF-A、ENMF-U、ENMF-I 的收敛速度比 GMF 和 ConvNCF 快得多，并且始终保持更好的表现。原因在于 GMF 和 ConvNCF 基于负采样进行训练的，通常需要更多的训练轮数，并且很难将模型优化到最优的状态。其次，实验结果表明，ENMF-A 的收敛所需的轮数比 ENMF-U 和 ENMF-I 更少。这表明，通过采用交替学习，ENMF-A 可以缓解 ENMF-U 和 ENMF-I 造成的用户和物品训练不平衡的问题，从而进一步加速模型训练过程。

### 3.4.2.3 超参数影响分析

在本小节中，我们通过实验展示不同的超参数设置对于本文所提出的高效非采样学习方法结果的影响。具体来说，有 3 个超参数需要被重点关注，分别是：向量维度，负样本权重，以及网络留存率。

我们首先进行实验测试向量维度  $d$  对于结果的影响。图3.6显示了  $HR@100$  指标随向量维度变化的表现情况。对于其他指标，观察结果一致。从该图可以看出：我们的 ENMF 方法在不同向量维度设置下均优于其他模型。值得注意的是，向量维度为 32 的 ENMF 甚至比向量维度为 64 的 ConvNCF 模型表现效果更好。这进一步验证了我们的 ENMF 方法中基于非采样学习所带来的推荐性能优势。此外，随

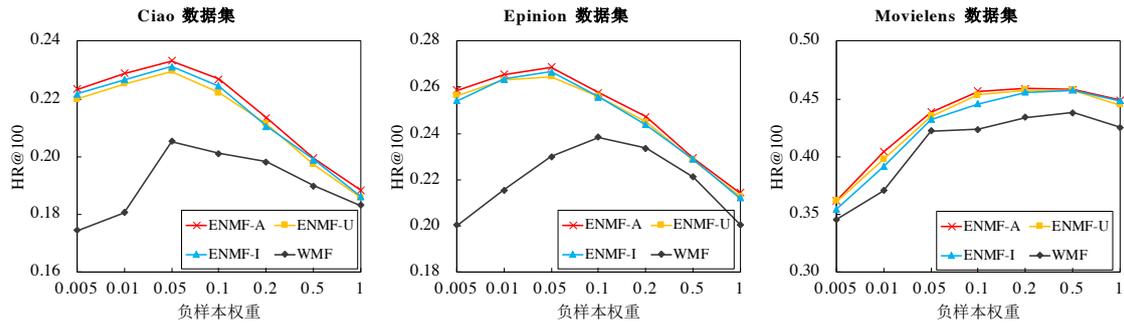


图 3.7 WMF, ENMF-U, ENMF-I 和 ENMF-A 随负样本权重变化的表现情况

随着向量维数的增加,所有模型的性能都会提高。这表明,更大的维度可以增加建模能力,从而学到更多用户和物品的隐含信息,提升推荐效果。这一观察结果也与之前的工作保持一致<sup>[12,20,39]</sup>。然而,对于大多数深度学习和神经网络方法,更大的向量维度也意味着需要更多的训练时间。因此,高效的神经网络学习算法对于提升推荐模型实际应用中的效率和效果意义重大。

接下来,我们进行实验测试负样本权重  $c_0$  对于基于非采样学习的推荐模型 WMF, ENMF-U, ENMF-I 和 ENMF-A 的效果影响,结果展示于图3.7中。值得一提的是,在本章中为了方便,我们将负样本权重设置为一个固定的常数,并且已经在之前的实验中显示了优于现有推荐方法的性能。在实际使用中,也可以将权重设置为跟物品流行度相关的数值来进一步提升模型的效果。据图,我们有如下的观察:首先,对于 Ciao 和 Epinion 数据集,当  $c_0$  约为 0.05 时,性能达到峰值,而对于 Movielens 数据集,模型表现最佳时的  $c_0$  约为 0.5。当  $c_0$  相对峰值更小或太大时,WMF 和我们的 ENMF 的性能都会有所降低。这突出了非采样学习策略中负样本权重的调节的重要性。其次,与 WMF 相比,我们的 ENMF 方法对负样本权重的选择更具鲁棒性。例如,在 Ciao 和 Epinion 数据集上,ENMF 的  $c_0$  在 0.005 到 0.2 之间时,始终优于 WMF 的最佳表现。最后,考虑到每个数据集的统计信息,我们发现负样本的最佳权重跟数据集的稠密度具有一定的相关性。Movielens 数据集相对密集(密度为 4.47%,而 Ciao 和 Epinion 分别为 0.19% 和 0.09%)。如之前的工作<sup>[8-9]</sup>所示,稀疏数据中大部分物品未被用户交互,更可能是因为用户未看到而不是用户对其不感兴趣。因此,为稀疏数据分配更小的负样本权重是更加合理的。

最后,我们进行实验测试了网络留存率  $\rho$  变化对于神经网络推荐模型 GMF, ConvNCF, ENMF-U, ENMF-I 和 ENMF-A 的效果影响,结果展示于图3.8中。图中  $\rho = 1$  时表示无去除操作并保留所有网络节点。从实验结果中可以看到:首先,通过调节网络留存率到最优的值,所有方法的效果都可以得到显著的改进。这说明了使用网络去除技术可以有效地防止神经网络模型过拟合并且提升推荐效果。其次,在不同的网络留存率设置下,我们的 ENMF 方法始终优于 GMF 和 ConvNCF。

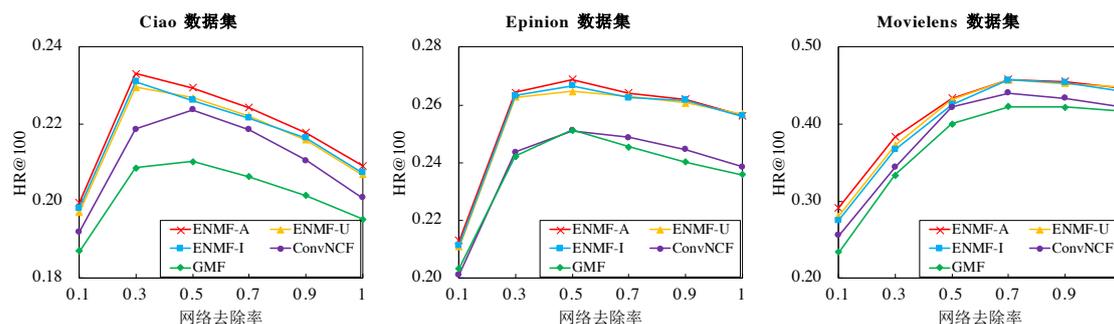


图 3.8 GMF, ConvNCF, ENMF-U, ENMF-I 和 ENMF-A 随网络留存率变化的表现情况

这再次展示非采样学习在模型优化方面的稳定性和优越性。最后，考虑到每个数据集的统计信息，我们发现最佳的网络留存率同样取决于数据集的大小和稠密程度。例如，Ciao 的结果相比于 Epinion 和 Movielens 的结果对网络留存率的设置更敏感。该发现是比较直观的，因为神经网络模型更容易在小而稀疏的数据集上过拟合，因此需要设置较小的留存率来提升神经网络模型的稳定性。

### 3.5 小结

本章研究了高效神经网络推荐系统中作为基础的“学习方法”问题。针对此问题，我们通过严谨的数学推理，对机器学习算法底层理论基础进行了创新，重构了目前常用的平方损失函数，新设计了一系列高效的、可用于神经网络训练的非采样学习方法。将从整体数据中学习的时间复杂度降低了一个数量级。极大地提高了深度网络的学习效率和训练的容易程度，突破了长期以来依赖“采样”的方式训练神经网络模型的瓶颈。成功使用“非采样”获得比传统“采样”方式更快更有效的模型参数学习，从而极大地提高了神经网络模型的实用性能。在三个数据集上的丰富实验和分析表明，我们提出的高效非采样学习算法以及实例化的 ENMF 推荐模型较已有方法在推荐效果和训练效率上都取得了显著提升。同时，这一研究是也是基础性的、开创性的。本质上，所有依赖负采样的机器学习任务和算法都可以使用这一解决方案，而这样的算法在机器学习领域随处可见<sup>[6,57-58]</sup>。更值得一提的是，与现有神经网络推荐系统研究往往聚焦于复杂和先进的模型结构不同，本章从“学习方法”的角度提出了一套新的研究思路来提升推荐效果，所设计的高效非采样算法填补了非采样神经网络推荐模型研究的空白，扩展了神经网络推荐系统研究的广度。本章内容也为接下来各章中所设计的高效神经网络推荐模型打下了技术基础。

本章相关成果发表于 CCF-A 类期刊 TOIS 2020<sup>[49]</sup> 和 TOIS 2022<sup>[50]</sup>。

## 第4章 结合异质交互行为的高效推荐建模

### 4.1 引言

上一章对于高效神经网络推荐系统中作为基础的“学习方法”进行了深入的探索。在本章中，我们将开始对“多场景高效神经网络推荐建模”问题进行研究。随着互联网的蓬勃发展，推荐系统已在不同场景下为海量用户提供智能信息获取服务以缓解信息过载问题<sup>[4,20]</sup>。在现有的推荐方法中，协同过滤类的方法因其简洁的模型设计和良好的性能，获得了产业界和学术界的极大关注<sup>[6,12,20,42,48]</sup>。然而，传统的协同过滤方法仅使用用户和物品的历史交互数据进行训练，因此在用户-物品交互较少的情况（即：冷启动或数据较为稀疏）下表现欠佳。

在推荐系统现实应用场景中，除了与预测目标直接相关的用户-物品交互数据外，也广泛存在着许多额外的异质交互行为数据，如用户之间的社交关系行为<sup>[12,108-110]</sup>以及用户对物品的多类型交互行为（如点击、加入购物车等在购买行为之外的辅助交互）<sup>[13]</sup>等。这些异质交互行为数据同样可以帮助提升推荐效果。然而，现有工作对于这些数据的精细化建模方面仍存在着不足，同时，当前学术界对于推荐系统模型的研究主要集中于对神经网络结构的探索，在模型训练效率方面却关注较少。

具体而言，在结合用户社交行为数据对推荐系统建模（又称为社交推荐）方面，基于用户对于不同领域之间的偏好具有一定程度的共性<sup>[30]</sup>这一假设，一些研究将预测社交领域的用户-用户交互视为预测用户-物品交互之外的另一个任务，通过挖掘用户对于社交好友的偏好来增强其对物品偏好的学习<sup>[30,79-80]</sup>。这一类工作经常使用到的技术是迁移学习<sup>[81-82,111-112]</sup>，其核心思想是将共有知识从源领域迁移到目标领域。如前所述，社交行为中同样包含用户的偏好信息，这就为迁移学习提供了基础。然而，现有的迁移学习方法通常是静态的<sup>[30,80-81]</sup>，不同领域之间的知识共享程度无法针对不同的用户动态改变。在现实场景中，两个领域的信息丰富程度及偏好共享程度通常因不同用户而异。因此，为了更好地从用户的社交行为中学习其对于物品的偏好，需要设计针对不同用户的自适应迁移学习方案。针对这一问题，本章提出了基于自适应迁移学习的高效社交推荐模型 EATNN (Efficient Adaptive Transfer Neural Network)，通过引入注意力机制，自适应地学习不同用户在社交领域和物品领域之间偏好的转移权重。进一步地，拓展上一章所设计的高效非采样算法来联合学习用户社交行为和对于物品的兴趣偏好，实现社交行为的充分利用。在现实数据集上的实验表明，所提出的 EATNN 模型相对于已有社交推

荐模型在精准度和训练效率上均有显著提升，进一步的实验也验证了 EATNN 可以有效缓解数据稀疏和冷启动问题。

在结合多类型交互行为数据对推荐系统建模（又称为多行为推荐）方面，现有研究中一类方法将对不同类型交互数据的刻画建模到对目标行为（如购买）的预测任务中<sup>[83-84]</sup>，另一类则是对每个行为进行分别预测，并结合多任务学习实现参数共享<sup>[13,80]</sup>。然而，这两类现有方法均存在一定的局限性。第一类方法仅优化一个预测任务，模型表达能力有限；第二类方法目前通常使用负采样策略进行优化，而负采样在多任务形式下其不稳定性会成倍增加，且某些行为的样本会被同时视为正样本和负样本，造成了优化矛盾，使得模型整体更难收敛到最优的状态。此外，现有方法未能建模不同行为之间的因果关系，缺乏对于行为之间关系的显式刻画。针对以上问题，本章提出一个基于因果推理的高效多行为协同过滤推荐模型 EHCF（Efficient Heterogeneous Collaborative Filtering），将用户多类型交互的关系以因果图的方式进行建模，并拓展上一章所设计的高效非采样算法到多任务学习框架下对模型进行统一优化。进一步地，考虑到图卷积神经网络<sup>[113]</sup>在对高阶关系学习任务上的良好表现，本章也提出了一个基于异质图神经网络的高效多行为推荐模型 GHCF（Graph Heterogeneous Collaborative Filtering）。其中新设计的用户-物品的表示向量传播层（Embedding Propagation Layer）可以高效地学习用户、物品、以及交互类型的高阶向量表示，从而提升推荐效果。在现实数据集上的实验表明，所提出的 EHCF 和 GHCF 模型相对于已有多行为推荐模型在精准度和训练效率上均有显著提升，进一步的实验也验证了它们在数据稀疏和冷启动情况下的有效性。

本章的结构如下：第4.2节介绍相关工作；第4.3节介绍结合社交行为的高效推荐建模问题、所设计的 EATNN 模型以及相关实验；第4.4节介绍结合多类型交互行为的高效推荐建模问题、所设计的 EHCF、GHCF 模型以及相关实验；最后，第4.5节对本章进行小结。

## 4.2 相关工作

### 4.2.1 结合社交行为的推荐

社交行为广泛存在于当前互联网平台上，对于社交行为的建模有助于提升推荐系统的精准度。现有的结合社交行为的推荐系统研究<sup>[12,26,30]</sup>主要基于以下两种假设：（1）用户和其社交好友的偏好是相似的，即用户兴趣受到社交好友的影响<sup>[76]</sup>；（2）用户对于社交领域中好友的偏好与其对于物品交互领域中的物品偏好是相似的，即用户对于不同领域之间的偏好具有一定程度的共性<sup>[30]</sup>。基于第一种假设，一些研究工作<sup>[76-78,114]</sup>在用户的偏好学习过程中引入来自于社交好友的相似

性约束,通过建模社交好友的影响来提升推荐系统的效果。近年来,神经网络的发展也使得动态建模社交影响成为可能,例如,Chen et al.<sup>[12]</sup>通过结合注意力机制和记忆网络对用户社交关系中的好友影响力差异及共同偏好差异进行建模,Sun et al.<sup>[115]</sup>引入循环神经网络来建模包含时态信息和社交行为的推荐场景。基于第二种假设,一些研究将预测社交领域的用户—用户关系视为预测用户—物品交互之外的另一个任务,通过挖掘用户对于社交好友的偏好来增强其对物品偏好的学习<sup>[30,79-80]</sup>。例如,Singh et al.<sup>[80]</sup>提出通过迁移学习来共享对社交预测和物品偏好预测的参数,Xiao et al.<sup>[30]</sup>提出在共享参数的同时考虑用户社交好友和物品的可见度(Visibility)对于权重的影响。然而,现有的基于迁移学习的方法通常是静态的,不同领域之间的知识共享程度无法针对不同的用户动态改变。为了更好地从用户的社交行为中学习其对于物品的偏好,需要设计针对不同用户的自适应迁移学习方案。综上所述,现有方法对于结合社交行为的推荐仍停留在较为粗犷的地步,缺乏对建模社交行为的精细化研究。

#### 4.2.2 结合多类型交互行为的推荐

对于在推荐系统中多类型交互数据的使用,现有的研究可以被分为两类。第一类将对不同类型交互数据的刻画建模到对目标行为(如购买)的预测任务中<sup>[83-84]</sup>。这一类方法的核心思想是不同类型的交互行为反应了用户对物品不同程度的兴趣偏好,对于目标行为的学习起到的作用有强弱之分。例如,Loni et al.<sup>[84]</sup>提出在贝叶斯个性化排序(BPR)<sup>[6]</sup>的基础上,通过考虑不同行为来改变负采样方式。这一类方法的不足是仅优化一个预测任务,模型表达能力有限。第二类是基于多任务学习的方式,对不同的交互行为分别进行预测,并通过参数共享协同学习<sup>[13,80,116-117]</sup>。例如,Gao et al.<sup>[13]</sup>在神经协同过滤模型(NCF)<sup>[20]</sup>的基础上,通过多任务学习实现多类型交互数据的共同利用与学习。这一类方法的不足在于通常是基于负采样技术进行优化,而负采样策略在多任务形式下其不稳定性会成倍增加,且某些行为的样本会被同时视为正样本和负样本,造成了优化矛盾,使得模型整体更难收敛到最优的状态。此外,现有方法对于结合多类型交互行为的推荐未能建模不同行为之间的因果推断关系,缺乏对于不同行为之间关系的显式刻画。另一方面,虽然近年来图卷积神经网络<sup>[113]</sup>在对高阶关系学习任务上取得了良好的表现,但现有方法或者是只对一种交互行为进行建模<sup>[41-42,118]</sup>,或者是只能学习节点的表示而无法学习关系边的表示<sup>[119-120]</sup>。这些局限性限制了当前图神经网络在多行为推荐任务上的适用性。综上所述,现有方法对于结合多类型交互行为的推荐在学习方式、模型结构等方面仍存在有明显缺陷。

表 4.1 结合社交行为的高效推荐建模中所用数学符号

符号	描述
$\mathbf{U}, \mathbf{V}$	分别代表用户集合和物品集合
$\mathbf{B}$	一批训练用户
$\mathbf{Y}$	用户—物品交互
$\mathcal{Y}$	正向用户—物品交互
$\mathbf{X}$	用户—用户社交交互
$\mathcal{X}$	正向用户—用户交互
$\mathbf{u}^I$	用户 $u$ 在物品 (Item) 领域的表示向量
$\mathbf{u}^S$	用户 $u$ 在社交 (Social) 领域的表示向量
$\mathbf{u}^C$	用户 $u$ 在物品和社交领域的共有 (Common) 表示向量
$\mathbf{p}_u^I$	迁移共有知识后用户 $u$ 在物品领域的表示向量
$\mathbf{p}_u^S$	迁移共有知识后用户 $u$ 在社交领域的表示向量
$\mathbf{q}_v$	物品 $v$ 的表示向量
$\mathbf{g}_t$	用户 $t$ 在社交领域作为被交互对象时的表示向量
$c_{uv}^I$	训练样例 $y_{uv}$ 的权重
$c_{ut}^S$	训练样例 $x_{ut}$ 的权重
$\alpha_{(I,u)}$	表示向量 $\mathbf{u}^I$ 在物品领域的注意力权重
$\alpha_{(C,u)}$	表示向量 $\mathbf{u}^C$ 在物品领域的注意力权重
$\beta_{(S,u)}$	表示向量 $\mathbf{u}^S$ 在社交领域的注意力权重
$\beta_{(C,u)}$	表示向量 $\mathbf{u}^C$ 在社交领域的注意力权重
$d$	向量维度
$\Theta$	神经网络参数集合

### 4.3 结合社交行为的高效推荐建模

在本节对于结合社交行为的高效推荐建模进行阐述，包括问题定义、解决方案、以及相关实验。

#### 4.3.1 问题定义

表4.1中展示了本节中用到的符号和关键概念。假设数据集中有  $M$  个用户和  $N$  个物品，用户集合记为  $\mathbf{U}$ ，物品集合记为  $\mathbf{V}$ 。我们使用索引  $u$  表示用户，使用  $v$  表示物品。用户—物品交互矩阵记为  $\mathbf{Y} = [y_{uv}]_{M \times N} \in \{0, 1\}$ ，表示  $u$  是否与物品  $v$  有过交互。在社交信息方面，用户—用户交互矩阵记为  $\mathbf{X} = [x_{ut}]_{M \times M} \in \{0, 1\}$ ，表

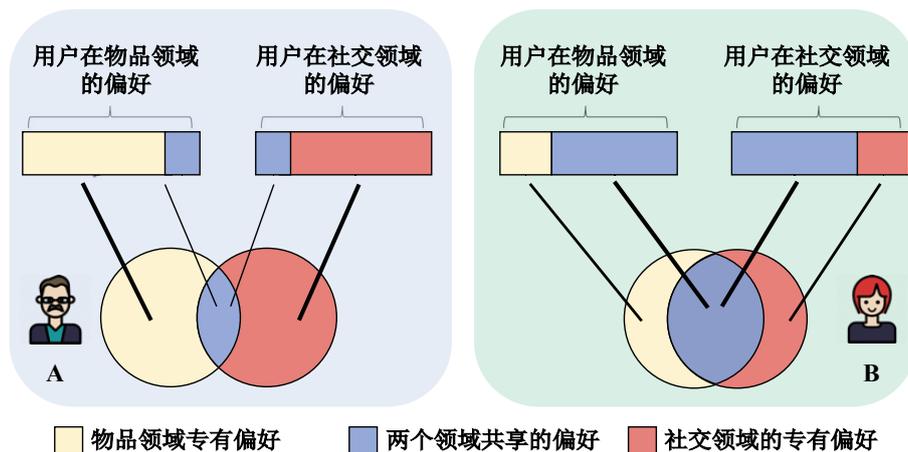


图 4.1 用户在社交领域和物品领域的共有偏好案例示意图

示用户  $u$  是否与用户  $t$  有社交行为。在此设置下，结合社交行为的推荐任务定义如下：

**输入：**用户集  $U$ ，物品集  $V$ ，用户—物品交互记录  $Y$ ，和用户—用户社交记录  $X$ 。

**输出：**给定一个用户，为其给出按照与物品交互概率排序得到的物品推荐列表。

### 4.3.2 基于自适应迁移学习的高效社交推荐模型

#### 4.3.2.1 模型概览

图4.1展示了本节设计自适应迁移学习方法的动机。如图中用户 A 和 B 所示，在现实场景中，两个领域的信息丰富程度及偏好共享程度通常因不同用户而异。因此，为了更好地从用户的社交行为中学习其对于物品的偏好，需要设计可以针对不同用户的自适应迁移学习方法。

本节我们将介绍所提出的基于自适应迁移学习的高效社交推荐模型 EATNN，模型的示意图如图4.2所示。首先对模型整体做一简要介绍：

1. EATNN 模型的主要目标是在效率和效果两方面提升结合社交行为的推荐系统。
2. 通过引入注意力机制，模型可以自动为每个用户分配合适的迁移学习权重，以更加精准地实现不同领域之间共享知识的迁移。
3. 在模型学习上，进一步拓展上一章所设计的高效非采样算法（基于用户的）来联合学习用户社交行为和对于物品的兴趣偏好，以实现社交行为的充分利用。

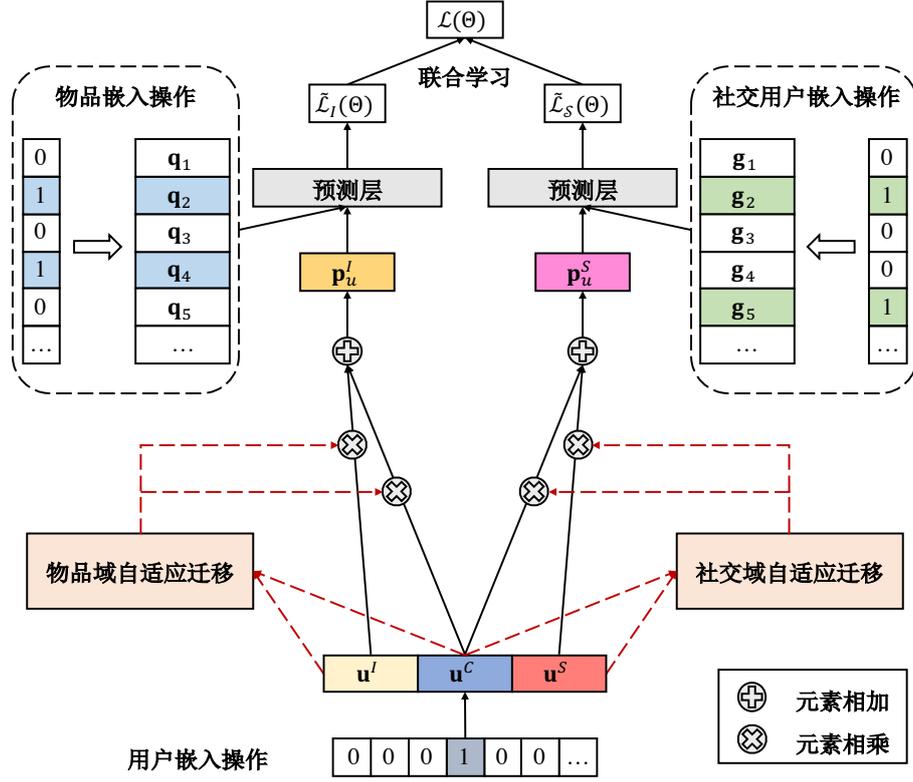


图 4.2 基于自适应迁移学习的高效社交推荐模型 (EATNN) 示意图

#### 4.3.2.2 基于注意力机制的自适应迁移

用户、物品，以及用户在社交领域作为被交互对象的用户首先通过嵌入操作将其转化为稠密的向量表示。用户作为连接物品领域和社交领域的桥梁，每个用户  $u$  包含三个表示向量：(1) 在物品和社交领域的共有表示向量  $\mathbf{u}^C$ ，代表用户对社交好友及物品的共有偏好；(2) 物品领域的表示向量  $\mathbf{u}^I$ ，代表用户对于物品独有的偏好；(3) 社交领域的表示向量  $\mathbf{u}^S$ ，代表用户对社交好友独有的偏好。

为了实现对于每个用户的自适应迁移学习，我们在模型中引入注意力机制<sup>[121]</sup>。具体地，注意力网络分别被应用于物品域和社交域，其中每个模块都是两层的神经网络。对于用户  $u$  来说，如果其在这两个领域之间的共有偏好较少，则在迁移学习中  $u$  的共有表示向量  $\mathbf{u}^C$  所占的权重也应较小，模型将使用更多来自特定领域的信息 ( $\mathbf{u}^I$  或  $\mathbf{u}^S$ )。形式上，物品域注意力网络和社交域注意力网络的计算如下：

$$\begin{aligned} \alpha_{(C,u)}^* &= \mathbf{h}_\alpha^T \sigma(\mathbf{W}_\alpha \mathbf{u}^C + \mathbf{b}_\alpha); & \alpha_{(I,u)}^* &= \mathbf{h}_\alpha^T \sigma(\mathbf{W}_\alpha \mathbf{u}^I + \mathbf{b}_\alpha) \\ \beta_{(C,u)}^* &= \mathbf{h}_\beta^T \sigma(\mathbf{W}_\beta \mathbf{u}^C + \mathbf{b}_\beta); & \beta_{(S,u)}^* &= \mathbf{h}_\beta^T \sigma(\mathbf{W}_\beta \mathbf{u}^S + \mathbf{b}_\beta) \end{aligned} \quad (4.1)$$

其中  $\mathbf{W}_\alpha \in \mathbb{R}^{k \times d}$ ， $\mathbf{b}_\alpha \in \mathbb{R}^k$ ，和  $\mathbf{h}_\alpha \in \mathbb{R}^k$  是物品域注意力网络的参数； $\mathbf{W}_\beta \in \mathbb{R}^{k \times d}$ ， $\mathbf{b}_\beta \in \mathbb{R}^k$ ， $\mathbf{h}_\beta \in \mathbb{R}^k$  是社交域注意力网络的参数； $d$  是表示向量的维度； $k$  是注意力

网络向量维度； $\sigma$  代表非线性激活函数  $ReLU$  [122]。

然后，注意力的最终权重通过 softmax 函数归一化得到：

$$\begin{aligned}\alpha_{(C,u)} &= \frac{\exp(\alpha_{(C,u)}^*)}{\exp(\alpha_{(C,u)}^*) + \exp(\alpha_{(I,u)}^*)} = 1 - \alpha_{(I,u)} \\ \beta_{(C,u)} &= \frac{\exp(\beta_{(C,u)}^*)}{\exp(\beta_{(C,u)}^*) + \exp(\beta_{(I,u)}^*)} = 1 - \beta_{(S,u)}\end{aligned}\quad (4.2)$$

其中  $\alpha_{(C,u)}$  和  $\beta_{(C,u)}$  分别表示共有表示向量  $\mathbf{u}^C$  对物品域和社交域的迁移权重，代表了有多少领域间共有知识可以被迁移到特定的领域中。迁移后对于物品域和社交域用户  $u$  的表示向量计算如下：

$$\mathbf{p}_u^I = \alpha_{(I,u)}\mathbf{u}^I + \alpha_{(C,u)}\mathbf{u}^C; \quad \mathbf{p}_u^S = \beta_{(S,u)}\mathbf{u}^S + \beta_{(C,u)}\mathbf{u}^C \quad (4.3)$$

其中  $\mathbf{p}_u^I$  和  $\mathbf{p}_u^S$  分别代表在迁移完共有知识后，用户在物品域和社交域的向量表示。

#### 4.3.2.3 模型预测

基于所学到的表示向量，模型的预测部分旨在生成用户对物品以及对社交好友偏好的概率分数。本模型的预测部分是基于一个神经网络的矩阵分解框架 [20]，对于每个预测任务使用不同的预测层。具体地，用户  $u$  对于物品  $v$  的预测分数  $\hat{y}_{uv}$  以及  $u$  对于社交好友  $t$  的预测分数  $\hat{x}_{ut}$  计算如下：

$$\hat{y}_{uv} = \mathbf{h}_I^T(\mathbf{p}_u^I \odot \mathbf{q}_v); \quad \hat{x}_{ut} = \mathbf{h}_S^T(\mathbf{p}_u^S \odot \mathbf{g}_t) \quad (4.4)$$

其中  $\mathbf{q}_v \in \mathbb{R}^d$  和  $\mathbf{g}_t \in \mathbb{R}^d$  分别代表物品  $v$  以及用户  $t$  作为被交互对象时的向量表示； $\odot$  符号表示哈达玛积，即对于两个向量进行逐项相乘； $\mathbf{h}_I \in \mathbb{R}^d$  和  $\mathbf{h}_S \in \mathbb{R}^d$  分别表示对于物品和对于社交好友的预测层，为向量空间的不同维度分配相应的权重。注意我们这里对于用户作为被交互对象时的向量表示采用了与其偏好向量不同的策略，这是为了引入更多参数，也可以使得模型的学习过程更加灵活和稳定。

#### 4.3.2.4 高效联合学习

在上一章的工作中，我们已经提出了针对双塔推荐模型的高效非采样学习算法，这里不再赘述其具体推导过程。将定理3.1应用于结合社交关系的推荐模型中，则对于物品领域的预测损失函数为：

$$\begin{aligned}\tilde{\mathcal{L}}_I(\Theta) &= \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}^+} ((c_v^{I^+} - c_v^{I^-})\hat{y}_{uv}^2 - 2\hat{y}_{uv}) \\ &+ \sum_{i=1}^d \sum_{j=1}^d \left( (h_{I,i}h_{I,j}) \left( \sum_{u \in \mathbf{B}} p_{u,i}^I p_{u,j}^I \right) \left( \sum_{v \in \mathbf{V}} c_v^{I^-} q_{v,i} q_{v,j} \right) \right)\end{aligned}\quad (4.5)$$

其中  $c_v^I$  表示训练样本  $y_{uv}$  的权重。

同样地，对于社交领域的预测损失函数为：

$$\begin{aligned} \tilde{\mathcal{L}}_S(\Theta) = & \sum_{u \in \mathbf{B}} \sum_{t \in \mathbf{U}^+} ((c_t^{S^+} - c_t^{S^-}) \hat{x}_{ut}^2 - 2\hat{x}_{ut}) \\ & + \sum_{i=1}^d \sum_{j=1}^d \left( (h_{S,i} h_{S,j}) \left( \sum_{u \in \mathbf{B}} p_{u,i}^S p_{u,j}^S \right) \left( \sum_{t \in \mathbf{U}} c_t^{S^-} g_{t,i} g_{t,j} \right) \right) \end{aligned} \quad (4.6)$$

其中  $c_t^S$  表示训练样本  $x_{ut}$  的权重。

在分别得到对物品和社交领域的预测损失函数后，我们将这两个任务同时进行训练，构成了端到端的联合学习框架，具体公式如下：

$$\mathcal{L}(\Theta) = \tilde{\mathcal{L}}_I(\Theta) + \mu \tilde{\mathcal{L}}_S(\Theta) + \lambda \|\Theta\|_2^2 \quad (4.7)$$

其中超参数  $\mu$  用来控制两个任务的训练比重， $L_2$  正则化项则被用于限制模型参数  $\Theta$ ，避免出现过拟合现象。

此外，为了更好地训练模型，我们采用 Adagrad<sup>[105]</sup> 作为模型的优化器，它的主要优势在于会自适应地调整训练过程中的学习率，选择合理的步长来对解空间进行搜索，从而避免了对于学习率这个超参数过多的调参过程。去除函数 (Dropout) 也被加入到模型中用来避免训练出现过拟合现象<sup>[123]</sup>，我们的随机去除主要是针对作为迁移学习桥梁的用户表示  $\mathbf{p}_u^I$  和  $\mathbf{p}_u^S$  进行的，去除操作之后，保留比例为  $\rho$  的向量参数。

#### 4.3.2.5 讨论

对 EATNN 模型的计算复杂度进行分析，由于 EATNN 在训练过程中同时对物品交互和社交关系进行优化，因此复杂度也将分为两部分。首先对于物品交互部分 (公式 (4.5))，每一批训练数据的计算复杂度为  $O((|\mathbf{B}| + |\mathbf{V}|)d^2 + |\mathcal{Y}_{\mathbf{B}}|d)$  (注意力机制的计算复杂度相对要小很多，因此可以忽略不计)，其中  $\mathcal{Y}_{\mathbf{B}}$  表示每批训练数据中实际存在的用户—物品正向交互。同理，对于社交关系部分 (公式 (4.6))，每一批训练数据的计算复杂度为  $O((|\mathbf{B}| + |\mathbf{U}|)d^2 + |\mathcal{X}_{\mathbf{B}}|d)$ ，其中  $\mathcal{X}_{\mathbf{B}}$  表示每批训练数据中实际存在的用户—用户正向交互。两部分相加即得整体的计算复杂度为  $O((2|\mathbf{B}| + |\mathbf{U}| + |\mathbf{V}|)d^2 + (|\mathcal{Y}_{\mathbf{B}}| + |\mathcal{X}_{\mathbf{B}}|)d)$ 。如果不使用高效非采样这两个任务下传统非采样方法的复杂度为  $O((|\mathbf{B}||\mathbf{V}| + |\mathbf{B}||\mathbf{U}|)d)$ ，考虑到现实场景中  $|\mathcal{Y}_{\mathbf{B}}| \ll |\mathbf{B}||\mathbf{V}|$ ， $|\mathcal{X}_{\mathbf{B}}| \ll |\mathbf{B}||\mathbf{U}|$ ，且  $d \ll |\mathbf{B}|$ ，优化后的 EATNN 较传统的非采样训练方法有了数量级的效率提升。

表 4.2 评估 EATNN 模型的数据集统计信息

数据集	用户数量	物品数量	用户—物品交互数量	用户—用户社交行为数量
<i>Ciao</i>	7,267	11,211	157,995	111,781
<i>Epinion</i>	20,608	23,585	454,002	351,486
<i>Flixster</i>	69,251	17,318	7,940,096	967,195

### 4.3.3 实验

本实验所用代码已开源<sup>①</sup>。

#### 4.3.3.1 数据集介绍

我们选择三个被广泛使用的推荐系统公开数据集进行实验，分别是 *Ciao*<sup>②</sup>，*Epinion*<sup>③</sup> 和 *Flixster*<sup>④</sup>。对三个数据集的简单介绍如下：

- *Ciao*：此数据集包含用户对所购买物品的评分以及用户的社交好友关系。由于本文重点关注隐式数据，因此具体的评分被转换为 0 或 1 的值，表示用户是否对某个物品进行了评分。
- *Epinions*：Epinions 是一个广泛使用的推荐数据集。它同样收集了用户对于对所购买物品的评分及用户之间的社交行为。在我们的实验中，相应的评分也被转换为隐式数据。
- *Flixster*：Flixster 是我们所用最大的一个数据集，其中包含有超过 700 万的用户—物品交互记录以及约 100 万的社交关系。

三个数据集的基本统计信息见表4.2。

#### 4.3.3.2 基准方法

为了验证所提出的 EATNN 模型的有效性，我们与现有的推荐算法进行了比较，按照是否可以利用社交行为信息将这些方法分为两类。其中只使用用户—物品交互数据的推荐模型包括：

- **BPR**<sup>[6]</sup>：贝叶斯个性化排序，通过优化正负样本的偏序损失来训练矩阵分解类的推荐方法。
- **ExpoMF**<sup>[8]</sup>：同样基于矩阵分解和非采样学习的推荐模型，其中训练样例权重通过对物品的曝光（Exposure）概率进行建模。

① <https://github.com/chenchongthu/EATNN>

② <http://www.jiliang.xyz/trust.html>

③ <https://alchemy.cs.washington.edu/data/epinions/>

④ <http://www.cs.ubc.ca/jamalim/datasets/>

表 4.3 EATNN 与对比方法的特点比较

模型特点	BPR	ExpoMF	NCF	SBPR	TranSIV	SAMN	EATNN
用户—物品交互	√	√	√	√	√	√	√
用户—用户社交关系	\	\	\	√	√	√	√
神经网络模型	\	\	√	\	\	√	√
自适应迁移学习	\	\	\	\	\	\	√
非采样学习	\	√	\	\	√	\	√

- **NCF**<sup>[20]</sup>: 神经网络协同过滤模型, 是一个有代表性的基于神经网络的推荐模型, 通过将矩阵分解与多层感知机相结合对隐式反馈数据进行建模。同时使用用户—物品交互数据以及用户—用户社交关系的推荐模型包括:
- **SBPR**<sup>[26]</sup>: 基于社交关系的贝叶斯个性化排序 (Social Bayesian Personalized Ranking) 模型, 假设用户更倾向于对好友交互过的物品进行交互操作。
- **TranSIV**<sup>[30]</sup>: 基于物品与社交可见度的迁移学习模型 (Transfer Model with Social and Item Visibilities)。是非神经网络模型中表现最好的方法之一, 通过静态迁移学习和非采样学习建模社交推荐问题, 并且在共享参数时考虑了物品与社交关系可见度的影响。
- **SAMN**<sup>[12]</sup>: 社交注意力记忆模型 (Social Attentional Memory Network)。是已有工作中表现最好的结合社交关系的推荐方法之一, 通过注意力网络和记忆网络来建模好友影响以及共有偏好的影响。

#### 4.3.3.3 评价方式

对每个数据集, 我们随机选择 80% 的交互历史数据来构建训练集, 10% 作为验证集用于调整超参数, 最后 10% 用来做测试集。在模型评价过程中, 我们将使用不同算法为每个用户对所有的候选物品进行排序 (除了训练集和验证集已知的有正向交互的物品)。为了评价前  $K$  个推荐结果的表现,  $\text{Recall@K}$  和  $\text{NDCG@K}$  将作为主要的评价指标, 其中  $\text{Recall@K}$  关注于用户交互过的正例是否被排在前  $K$  项中, 而  $\text{NDCG@K}$  则会细粒度的考虑前  $K$  项中正例交互的具体位置, 这两个指标都是越高越好。  $\text{Recall@K}$  的计算如下所示:

$$\text{Recall@K} = \frac{\sum_{j=1}^K \text{rel}_j}{\min(K, |\mathbf{Y}_u^{\text{test}}|)} \quad (4.8)$$

其中  $\text{rel}_j = 1/0$  表示所排序的第  $j$  项是否在测试集中,  $|\mathbf{Y}_u^{\text{test}}|$  表示用户  $u$  测试集中的物品。  $\text{NDCG@K}$  的计算方式与上章相同。

#### 4.3.3.4 超参数设置

所有基准算法的参数设定都是根据对应论文的初始化要求设置的，且也都在验证集上进行了调参以确保其达到最优的效果。我们对超参数的主要搜索空间包括：学习率的搜索范围为 [0.005, 0.01, 0.02, 0.05]，为了避免过拟合，网络留存率 (Dropout) 的搜索范围为 [0.0, 0.1, ..., 0.9]，特征向量长度  $d$  的搜索范围为 [8, 16, 32, 64]，训练样本权重  $c_0$  的搜索范围为 [0.005, 0.01, 0.05, 0.1, 0.2, 0.5, 1]。最终，根据预实验结果，每个批的大小被设置成 512，学习率被设置成 0.05，特征向量的长度被设置成 64，注意力网络向量长度设为 32。对于我们的 EATNN 方法，网络留存率  $\rho$  对于 Ciao 和 Epinion 设置为 0.3，对于 Flixster 设置为 0.7。对于训练样本权重，正样本权重设为 1，物品域和社交域的负样本权重在三个数据集上的设置如下，Ciao: 0.1 和 0.1；Epinion: 0.05 和 0.1；Flixster: 0.15 和 0.1。此外，在联合学习过程中，超参数  $\mu$  被设为 0.1。

#### 4.3.3.5 整体表现分析

本文所提出的 EATNN 方法与其他基准方法在三个数据集上的推荐表现见表4.4，为了对比不同长度推荐列表的效果，我们汇报了推荐列表长度为 10、50 和 100 的结果。从实验结果中我们有以下发现：

首先，结合社会行为的推荐方法通常比只使用用户—物品交互的方法表现更好。例如，在表4.4中，SBPR 的性能优于 BPR，TranSIV、SAMN、和 EATNN 的性能优于 BPR、ExpoMF 和 NCF。这与之前的工作结论一致<sup>[12,26,30]</sup>，这表明社交关系也在一定程度上反映了用户的兴趣，因此可以帮助提升推荐效果。

其次，我们的方法 EATNN 在这三个数据集上均表现出最好的效果，并且显著优于所有基线方法（包括神经网络模型 NCF 和 SAMN），其  $p$ -值小于 0.01。举例来说，与最近提出的深度学习模型 SAMN 相比，EATNN 在三个数据集上的平均提升分别为 6.97%、7.62% 和 3.51%。我们的模型相比于基线方法的提升可以归因于两个原因：（1）我们的模型使用注意机制在物品域和社交域之间自适应地迁移共有知识，这可以更加精细地建模社交信息；（2）我们的模型的参数是使用非采样学习在整个数据上联合优化的，相比于基于负采样的方法（BPR、NCF、SAMN）更容易使模型优化到最优的状态。

最后，观察模型在每个数据集的表现，我们发现 EATNN 的相比于基线方法的效果提升与物品域数据的稀疏性相关。如 Flixster 数据集在用户-物目交互上相对稠密（每个用户平均 114.66 次交互，而 Ciao 和 Epinions 分别为 21.74 次和 22.03 次）。用户偏好虽然难以从稀疏数据（冷启动）中学习，但可以通过从社交领域学习

表 4.4 EATNN 与不同推荐方法的效果比较, \*\* 表示较最好的基准方法有显著提升, 显著性是  $p < 0.01$ 。“RI”表示 EATNN 相较对应方法的相对提升

<i>Ciao</i>	Recall@10	Recall@50	Recall@100	NDCG@10	NDCG@50	NDCG@100	RI
<b>BPR</b>	0.0591	0.1600	0.2135	0.0409	0.0688	0.0805	+20.08%
<b>ExpoMF</b>	0.0642	0.1556	0.2050	0.0445	0.0706	0.0816	+17.03%
<b>NCF</b>	0.0667	0.1584	0.2141	0.0456	0.0718	0.0837	+13.84%
<b>SBPR</b>	0.0623	0.1631	0.2146	0.0436	0.0695	0.0832	+16.30%
<b>TranSIV</b>	0.0678	0.1651	0.2184	0.0473	0.0753	0.0865	+10.20%
<b>SAMN</b>	<u>0.0719</u>	<u>0.1671</u>	<u>0.2233</u>	<u>0.0495</u>	<u>0.0768</u>	<u>0.0883</u>	+6.97%
<b>EATNN</b>	<b>0.0778**</b>	<b>0.1764**</b>	<b>0.2305**</b>	<b>0.0547**</b>	<b>0.0824**</b>	<b>0.0943**</b>	–
<i>Epinion</i>	Recall@10	Recall@50	Recall@100	NDCG@10	NDCG@50	NDCG@100	RI
<b>BPR</b>	0.0528	0.1477	0.2115	0.0353	0.0613	0.0751	+21.49%
<b>ExpoMF</b>	0.0611	0.1508	0.2077	0.0422	0.0673	0.0798	+11.82%
<b>NCF</b>	0.0535	0.1489	0.2144	0.0367	0.0624	0.0772	+19.06%
<b>SBPR</b>	0.0547	0.1511	0.2142	0.0387	0.0665	0.0783	+15.71%
<b>TranSIV</b>	<u>0.0631</u>	0.1552	0.2227	<u>0.0423</u>	0.0681	0.0829	+8.49%
<b>SAMN</b>	0.0621	<u>0.1583</u>	<u>0.2274</u>	0.0417	<u>0.0698</u>	<u>0.0842</u>	+7.62%
<b>EATNN</b>	<b>0.0696**</b>	<b>0.1675**</b>	<b>0.2309**</b>	<b>0.0474**</b>	<b>0.0749**</b>	<b>0.0887**</b>	–
<i>Flixster</i>	Recall@10	Recall@50	Recall@100	NDCG@10	NDCG@50	NDCG@100	RI
<b>BPR</b>	0.1733	0.3945	0.5272	0.1612	0.2193	0.2568	+35.88%
<b>ExpoMF</b>	0.2596	0.4488	0.5659	0.2012	0.2633	0.3002	+10.94%
<b>NCF</b>	0.2613	0.4564	0.5632	0.2112	0.2687	0.3075	+8.81%
<b>SBPR</b>	0.2314	0.4517	0.5697	0.1989	0.2514	0.3016	+14.05%
<b>TranSIV</b>	0.2748	0.4633	<u>0.5749</u>	0.2277	0.2804	0.3224	+4.35%
<b>SAMN</b>	<u>0.2767</u>	<u>0.4661</u>	0.5746	<u>0.2316</u>	<u>0.2833</u>	<u>0.3251</u>	+3.51%
<b>EATNN</b>	<b>0.2948**</b>	<b>0.4736**</b>	<b>0.5896**</b>	<b>0.2401**</b>	<b>0.2962**</b>	<b>0.3319**</b>	–

的知识来迁移。结果表明, 基于迁移学习的模型在稀疏数据集上带来的提升更多。为了进一步验证上述发现, 我们在接下来的第4.3.3.7小节进行了更进一步的实验。

#### 4.3.3.6 训练效率分析

在本小节中, 我们进行实验探索所设计的 EATNN 模型与基线模型中表现较好的 TranSIV 和 SAMN 的训练效率。所有的实验都是在同一台服务器上运行的, 其配置为: 英特尔 Xeon9 核心 CPU (2.4GHz), 使用了一张 NVIDIA GeForce GTX TITAN X GPU。

表 4.5 EATNN 与不同推荐模型的训练耗时情况, 其中 s/m/h 分别代表秒/分钟/小时

模型	Ciao			Epinion			Flixster		
	每轮时长	训练轮数	总训练时长	每轮时长	训练轮数	总训练时长	每轮时长	训练轮数	总训练时长
TranSIV	55s	50	46m	410s	50	342m	37m	50	31h
SAMN	31s	500	258m	92s	500	767m	56m	200	8d
EATNN-E	13s	200	43m	97s	200	324m	32m	200	5d
EATNN	1.8s	200	6m	11s	200	37m	8m	200	27h

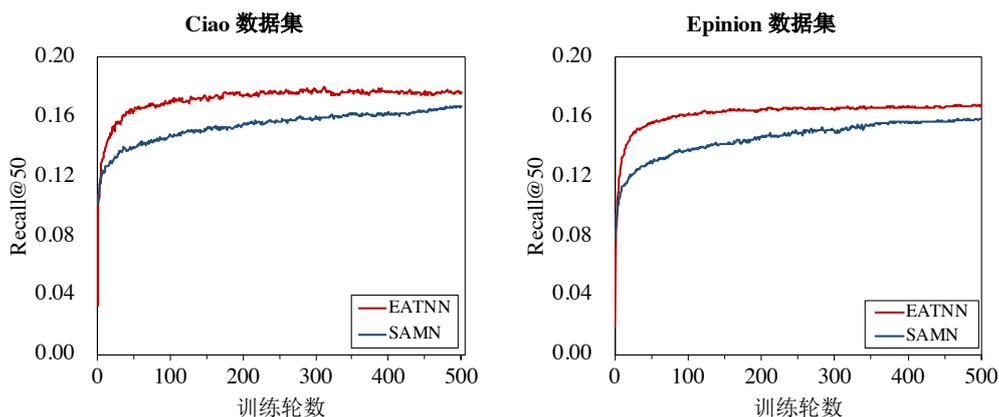


图 4.3 EATNN 和 SAMN 随训练轮数的收敛情况

我们首先比较了这三种方法的总体运行时间。EATNN-E 表示使用原始非采样损失函数 (公式 (3.2)) 进行计算的方法, 其结果也被添加进来, 以更好地显示我们方法的训练效率。实验结果如表4.5所示。首先, 可以观察到, EATNN 的训练时间远小于 EATNN-E, 这表明与原始损失函数相比, 使用我们上章所提出的高效非采样算法可以更加快速的从整体数据中学习模型参数。其次, EATNN 的训练速度比 TranSIV、SAMN 和 EATNN-E 快得多。特别是对于最大的数据集 Flixster, EATNN 只需要 27 小时就可以达到最佳性能, 而 SAMN 和 EATNN-E 分别需要大约 8 天和 5 天才能训练完毕。对于其他数据集, EATNN 的训练所需时间同样远低于其他方法。在真实的应用场景中, 模型训练时间也是需要考虑的一个重要因素。我们的 EATNN 模型在训练效率方面显示出显著的优势, 表明了算法具有很强的实用性。

我们还研究了神经网络模型 SAMN 和我们的 EATNN 的学习收敛过程。图4.3显示了两个模型随训练轮数的收敛情况。为了方便, 我们仅在图上展示了以 Recall@50 作为评价指标在 Ciao 和 Epinion 数据集上的结果。对于 Flixster 数据集和其他评价指标的结果与之类似。从图中可以看出, EATNN 的收敛速度比 SAMN 快得多, 并且始终保持更好的表现。原因在于 SAMN 是基于负采样进行训练的, 通常需要更多的训练轮数, 并且很难将模型优化到最优的状态。

表 4.6 Epinion 数据集冷启动条件下的实验结果, 25% 和 50% 代表训练集所占数据集比例

25%	Recall@10	Recall@50	Recall@100	NDCG@10	NDCG@50	NDCG@100	RI
<b>BPR</b>	0.0211	0.656	0.1009	0.0394	0.0501	0.0625	+61.84%
<b>ExpoMF</b>	0.0466	0.0752	0.1068	0.0497	0.0584	0.0699	+22.13%
<b>NCF</b>	0.0440	0.0781	0.1162	0.0449	0.0570	0.0709	+23.22%
<b>SBPR</b>	0.0387	0.0784	0.1157	0.0423	0.0543	0.0691	+29.07%
<b>TranSIV</b>	<u>0.0519</u>	<u>0.0859</u>	<u>0.1211</u>	<u>0.0549</u>	<u>0.0657</u>	<u>0.0785</u>	+8.68%
<b>SAMN</b>	0.0494	0.0832	0.1197	0.0517	0.0604	0.0731	+14.43%
<b>EATNN</b>	<b>0.0567**</b>	<b>0.0934**</b>	<b>0.1328**</b>	<b>0.0593**</b>	<b>0.0708**</b>	<b>0.0853**</b>	-

50%	Recall@10	Recall@50	Recall@100	NDCG@10	NDCG@50	NDCG@100	RI
<b>BPR</b>	0.0408	0.1143	0.1680	0.0525	0.0743	0.0912	+31.03%
<b>ExpoMF</b>	0.0611	0.1223	0.1701	0.0606	0.0827	0.0982	+13.30%
<b>NCF</b>	0.0576	0.1216	0.1745	0.0559	0.0784	0.0955	+17.30%
<b>SBPR</b>	0.0433	0.1257	0.1742	0.0554	0.0798	0.0974	+23.00%
<b>TranSIV</b>	<u>0.0648</u>	0.1284	0.1797	<u>0.0633</u>	0.0856	0.1010	+8.34%
<b>SAMN</b>	0.0642	<u>0.1301</u>	<u>0.1805</u>	0.0630	<u>0.0867</u>	<u>0.1024</u>	+7.81%
<b>EATNN</b>	<b>0.0712**</b>	<b>0.1352**</b>	<b>0.1909**</b>	<b>0.0709**</b>	<b>0.0921**</b>	<b>0.1101**</b>	-

#### 4.3.3.7 冷启动表现分析

在本小节中我们进行实验探索 EATNN 模型在冷启动（用户—物品交互数据稀疏）条件下的表现。实验中用户—物品交互数据设置包括：1) 25% 用于训练, 75% 用于测试以及 2) 50% 用于训练, 50% 用于测试。同时, 结合社交行为的模型 (SBPR、TranSIV、SAMN 和 EATNN) 依然使用全部的社交数据。

我们在表4.6中展示了在 Epinion 数据集上的实验结果, 从表中可以看到: 首先, 与只使用用户—物品交互的方法相比, 结合社交信息的方法表现出更好的性能, 并且当用于训练的数据越少时, 相对提升越多。考虑到用户在社交领域和物品领域的偏好是相关的, 从社交行为中学习到的知识可以弥补用户—物品交互的不足, 帮助提升推荐效果。其次, EATNN 的表现显著优于其他基线方法。例如, 在 25% 的训练数据时比最佳基线方法提高了 8.68%, 在 50% 的训练数据时比最佳基线方法提高了 7.81%。这显示了 EATNN 通过自适应迁移学习利用用户的社交信息来解决推荐系统冷启动问题的有效性。

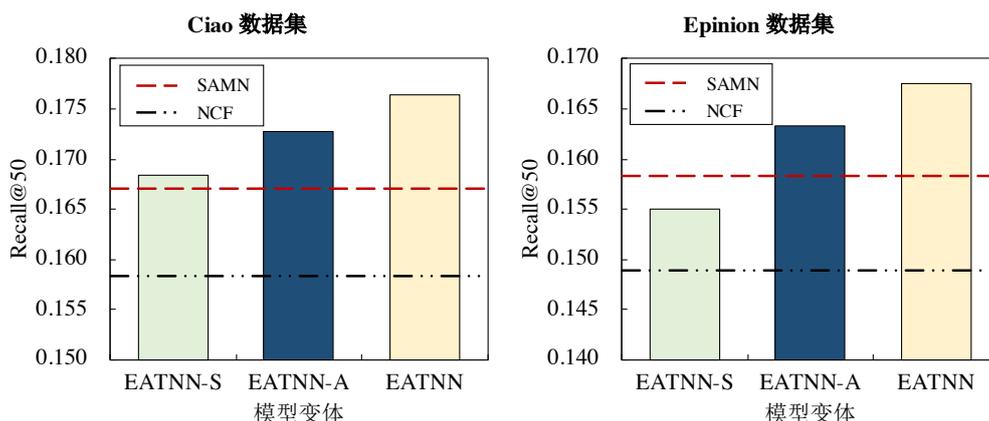


图 4.4 EATNN 和其模型变体的表现情况

#### 4.3.3.8 消融实验分析

为了进一步了解社交信息和设计的基于注意力机制的自适应迁移学习的有效性，我们对以下 EATNN 的模型变体进行了实验：

- **EATNN-S**: EATNN 去除社交信息的模型变体（即只用损失函数 (4.5) 训练模型）。
- **EATNN-A**: EATNN 中不使用注意力机制的模型变体，其中迁移学习的比例设为固定值 (0.5)。

图4.4中展示了两个变体在实验数据集上的推荐结果，为了直观地进行比较，我们在图中也标出了 NCF（表现最好的仅使用用户-物品交互的推荐方法）和 SAMN（表现最好的结合社交信息的推荐方法）的效果。有关实验结果的主要分析和结论如下：（1）EATNN-S 在我们的方法中表现最差，因为其没有利用社交信息来帮助对用户偏好建模。尽管如此，EATNN-S 仍然显著优于 NCF ( $p < 0.01$ )，表明非采样学习能够显著提升建模效果；2）EATNN 和变种 EATNN-A 的性能都优于 SAMN ( $p < 0.01$ )。当采用基于注意力的自适应迁移学习时，EATNN 与静态方法 EATNN-A 相比，表现得到了进一步的显著提升 ( $p < 0.05$ )。这表明物品和社交领域之间的共享知识是多样且多变的，应该针对不同的用户进行自适应迁移。实验结果也表明，本文所提出的基于注意力的自适应迁移能够有效地学习到共享知识的权重。

#### 4.3.3.9 自适应迁移学习案例分析

注意力网络的权重在一定程度上反映了模型的学习和推荐过程。在本小节我们提供了一些案例来展示在模型学习中的自适应迁移学习过程。表4.7展示了来自 Epinion 数据集的一些不同样本。从表中可以看到，EATNN 可以学习到这两个域之间的差异，并自适应地分配共享知识的权重。例如，表中前两位是活跃的用户，

表 4.7 自适应迁移学习权重展示

	训练样本数		物品域注意力权重		社交域注意力权重	
	用户—物品交互	用户—用户交互	$\alpha_{(C,u)}$	$\alpha_{(I,u)}$	$\beta_{(C,u)}$	$\beta_{(S,u)}$
案例 1	72	85	0.663	0.337	0.774	0.226
案例 2	11	54	0.741	0.259	0.647	0.353
案例 3	0	29	0.914	0.086	0.479	0.521
案例 4	34	0	0.422	0.578	0.882	0.118
案例 5	1	15	0.823	0.177	0.511	0.489

他们在物品和社交领域都有丰富的交互信息，注意力权重反映了共享知识是如何迁移的。以案例 1 为例，66.3% 的  $\mathbf{u}^C$  用于预测用户对物品的偏好，而 77.4% 的  $\mathbf{u}^C$  用于预测用户对社交网络中其他用户的偏好。注意力权重也能在一定程度上反映出交互信息的丰富程度。以案例 3 为例，用户在训练集中没有物品交互，其从物品交互中学到的向量的注意力权重约为 0。同样的结果也可以从案例 4 中观察到，其在训练集中没有社交信息。案例 5 的结果显示，该用户的物品交互信息不够丰富，因此需要从社交领域迁移更多知识来帮助对其偏好进行学习。

#### 4.4 结合多类型交互行为的高效推荐建模

在本节对于多类型交互行为的高效推荐建模进行阐述，包括问题定义、两个解决方案、以及相关实验。

##### 4.4.1 问题定义

表4.8展示了本节中两个模型所通用的符号和关键概念。具体更加详细的符号含义会在模型介绍中给出注释。假设数据集中有  $M$  个用户和  $N$  个物品，用户集合记为  $\mathbf{U}$ ，物品集合记为  $\mathbf{V}$ 。我们使用索引  $u$  表示用户，使用  $v$  表示物品。用户的多类型交互行为被记为  $\{\mathbf{Y}_{(1)}, \mathbf{Y}_{(2)}, \dots, \mathbf{Y}_{(K)}\}$ ，其中  $\mathbf{Y}_{(k)} = [y_{(k)uv}]_{M \times N} \in \{0, 1\}$  表示用户  $u$  和物品  $v$  是否有第  $k$  种类型的交互。通常来说，多类型交互行为中有最主要的目标行为，如电子商务中的购买行为。目标行为被记为  $\mathbf{Y}_{(K)}$ 。在此设置下，结合多类型交互行为的推荐任务定义如下：

**输入：** 用户集  $\mathbf{U}$ ，物品集  $\mathbf{V}$ ，用户—物品多类型交互记录  $\{\mathbf{Y}_{(1)}, \mathbf{Y}_{(2)}, \dots, \mathbf{Y}_{(K)}\}$ ，其中目标行为被记为  $\mathbf{Y}_{(K)}$ 。

**输出：** 给定一个用户，为其给出按照与物品目标行为交互概率排序得到的物品推

表 4.8 结合多类型交互行为的高效推荐建模中所用数学符号

符号	描述
$U, V$	分别代表用户集合和物品集合
$B$	一批训练用户
$\{Y_{(1)}, Y_{(2)}, \dots, Y_{(K)}\}$	用户—物品的多种类型交互
$K$	交互类型数量
$\mathcal{Y}_k$	第 $k$ 种类型的正向用户—物品交互
$c_{uv}^k$	训练样例 $y_{(k)uv}$ 的权重
$d$	向量维度
$\Theta$	神经网络参数集合

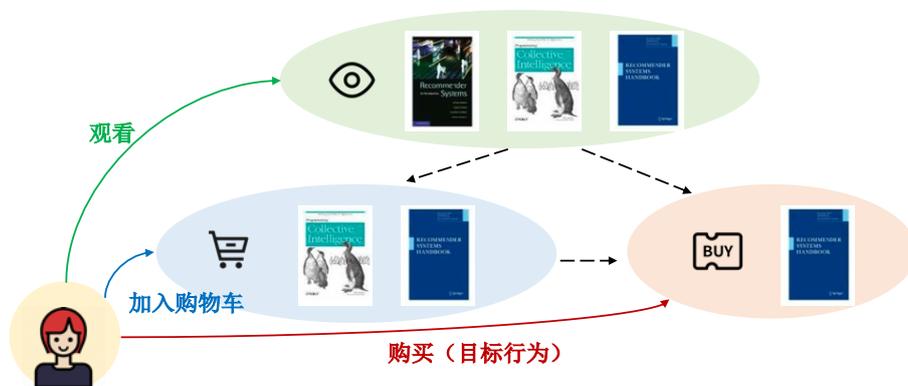


图 4.5 用户多类型交互行为之间因果关系示意图

荐列表。

#### 4.4.2 基于因果推理的高效多行为推荐模型

##### 4.4.2.1 模型概述

图4.5展示了本节设计基于因果推理的多行为推荐方法的动机。通常来说，用户的多类型交互行为之间不是互相独立的，存在有一定的因果关系。举例来说，在电子商务平台上，用户对物品的购买行为依赖于观看行为和加入购物车行为，且加入购物车行为也依赖于观看行为。因此，在建模时考虑行为之间的因果关系可以更加精细地从多类型交互数据中学习用户偏好。

本节我们将介绍所提出的基于因果推理的高效多行为推荐模型 EHCF，模型的整体示意图如图4.6所示。用户和物品通过嵌入层转换为稠密向量表示，值得一提的是，与以往工作中通常输入用户—物品对  $(u, v)$  进行训练不同，我们使用上一章所提出的高效非采样算法（基于用户的）对整体模型进行优化，其中以用户和其

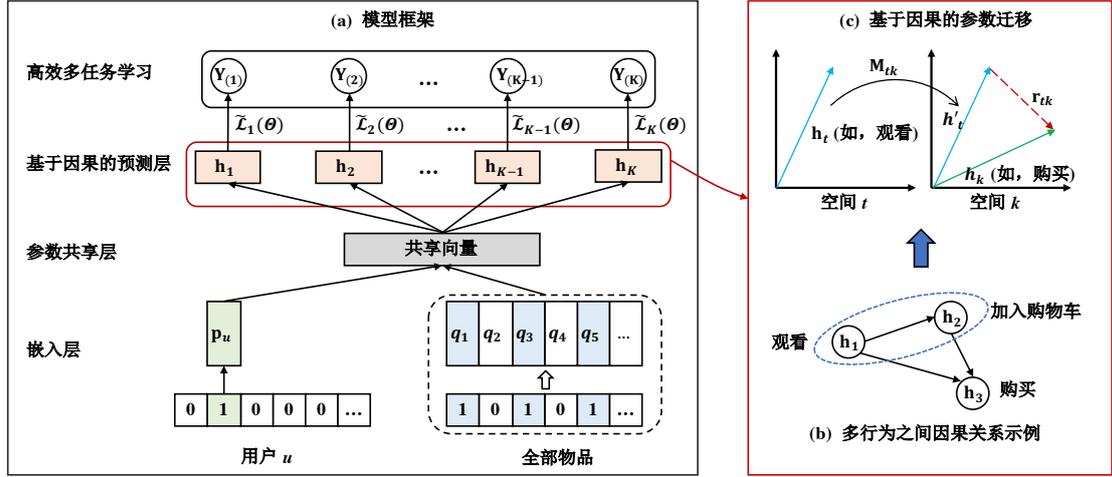


图 4.6 基于因果推理的高效多行为推荐模型 (EHCF) 示意图

所有交互过的物品作为输入。EHCF 模型采用多任务学习的方式，将每种交互行为作为单独的预测任务，并且对于所有的任务共享用户和物品的表示向量。为了在同样的输入下预测不同类型的交互行为，需要为每个任务学习一个单独的预测层。用  $\mathbf{h}_k$  表示第  $k$  类行为的预测层，则用户  $u$  对物品  $v$  第  $k$  种类型交互的概率分数计算如下：

$$\hat{y}_{(k)uv} = \mathbf{h}_k^T (\mathbf{p}_u \odot \mathbf{q}_v) = \sum_{i=1}^d h_{k,i} p_{u,i} q_{v,i} \quad (4.9)$$

其中  $\mathbf{p}_u \in \mathbb{R}^d$  和  $\mathbf{q}_v \in \mathbb{R}^d$  表示用户  $u$  和物品  $v$  的表示向量； $\odot$  符号表示哈达玛积，即对于两个向量进行逐项相乘。

#### 4.4.2.2 基于因果的预测层学习

基于因果的预测层学习是 EHCF 模型的关键设计。正如之前所介绍的，在现实生活中，用户的多类型交互行为之间通常存在一定的因果关系。因此，不同行为的预测层应该是相互关联的。为了建模行为间因果关系，在预测一种行为时需要考虑其因果图上前一个行为的预测影响。具体地，如果在因果图上行为  $t$  到行为  $k$  之间存在有递进关系（如用户的观看行为到购买行为），则这种关系在本节中通过如下方式对其预测层  $\mathbf{h}_t$  和  $\mathbf{h}_k$  进行建模（如图4.6所示）：

$$f_{\mathbf{h}_t \rightarrow \mathbf{h}_k} = \mathbf{h}_t \mathbf{M}_{tk} + \mathbf{r}_{tk} \quad (4.10)$$

其中  $\mathbf{M}_{tk} \in \mathbb{R}^{d \times d}$  表示从行为  $t$  到行为  $k$  的迁移矩阵，目的是将向量  $\mathbf{h}_t$  空间投影至  $\mathbf{h}_k$ ； $\mathbf{r}_{tk} \in \mathbb{R}^d$  表示两种行为迁移后的偏差。基于公式 (4.10)，对第  $k$  种行为的预

测向量表示为：

$$\mathbf{h}_k = \sum_t (f_{\mathbf{h}_t \rightarrow \mathbf{h}_k}) = \sum_t (\mathbf{h}_t \mathbf{M}_{tk} + \mathbf{r}_{tk}) \quad (4.11)$$

即第  $k$  种行为的预测向量取决于其所有因果图上的前序行为的预测向量，并通过向量迁移提升泛化性。

#### 4.4.2.3 高效多任务学习

如前所述，在上一章的工作中我们已经提出了针对双塔推荐模型的高效非采样学习算法。将定理3.1应用到结合多类型交互数据的推荐场景上，则基于计算得到的用户、物品以及各类型交互预测层向量，对于第  $k$  类交互预测的损失函数为：

$$\begin{aligned} \tilde{\mathcal{L}}_k(\Theta) = & \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}^+} \left( (c_v^{k+} - c_v^{k-}) \hat{y}_{(k)uv}^2 - 2\hat{y}_{(k)uv} \right) \\ & + \sum_{i=1}^d \sum_{j=1}^d \left( (h_{k,i} h_{k,j}) \left( \sum_{u \in \mathbf{B}} p_{u,i} p_{u,j} \right) \left( \sum_{v \in \mathbf{V}} c_v^{k-} q_{v,i} q_{v,j} \right) \right) \end{aligned} \quad (4.12)$$

其中  $c_v^k$  表示训练样本  $y_{(k)uv}$  的权重。

多任务学习是一种对几个不同但相关的任务进行联合训练的方式，以便在每个任务之间互相起到辅助、增强的作用，从而获得更好的训练效果<sup>[124]</sup>。因此，我们对用户多个类型行为的预测任务同时进行训练，构成了端到端的联合学习，具体公式如下：

$$\mathcal{L}(\Theta) = \sum_{k=1}^K \mu_k \tilde{\mathcal{L}}_k(\Theta) + \lambda \|\Theta\|_2^2 \quad (4.13)$$

其中  $K$  是交互行为的种类数量，超参数  $\mu_k$  用来控制不同任务的训练比重，我们设置  $\sum_{k=1}^K \mu_k = 1$  以便于调整这些超参数； $L_2$  正则化项则被用于限制模型参数  $\Theta$ ，避免出现过拟合现象。

如前所述，我们上章所介绍的高效非采样学习算法可以直接在现代深度学习工具包如 TensorFlow 上进行参数更新，因此这里不再对梯度计算方法做过赘述。此外，为了更好地训练模型，我们采用 Adagrad<sup>[105]</sup> 作为模型的优化器，它的主要优势在于会自适应地调整训练过程中的学习率，选择合理的步长来对解空间进行搜索，从而避免了对于学习率这个超参数过多的调参过程。去除函数 (Dropout) 也被加入到模型中用来避免训练出现过拟合现象<sup>[123]</sup>，我们的随机去除主要是针对作为不同任务间共享向量的用户表示  $\mathbf{p}_u$  进行的，去除操作之后，保留比例为  $\rho$  的向量参数。

#### 4.4.2.4 讨论

对 EHCF 模型的计算复杂度进行分析，由于 EHCF 在训练过程中同时对  $K$  个交互类型的预测任务进行优化，因此复杂度也将分为  $K$  部分。对于第  $k$  个任务（公式 (4.12)），每一批训练数据的计算复杂度为  $O((|\mathbf{B}| + |\mathbf{V}|)d^2 + |\mathcal{Y}_{\mathbf{B}}^k|d)$ （预测层参数更新的计算复杂度相对要小很多，可以忽略），其中  $\mathcal{Y}_{\mathbf{B}}^k$  表示每批训练数据中实际存在的第  $k$  类用户—物品正向交互。因此，通过多任务学习整体的计算复杂度为  $O(K(|\mathbf{B}| + |\mathbf{V}|)d^2 + \sum_{k=1}^K |\mathcal{Y}_{\mathbf{B}}^k|d)$ 。如果不使用高效非采样则传统非采样方法的复杂度为  $O(K|\mathbf{B}||\mathbf{V}|d)$ ，考虑到现实场景中  $|\mathcal{Y}_{\mathbf{B}}^k| \ll |\mathbf{B}||\mathbf{V}|$ ，且  $d \ll |\mathbf{B}|$ ，优化后的 EHCF 较传统的非采样训练方法效率提升明显。

#### 4.4.3 基于异质图神经网络的高效多行为推荐模型

##### 4.4.3.1 预备知识

近年来图神经网络因其对于结构化数据的强大学习能力受到了产业界和学术界的广泛关注<sup>[113,125-126]</sup>。然而，当前关于图神经网络的研究主要集中于学习单类型关系图中的节点表示。给定一个图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ，其中  $\mathcal{V}$  表示图上的节点集合， $\mathcal{E}$  表示图上的边集合，则通过一个单层的图卷积层学习节点表示定义如下：

$$\mathbf{E} = \sigma(\hat{\mathbf{A}}\mathbf{E}^{(0)}\mathbf{W}) \quad (4.14)$$

其中  $\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\mathbf{D}^{-\frac{1}{2}}$  表示图上节点附加自连接的范化邻接矩阵； $\mathbf{D}$  是节点度 (Degree) 的对角矩阵，定义为： $\mathbf{D}_{ii} = \sum_j (\mathbf{A} + \mathbf{I})_{ij}$ ； $\mathbf{I}$  表示单位矩阵 (Identity Matrix)， $\mathbf{E}^{(0)}$  是节点集通过嵌入层初始化的向量表示； $\mathbf{W}$  是图卷积层的参数； $\sigma$  是非线性激活函数。以上图卷积层学习节点的表示是通过对其直接相关的邻接节点计算得到。为了建模图上的高阶邻接关系（两个节点非直接相连，但可以沿着边通过多次传播相连），可以叠加多个图卷积操作如下：

$$\mathbf{E}^{(l)} = \sigma(\hat{\mathbf{A}}\mathbf{E}^{(l-1)}\mathbf{W}^{(l)}) \quad (4.15)$$

其中  $l$  代表第  $l$  个表示传播层， $\mathbf{W}^{(l)}$  表示  $l$  层的用于表示聚合的参数矩阵。

对于包含有多个类型边的关系图（异质图） $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ ，其中  $\mathcal{R}$  表示图上边的集合，一种常用的建模方式是<sup>[119-120]</sup>：

$$\mathbf{E}^{(l)} = \sigma(\hat{\mathbf{A}}\mathbf{E}^{(l-1)}\mathbf{W}_r^{(l)}) \quad (4.16)$$

其中  $\mathbf{W}_r^{(l)}$  表示  $l$  层中与边种类相关的聚合参数矩阵。虽然这种方式可以建模包含多个类型边的关系图数据，但是仍存在两点不足：(1) 会显著增加模型参数量，尤其是在图上边种类较多的情况下；(2) 只能学习到图上的节点表示，而未能学习

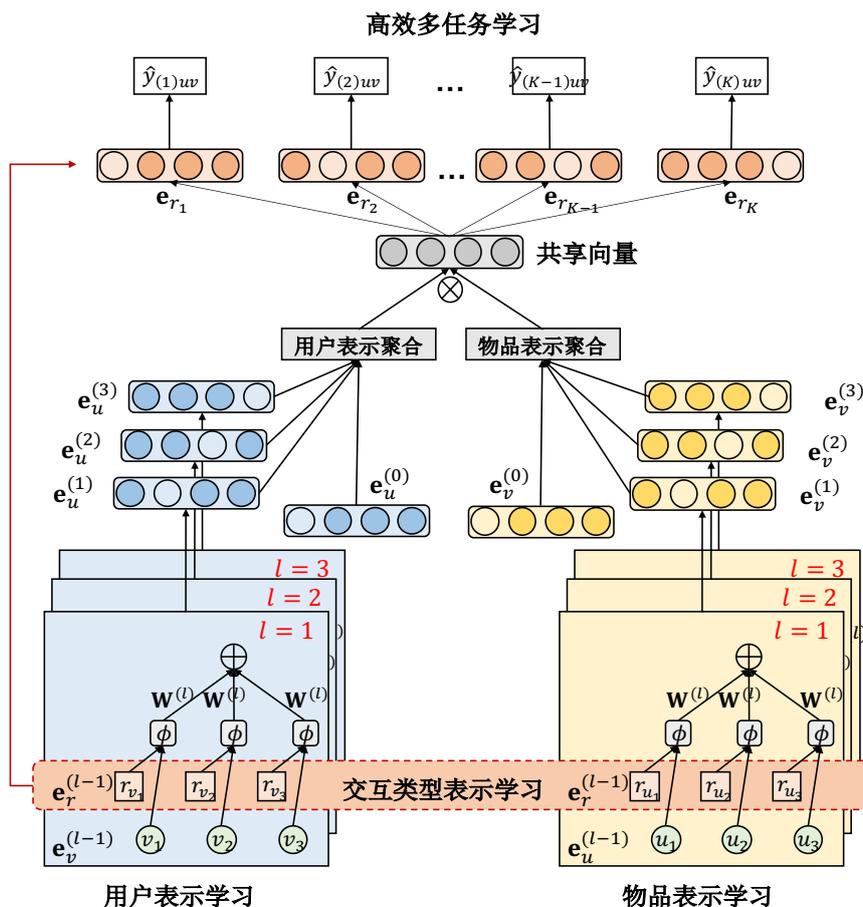


图 4.7 基于异质图神经网络的高效多行为推荐模型（GHCF）示意图

边的表示。因此，需要对图卷积层进行更进一步的设计以适应于多行为推荐任务。

#### 4.4.3.2 模型概览

以往工作中图神经网络在学习图数据时展示出了强大建模能力<sup>[113,125-126]</sup>，考虑到用户对物品的多类型交互天然的与异质图数据相符，为了建模用户和物品之间高阶的、多类型的交互行为，本节提出基于异质图神经网络的高效多行为推荐模型（GHCF），如图4.7所示。

GHCF 模型包含有三个重要组成部分，分别是：

1. 基于交互类型的图卷积层，不仅可以同时学习用户、物品、以及交互类型的向量表示，也避免了参数量的显著增加。
2. 模型预测部分，通过学到的向量表示预测用户对于物品每种类型交互行为的概率。
3. 高效多任务学习部分，扩展所提出的高效非采样学习对所有预测任务进行联合优化。

#### 4.4.3.3 基于交互类型的图卷积层

在 GHCF 中, 通过采用图卷积神经网络来建模用户和物品的协同过滤信息。图卷积神经网络的基本思想是通过平滑节点的邻接节点特征来学习当前节点的表示。在 GHCF 中, 用户 (或物品) 的表示是通过聚合其来自所有交互过的物品 (或用户) 来建模的。首先, 以用户  $u$  为例, 对于公式 (4.16), 可以重构为:

$$\mathbf{e}_u^{(l)} = \sigma \left( \sum_{(v,r) \in \mathcal{N}(u)} \frac{1}{\sqrt{|\mathcal{N}_u| |\mathcal{N}_v|}} \mathbf{W}_r^{(l)} \mathbf{e}_v^{(l-1)} \right) \quad (4.17)$$

其中  $\mathcal{N}(u)$  和  $\mathcal{N}(v)$  分别表示用户  $u$  和物品  $v$  直接相连的节点集合; 其中  $\mathbf{W}_r^{(l)}$  表示  $l$  层中与边种类相关的聚合参数矩阵; 正则化项  $\frac{1}{\sqrt{|\mathcal{N}_u| |\mathcal{N}_v|}}$  用于避免表示向量的规模随着图卷积操作的叠加而激增。然而如前所述, 该方式会显著增加模型参数量, 并且只能学习到图上的用户和物品的表示, 而未能学习交互类型的表示。

为了解决上述问题, 在我们的模型中, 我们对相邻节点  $v$  与其交互类型  $r$  进行组合操作 ( $\phi$ ), 以对包含有交互类型的用户-物品关系建模。受知识图谱表示学习方法中使用的实体-关系组合操作的启发<sup>[127-128]</sup>, 基于交互类型的图卷积层定义为:

$$\mathbf{e}_u^{(l)} = \sigma \left( \sum_{(v,r) \in \mathcal{N}(u)} \frac{1}{\sqrt{|\mathcal{N}_u| |\mathcal{N}_v|}} \mathbf{W}^{(l)} \phi(\mathbf{e}_v^{(l-1)}, \mathbf{e}_r^{(l-1)}) \right) \quad (4.18)$$

其中  $\mathbf{W}^{(l)}$  表示  $l$  层的用于表示聚合的参数矩阵;  $\phi$  是一个组合运算符, 用于将交互类型嵌入到图卷积操作过程中; 激活函数  $\sigma$  采用了 LeakyReLU<sup>[129]</sup>。公式 (4.18) 使得我们的模型可以在线性参数下同时学习用户、物品和交互类型的表示。在本文中, 组合运算  $\phi$  使用如下操作:

$$\phi(\mathbf{e}_v, \mathbf{e}_r) = \mathbf{e}_v \odot \mathbf{e}_r \quad (4.19)$$

其中  $\odot$  表示向量按元素逐项相乘。值得一提的是, 这里也可以替换为其他组合运算操作如元素相减<sup>[127]</sup>或是采用神经网络的方式<sup>[20]</sup>。

另外, 在图卷积操作的过程中我们只对邻居节点的表示进行了聚合, 而没有聚合自身的表示。之前工作<sup>[42]</sup>中也采用了同样的做法, 并且通过实验展示了将图卷积每一层的表示进行聚合 (在接下来的小节介绍) 已经在一定程度上建模了节点本身的表示。

在对节点的表示按照公式 (4.18) 更新之后, 对于交互关系的表示按照如下方式计算:

$$\mathbf{e}_r^{(l)} = \mathbf{W}_{rel}^{(l)} \mathbf{e}_r^{(l-1)} \quad (4.20)$$

其中  $\mathbf{W}_{rel}^{(l)}$  是  $l$  层的用于关系转换的参数矩阵。

在第一层的图卷积操作过程中,  $\mathbf{e}_u^{(0)}$ ,  $\mathbf{e}_v^{(0)}$ , 和  $\mathbf{e}_r^{(0)}$  分别表示用户、物品、交互类型的初始化表示向量。更高层的图卷积操作则是根据前一层邻接节点和交互类型的表示向量聚合得到。

#### 4.4.3.4 模型预测

在经过  $L$  层的图卷积操作后, 我们获得了用户  $u$ 、物品  $v$  和关系  $r$  的多个表示。从不同层获得的表示强调了从不同跳 (Hop) 传递来的信息。例如, 第一层对具有直接交互的用户和物品强制平滑操作, 第二层则是平滑在交互物品 (用户) 上有重叠的用户 (物品), 更高的层则建模更高阶的关系信息。因此, 我们进一步将不同层学到的表示聚合起来, 得到用户、物品、和交互关系的最终表示:

$$\mathbf{e}_u = \sum_{l=0}^L \frac{1}{L+1} \mathbf{e}_u^{(l)}; \mathbf{e}_v = \sum_{l=0}^L \frac{1}{L+1} \mathbf{e}_v^{(l)}; \mathbf{e}_r = \sum_{l=0}^L \frac{1}{L+1} \mathbf{e}_r^{(l)} \quad (4.21)$$

其中在聚合时对每一层设置了同样的权重  $1/(L+1)$ 。虽然该设置比较简单, 但已经可以取得不错的效果, 这里也可以替换为其他权重设置方式如注意力机制<sup>[121]</sup>方法。

在得到用户、物品、和交互关系的最终表示之后, 用  $\mathbf{e}_{r_k}$  指代第  $k$  种交互行为的表示向量, 则用户  $u$  对物品  $v$  第  $k$  种类型交互的概率分数计算如下:

$$\hat{y}_{(k)uv} = \mathbf{e}_u^T \cdot \text{diag}(\mathbf{e}_{r_k}) \cdot \mathbf{e}_v = \sum_i^d e_{u,i} e_{r_k,i} e_{v,i} \quad (4.22)$$

其中  $\text{diag}(\mathbf{e}_{r_k})$  表示对角元素为  $\mathbf{e}_{r_k}$  的对角矩阵;  $d$  表示向量的维度。

#### 4.4.3.5 高效多任务学习

GHCF 模型同样使用我们所提出的高效非采样学习算法进行多任务联合学习, 将定理3.1应用到结合多类型交互数据的推荐场景上, 则基于计算得到的用户、物品以及各类型交互向量, 对于第  $k$  类交互预测的损失函数为:

$$\begin{aligned} \tilde{\mathcal{L}}_k(\Theta) = & \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}_{(u)}^{k+}} \left( (c_v^{k+} - c_v^{k-}) \hat{y}_{(k)uv}^2 - 2c_v^{k+} \hat{y}_{(k)uv} \right) \\ & + \sum_{i=1}^d \sum_{j=1}^d \left( (e_{r_k,i} e_{r_k,j}) \left( \sum_{u \in \mathbf{B}} e_{u,i} e_{u,j} \right) \left( \sum_{v \in \mathbf{V}} c_v^{k-} e_{v,i} e_{v,j} \right) \right) \end{aligned} \quad (4.23)$$

其中  $c_v^k$  表示训练样本  $y_{(k)uv}$  的权重。

对用户多个类型行为的预测任务同时进行训练，具体公式如下：

$$\mathcal{L}(\Theta) = \sum_{k=1}^K \mu_k \tilde{\mathcal{L}}_k(\Theta) + \lambda \|\Theta\|_2^2 \quad (4.24)$$

其中  $K$  是交互行为的种类数量，超参数  $\mu_k$  用来控制不同任务的训练比重，我们设置  $\sum_{k=1}^K \mu_k = 1$  以便于调整这些超参数； $L_2$  正则化项则被用于限制模型参数  $\Theta$ ，避免出现过拟合现象。此外，为了更好地训练模型，我们采用 Adagrad<sup>[105]</sup> 作为模型的优化器，它的主要优势在于会自适应地调整训练过程中的学习率，选择合理的步长来对解空间进行搜索，从而避免了对于学习率这个超参数过多的调参过程。去除函数 (Dropout) 也被加入到模型中用来避免训练出现过拟合现象<sup>[123]</sup>，包括两类操作：(1) 随机去除关系图上的一定比例的邻接节点；(2) 随机去除用户表示向量上一定比例的元素。

#### 4.4.4 实验

在实验部分，我们统一测试结合多类型交互行为的推荐场景下本节所提出的 EHCF 模型和 GHCF 模型的表现效果。实验所用代码已开源，详见 EHCF<sup>①</sup>和 GHCF<sup>②</sup>。

##### 4.4.4.1 数据集介绍

我们在两个包含用户多类型交互行为的电子商务数据集上进行了实验，分别是**贝贝**数据集和**淘宝**数据集。对两个数据集的简单介绍如下：

- **贝贝**：此数据集来自中国母婴用品平台贝贝网<sup>③</sup>，收集了平台上用户从 2017 年 6 月 1 日至 2017 年 6 月 30 日之间的多种交互行为记录，包括观看、加入购物车、以及购买行为。
- **淘宝**：此数据集来自电子商务平台淘宝<sup>④</sup>，收集了平台上用户从 2017 年 11 月 25 日至 2017 年 12 月 3 日之间的多种交互行为记录，同样包括观看、加入购物车、以及购买行为。

对这两个数据集，我们遵循现有工作<sup>[13,83]</sup>中常见的做法进行了预处理，包括去除重复交互记录以及交互频率少于 5 次的用户和物品。经过预处理后两个数据集的基本统计信息见表 4.9。

① <https://github.com/chenchongthu/EHCF>

② <https://github.com/chenchongthu/GHCF>

③ <https://www.beibei.com>

④ <https://www.taobao.com>

表 4.9 贝贝和淘宝数据集的基本统计信息

数据集	用户数量	物品数量	观看数量	加入购物车数量	购买数量
贝贝	21,716	7,977	2,412,586	642,622	304,576
淘宝	48,749	39,493	1,548,126	193,747	259,747

#### 4.4.4.2 基准方法

为了验证所提出的 EHCF 模型和 GHCF 模型的有效性，我们与现有的推荐算法进行了比较，按照是否可以利用多类型交互行为信息将这些方法分为两类，包括单行为推荐模型和多行为推荐模型，其中单行为推荐模型包括：

- **BPR**<sup>[6]</sup>：贝叶斯个性化排序，通过优化正负样本的偏序损失来训练矩阵分解类的推荐方法。
- **ExpoMF**<sup>[8]</sup>：同样基于矩阵分解和非采样学习的推荐模型，其中训练样例权重通过对物品的曝光（Exposure）概率进行建模。
- **NCF**<sup>[20]</sup>：神经网络协同过滤模型，是一个有代表性的基于神经网络的推荐模型，通过将矩阵分解与多层感知机相结合对隐式反馈数据进行建模。
- **ENMF**<sup>[49]</sup>：本文在上一章所提出的高效神经网络矩阵模型，通过高效非采样学习从整体数据上优化模型参数，是当前单行为推荐任务上表现最好的方法之一。
- **LightGCN**<sup>[42]</sup>：轻量化图卷积推荐模型，通过简化图卷积神经网络的设计使其更适合推荐任务，是当前基于图神经网络的表现最好的推荐方法之一。

对比的多行为推荐模型包括：

- **CMF**<sup>[80]</sup>：联合矩阵分解模型，将多种类型交互行为数据矩阵同时进行分解，在共享用户向量的同时使用不同的物品向量。
- **MC-BPR**<sup>[84]</sup>：多通道贝叶斯个性化排序模型，通过考虑不同交互类型之间的联系改进负采样策略的多行为推荐方法。
- **NMTR**<sup>[13]</sup>：在神经网络模型 NCF 的基础上借助多任务学习优化不同交互类型的预测任务，通过负采样策略对模型参数进行学习。

表4.10列出了本文所提出的 EHCF、GHCF 与基准方法的特点比较。

#### 4.4.4.3 评价方式

在模型评价过程中，我们将使用不同方法为每个用户对所有的候选物品进行排序（除了训练集和验证集已知的有正向交互的物品）<sup>[107]</sup>。同时，我们采用了在推荐系统评价中被广泛使用的留一法来进行方法评价<sup>[13,20]</sup>。对于每个用户对于物

表 4.10 EHCf、GHCF 与基准方法的特点比较

模型特点	BPR	NCF	ExpoMF	ENMF	LightGCN	CMF	MC-BPR	NMTR	EHCf	GHCF
购买行为	√	√	√	√	√	√	√	√	√	√
多类型行为	\	\	\	\	\	√	√	√	√	√
神经网络模型	\	√	\	√	√	\	\	√	√	√
非采样学习	\	\	√	√	\	\	\	\	√	√
因果推断建模	\	\	\	\	\	\	\	\	√	\
高阶关系建模	\	\	\	\	√	\	\	\	\	√

品的交互记录，我们保留最后一个交互作为测试数据，倒数第二个作为验证数据，其他的所有交互用做训练。在评价指标上，我们采用命中率（HR）和归一化的累积增益（NDCG）。HR@K 主要评价所预测的正例是否在推荐列表中的前 K 个，而 NDCG@K 则还会考虑所预测的正例在推荐列表中的具体位置，越靠前则得分越高。两个指标的计算公式参见 (3.12)。

#### 4.4.4.4 超参数设置

所有基准算法的参数设定都是根据对应论文的初始化要求设置的，且也都在验证集上进行了调参以确保其达到最优的效果。我们对超参数的主要搜索空间包括：学习率的搜索范围为 [0.005, 0.01, 0.02, 0.05]，为了避免过拟合，网络留存率 (Dropout) 的搜索范围为 [0.0, 0.1, ..., 0.9]，特征向量长度  $d$  的搜索范围为 [8, 16, 32, 64]，训练样本权重  $c_0$  的搜索范围为 [0.005, 0.01, 0.05, 0.1, 0.2, 0.5, 1] 等。最终，根据预实验结果，每个批的大小被设置成 512，学习率被设置成 0.05，特征向量的长度被设置成 64。对于 NCF 和 NMTR 模型，神经网络层数设为 2。对于图卷积神经网络模型 LightGCN 和我们的 GHCF，层数设为 4。对于 EHCf 和 GHCF 方法，网络留存率  $\rho$  设置为 0.8。对于训练样本权重，正样本权重设为 1，负样本权重  $c_0$  对贝贝数据集设为 0.01，对淘宝数据集设为 0.1。训练任务权重  $\mu$  的影响将在接下来的实验中具体分析。

#### 4.4.4.5 整体表现分析

本文所提出的方法 (EHCf 和 GHCF) 与其他基准方法在两个数据集上的推荐表现见表 4.11，为了对比不同长度推荐列表的效果，我们汇报了推荐列表长度为 10、50 和 100 的结果。从实验结果中我们有以下发现：

首先，结合多类型交互行为的推荐方法通常比只使用一种交互行为 (购买行为) 的方法表现更好。例如，在表 4.11 中，基准方法 NMTR 和我们的 EHCf 和 GHCF 均显著优于全部单行为模型。这与之前的工作结论一致<sup>[13]</sup>，这表明用户的多类型

表 4.11 EDCF, GHCF 与不同基线推荐方法的效果比较, \*\* 表示较最好的基准方法有显著提升, 显著性是  $p < 0.01$ 

	贝贝	HR@10	HR@50	HR@100	NDCG@10	NDCG@50	NDCG@100
单行为模型	BPR	0.0437	0.1246	0.2192	0.0213	0.0407	0.0539
	ExpoMF	0.0452	0.1465	0.2246	0.0227	0.0426	0.0553
	NCF	0.0441	0.1562	0.2343	0.0225	0.0445	0.0584
	ENMF	0.0464	0.1637	0.2586	0.0247	0.0484	0.0639
	LightGCN	0.0451	0.1613	0.2495	0.0232	0.0466	0.0611
多行为模型	CMF	0.0482	0.1582	0.2843	0.0251	0.0462	0.0661
	MC-BPR	0.0504	0.1743	0.2755	0.0254	0.0503	0.0653
	NMTR	0.0524	0.2047	0.3189	0.0285	0.0609	0.0764
	EDCF	<u>0.1523</u>	<u>0.3316</u>	<u>0.4312</u>	<u>0.0817</u>	<u>0.1213</u>	<u>0.1374</u>
	GHCF	<b>0.1922**</b>	<b>0.3794**</b>	<b>0.4711**</b>	<b>0.1012**</b>	<b>0.1426**</b>	<b>0.1575**</b>
	淘宝	HR@10	HR@50	HR@100	NDCG@10	NDCG@50	NDCG@100
单行为模型	BPR	0.0376	0.0708	0.0871	0.0227	0.0269	0.0305
	ExpoMF	0.0386	0.0713	0.0911	0.0238	0.0270	0.0302
	NCF	0.0391	0.0728	0.0897	0.0233	0.0281	0.0321
	ENMF	0.0398	0.0743	0.0936	0.0244	0.0298	0.0339
	LightGCN	0.0415	0.0814	0.1025	0.0237	0.0325	0.0359
多行为模型	CMF	0.0483	0.0774	0.1185	0.0252	0.0293	0.0357
	MC-BPR	0.0547	0.0791	0.1264	0.0263	0.0297	0.0361
	NMTR	0.0585	0.0942	0.1368	0.0278	0.0334	0.0394
	EDCF	<u>0.0717</u>	<u>0.1618</u>	<u>0.2211</u>	<u>0.0403</u>	<u>0.0594</u>	<u>0.0690</u>
	GHCF	<b>0.0807**</b>	<b>0.1892**</b>	<b>0.2599**</b>	<b>0.0442**</b>	<b>0.0678**</b>	<b>0.0792**</b>

交互行为也在一定程度上反映了用户的兴趣, 因此可以帮助提升推荐效果。

其次, 使用非采样学习策略的方法通常比基于负采样的方法表现更好。例如, 在实验结果中, ExpoMF 的性能优于 BPR, 我们的 EDCF, GHCF 表现优于所有基线方法。这同样与之前的工作结论一致<sup>[43,67]</sup>, 即: 非采样学习方法能够取得更优秀的模型表现。

第三, 我们提出的 EDCF 和 GHCF 方法在两个数据集上均展示出了最好的效果, 显著优于所有的基线方法 ( $p$  值  $< 0.01$ )。即便是与当时最新提出的深度学习模型 NMTR 相比, 我们的方法也分别在贝贝数据集和淘宝数据集取得了超过 47% 和 57% 的相对提升。虽然基于负采样的方法在之前的工作中得到了广泛的使用, 但本文认为其并不适合从多行为数据中学习推荐模型。举例来说, 为了生成一个训练样例, 基于负采样的方法 (如 MC-BPR、NMTR) 需要为每个交互类型采样一个

表 4.12 EHCf 与不同推荐模型的训练耗时情况，其中 s/m 分别代表秒/分钟

模型	贝贝			淘宝		
	每轮时长	训练轮数	总训练时长	每轮时长	训练轮数	总训练时长
NCF	62s	100	104m	115s	100	192m
ENMF	<b>3.2s</b>	100	<b>6m</b>	<b>6s</b>	100	<b>10m</b>
NMTR	165s	200	550m	180s	200	600m
EHCf-Original	62s	200	207m	192s	200	640m
EHCf	<b>7s</b>	200	<b>24m</b>	<b>16s</b>	200	<b>54m</b>

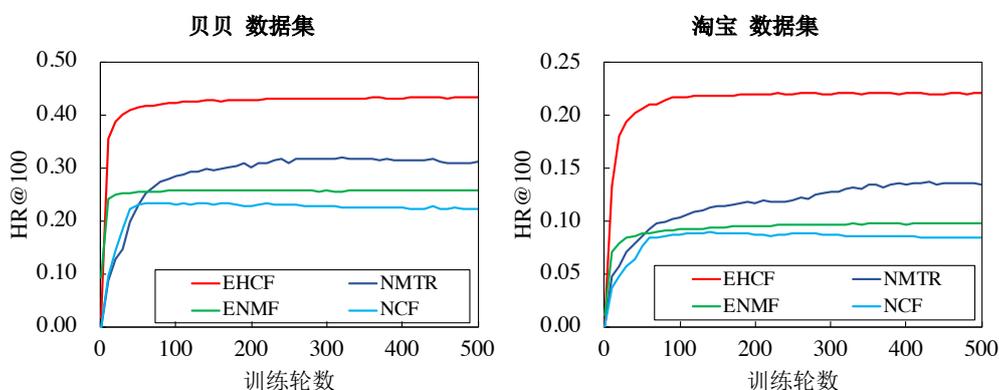


图 4.8 ENMF、NCF、NMTR、和 ECHF 模型随训练轮数的收敛情况

负例。这会产生非常大的随机性（是单行为场景的  $K$  倍），且某些行为的样本会被同时视为正样本和负样本，造成了优化矛盾，使得模型整体更难收敛到最优的状态。与之前的工作不同，我们的 EHCf 和 GHCF 模型中的参数是在整个数据上联合优化的。这就解释了为什么我们的 EHCf 在贝贝和淘宝上分别比单行为模型 ENMF 好 79.4% 和 108.8%，而 NMTR 方法在这两个数据集上只比 NCF 好 34.6% 和 46.9%。现有的基于负采样的多行为推荐方法无法充分利用辅助行为数据，这也是本工作的重点之一。

最后，比较本文所提出的 EHCf 模型和 GHCF 模型可以发现，GHCF 在两个数据集上相比于 EHCf 分别提升了 16.9% 和 14.2%。这也进一步验证了本文所提出的可以同时学习关系表示的图卷积层对于建模高阶的用户—物品多类型交互数据的有效性。

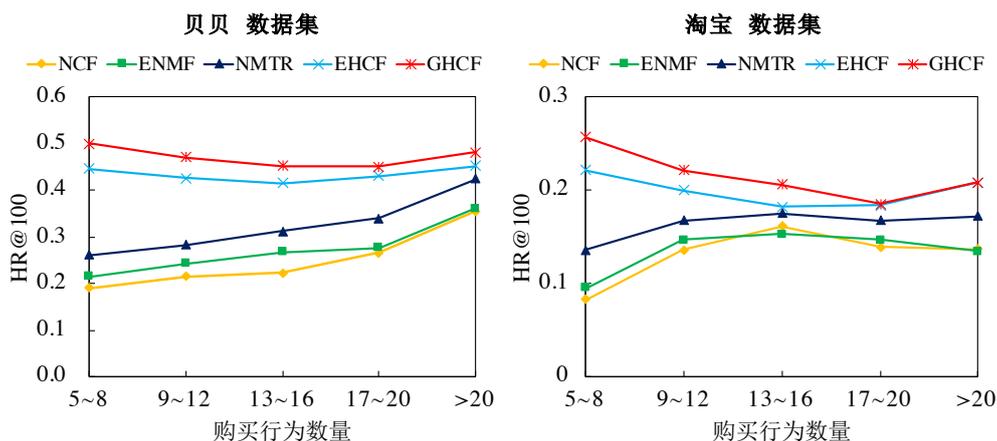


图 4.9 NCF、ENMF、NMTR、EHCF 和 GHCF 模型在不同购买行为数量下的表现情况

#### 4.4.4.6 训练效率分析

在本小节中，我们进行实验探索所设计的 EHCF 模型与基线模型中表现较好的 NCF 和 NMTR 的训练效率。所有的实验都是在同一台服务器上运行的，其配置为：英特尔 Xeon9 核心 CPU (2.4GHz)，使用了一张 NVIDIA GeForce GTX TITAN X GPU。

我们首先比较了这几种方法的总体运行时间。EHCF-Original 表示使用原始非采样损失函数（公式 (3.2)）进行训练的方法，其结果也被添加进来，以更好地显示 EHCF 方法的训练效率。实验结果如表 4.12 所示。首先，从表中可以看到，EHCF 所需的训练时间远少于 EHCF-Original，这表明与原始损失函数相比，使用我们上章所提出的高效非采样算法可以更加快速的从整体数据中学习模型参数。其次，与 NCF 和 NMTR 相比，我们的方法在单行为数据和多行为数据上的训练速度都要快得多。例如在淘宝数据集上，NMTR 需要大约 10 小时才能训练完毕，而 EHCF 只需 54 分钟即可达到最佳性能。在现实场景中，训练时间成本是需要考虑的一个重要因素。我们的 EHCF 模型在训练效率上显示出显著的优势，这使得它在现实生活中更具有实用性。

我们还研究了 ENMF、NCF、NMTR、和 ECHF 模型的学习过程。图 4.8 显示了这些方法随训练轮数的收敛情况，为了方便，我们仅在图上展示了以 HR@100 作为评价指标的结果。对于其他指标的结果，收敛情况与之类似。从图上可以看到，ENMF 和 EHCF 分别在单行为和多行为场景下收敛速度显著优于 NCF 和 NMTR，并且始终保持更好的表现效果。原因在于 NCF 和 NMTR 是基于负采样进行训练的，通常需要更多的训练轮数，并且在最终的优化效果上弱于非采样训练方法。

表 4.13 EHCf 及其变体在淘宝数据集上的表现效果

	淘宝	HR@10	HR@50	HR@100	NDCG@10	NDCG@50	NDCG@100
数据消融	购买	0.0398	0.0743	0.0936	0.0244	0.0298	0.0339
	购买和加购	0.0612	0.1187	0.1569	0.0297	0.0465	0.0527
	购买和观看	0.0673	0.1438	0.2061	0.0384	0.0493	0.0593
模型消融	去除因果建模	0.0633	0.1112	0.1525	0.0312	0.0389	0.0457
	去除联合学习	0.0611	0.1171	0.1552	0.0304	0.0448	0.0509
完整模型	EHCf	<b>0.0717**</b>	<b>0.1618**</b>	<b>0.2211**</b>	<b>0.0403**</b>	<b>0.0594**</b>	<b>0.0690**</b>

#### 4.4.4.7 稀疏数据表现分析

稀疏数据经常为推荐系统准确性带来挑战，因为系统很难对交互数量较少的非活跃用户的偏好建模。在现实场景中，相比于购买行为，用户对物品的观看和加入购物车的行为数量会更加丰富，因此结合多类型交互数据的推荐模型可以用以缓解稀疏数据带来的挑战。本小节通过实验验证所提出的 EHCf 和 GHCF 模型在稀疏数据场景下的表现效果，具体地，我们根据用户的购买行为数量将用户分为若干组，包括 [5—8, 9—12, 13—16, 17—20, >20]，在每个组上对 NCF、ENMF、NMTR、EHCf 和 GHCF 模型进行了实验，结果展示于图4.9中。为了方便，我们仅在图上展示了以 HR@100 作为评价指标的结果。对于其他指标的结果，表现趋势与之类似。

从图中结果可以看出，首先我们的 GHCF 和 EHCf 模型在不同的用户分组上始终优于其他方法，包括当时最新的 NMTR。尤其是对于第一个只有 5-8 个购买记录的用户组，我们的方法与其他方法相比提升更加明显。其次，有些方法随着用户的购买行为数量增加时效果有轻微下降，这是因为其他辅助行为数据的数量差异。例如，在淘宝数据集上，拥有 5-8 个购买记录的用户辅助行为记录数量远远多于拥有 17-20 个购买记录的用户。通常，辅助行为（如观看）的数据相比于目标行为（如购买）更容易收集，并且数据量更大。因此，利用辅助行为可以有效缓解数据稀疏问题。实验结果也验证了我们的 EHCf 和 GHCF 模型在稀疏数据场景下的有效性。

#### 4.4.4.8 消融实验分析

在本小节中，我们分别对 EHCf 和 GHCF 模型做消融实验，以展示多类型交互行为数据对于推荐模型效果提升的作用。首先，对 EHCf 的模型变种进行实验，结果展示于表4.13中。其中表里去除因果建模时，EHCf 里每个交互类型的预测层  $h_k$  通过随机初始化产生并且互相之间不再建立联系。去除联合学习时，对每个预

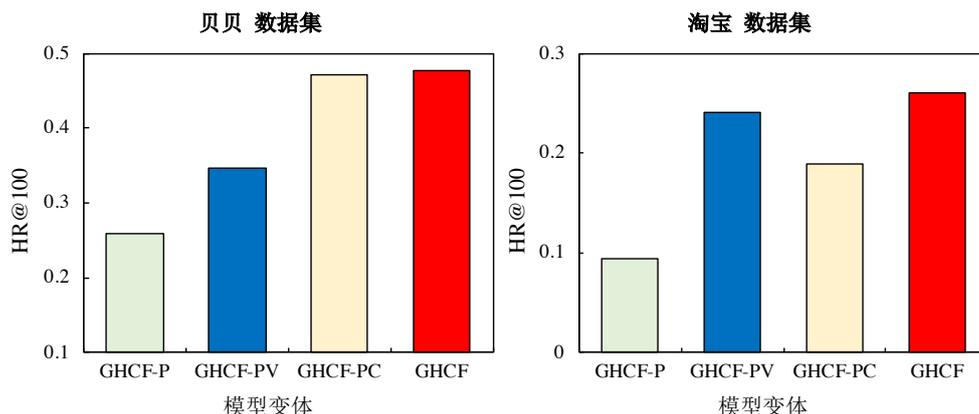


图 4.10 GHCF 及其变体的表现效果

测任务依次进行训练 (One-by-one Learning)。表4.13中展示了 EHCF 在淘宝数据集上的结果, 对于贝贝数据集, 观察到的结论与之类似。如表所示, 在 EHCF 中额外使用观看数据和加入购物车数据相比于只使用购买数据都会带来效果提升, 这显示了使用用户和物品的辅助交互行为对于提升推荐系统效果的有效性。此外, 去除因果建模和去除联合学习的模型性能都比完整的 EHCF 模型差, 这验证了所提出的基于因果推理的预测和多任务联合学习方法对于提升多行为推荐效果的显著作用。

接下来对几种 GHCF 变体进行了实验以验证在 GHCF 模型中辅助行为数据的有效性, 包括:

- GHCF-P: 该 GHCF 模型变体只使用了用户对物品的购买数据。
- GHCF-PV: 该 GHCF 模型变体使用了用户对物品的购买数据和观看数据。
- GHCF-PC: 该 GHCF 模型变体使用了用户对物品的购买数据和加入购物车数据。

图4.10显示了 GHCF 不同变体 HR@100 的结果。对于其他评价指标, 观察结果类似。如图所示, 在 GHCF 模型中添加观看数据和加入购物车数据相比于只使用购买数据都可以提高推荐性能。当使用所有这三种行为数据时, 我们的 GHCF 的效果相比于只使用两类数据时又得到了进一步提升。这验证了辅助行为对用户偏好建模的有效性。此外我们也观察到, 在贝贝数据集上, 购物行为对于结果提升的贡献大于在淘宝数据集上的贡献。原因可能是由于两个数据集上所包含的辅助行为数据的丰富程度不同。

#### 4.4.4.9 超参数影响分析

在本小节中, 我们通过实验展示一些超参数设置对于模型结果的影响。

首先, GHCF 模型通过图卷积操作建模用户和物品之间的高阶关系, 因此, 图

表 4.14 图卷积层数对 GHCF 模型结果的影响

	贝贝		淘宝	
	HR@100	NDCG@100	HR@100	NDCG@100
GHCF-1	0.4569	0.1494	0.2473	0.0755
GHCF-2	0.4636	0.1498	0.2501	0.0778
GHCF-3	0.4674	0.1551	0.2567	0.0787
GHCF-4	0.4711	0.1575	0.2599	0.0792
GHCF-5	0.4681	0.1554	0.2558	0.0782

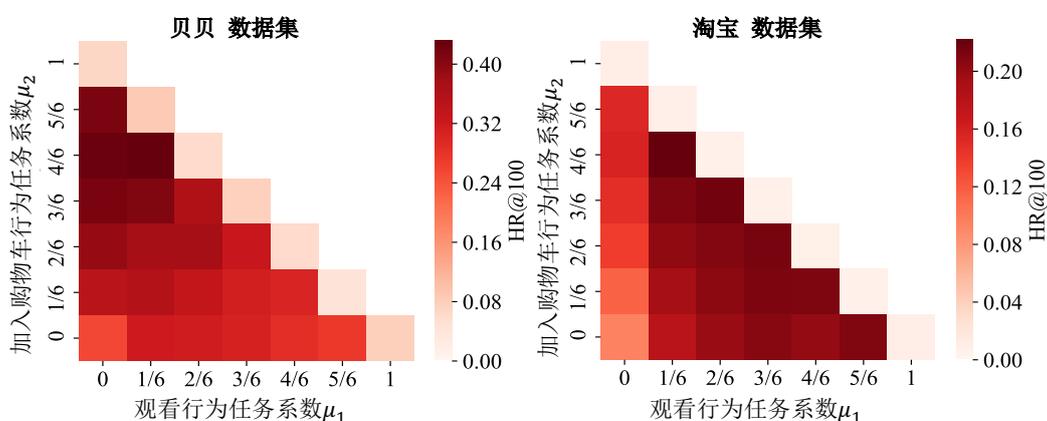


图 4.11 多任务学习系数对于 EHCf 模型的结果影响

卷积的层数是需要考虑的一个重要超参数。图卷积层数对于 GHCF 结果的影响展示于表4.14中，其中 GHCF- $k$  表示包含  $k$  个图卷积层的 GHCF 模型。从表中可以看出，通过将 GHCF 的图卷积层数量从 1 增加到 4，贝贝和淘宝数据集的推荐结果都得到了提升。结果显示，通常 4 个传播层足以建模用户和物品的高阶多类型交互关系。更多的层数可能会引入噪音并导致模型出现过拟合的现象。此外，当改变图卷积层的数量时，GHCF 在这两个数据集上仍显著优于其他基线方法。上述观察结果验证了通过多层图神经网络建模用户和物品的异质高阶关系对于提升推荐效果的有效性。

之后我们进一步测试了多任务学习中每个任务系数  $\mu$  对于 EHCf 和 GHCF 模型的影响。具体地， $\mu_k$  控制了模型优化时第  $k$  个任务的权重。在贝贝和淘宝数据集中三种行为类型对应的权重系数为：观看 ( $\mu_1$ )、加入购物车 ( $\mu_2$ )、和购买 ( $\mu_3$ )，且有  $\mu_1 + \mu_2 + \mu_3 = 1$ 。对于这些系数，我们在  $[0, 1/6, 2/6, 3/6, 4/6, 5/6, 1]$  的范围内调参，当给定  $\mu_1$  和  $\mu_2$  的值时， $\mu_3$  的值会直接确定。图4.11和图4.12分别展示了多任务学习系数对于 EHCf 和 GHCF 模型的结果影响。图中区块颜色越深代表

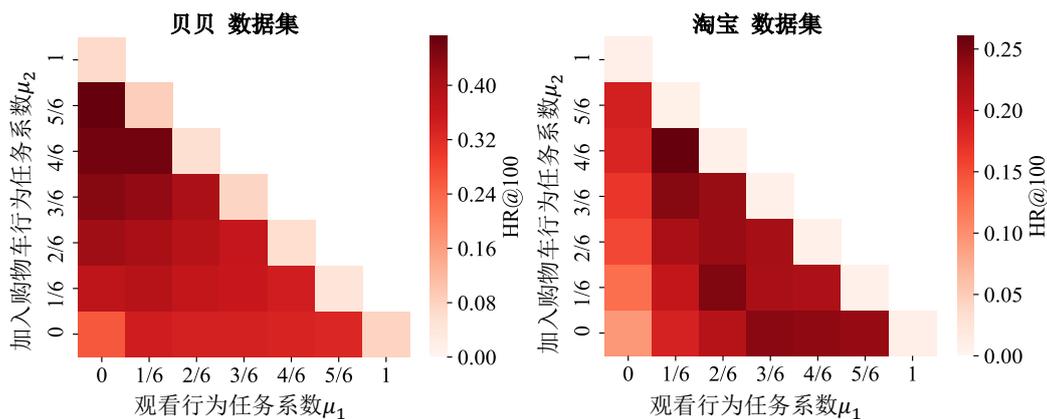


图 4.12 多任务学习系数对于 GHCF 模型的结果影响

效果越好。可以看到，最外层的区块颜色相对较浅，因为它们代表了  $\mu_3$  的值为 0，即没有优化目标任务（购买）。对于 EHCF 和 GHCF 模型，在贝贝数据集上，采用相对较大的  $\mu_2$  会提升模型效果，而在淘宝数据集上，相对较大的  $\mu_1$  会带来更好的性能。原因可能是两个数据集中辅助行为数据的丰富程度不同，因此在联合训练中所占的比重也应有所不同。

## 4.5 小结

本章面向高效神经网络推荐系统中作为核心的“推荐模型”问题，重点研究了结合异质交互行为的高效推荐建模。具体地，本章研究的异质交互行为包括用户之间的社交关系行为以及用户对物品的多类型交互行为。在结合社交行为数据方面，本章提出了基于自适应迁移学习的高效社交推荐模型 EATNN，通过引入注意力机制，自适应地在社交领域和物品领域之间迁移用户的共享偏好；在结合多类型交互行为数据方面，本章分别提出了基于因果推理的多任务学习模型 EHCF 和基于异质图神经网络的高效多行为推荐模型 GHCF。通过显式地建模不同行为之间的因果关系以及高阶联系，挖掘隐含的用户偏好，实现对于用户不同交互行为的精准建模。在多个数据集上的丰富实验和分析表明，我们所提出的 ETANN、EHCF、和 GHCF 模型较已有方法在推荐效果和训练效率上均取得了显著的提升，同时也可以大幅提升冷启动或数据稀疏情况下的推荐效果。

本章相关成果发表于 CCF A 类会议论文 SIGIR 2019<sup>[51]</sup>、AAAI 2020<sup>[52]</sup>、和 AAAI 2021<sup>[53]</sup>。

## 第5章 结合内容信息的高效推荐建模

### 5.1 引言

上一章对结合异质交互行为数据的高效神经网络推荐模型进行了研究。在现实的推荐系统应用场景中，除了上章所介绍的异质交互数据以外，平台通常也可以收集到许多额外的内容信息，如用户、物品的属性特征以及结构化的知识图谱信息等<sup>[29,88]</sup>。为了提供更精准的推荐结果，近年来出现了许多聚焦于如何引入内容信息到推荐系统建模过程中的研究工作。这些方法通过利用收集到的内容信息来补充用户和物品之间的交互关系，不仅可以提升推荐系统的表现，也能够一定程度上为推荐结果提供可解释性<sup>[95,130]</sup>。

在真实场景下，这些内容信息的引入和逐渐复杂的模型结构往往使得推荐系统面临很大的训练效率问题。此外，由于隐式数据中用户与物品的交互数量较少，现有的推荐方法通常依赖负采样方法对模型进行训练优化。然而，如前文所述，负采样训练会存在训练不足、负样例选择有偏等问题，使得模型无法优化到最优的状态。这些因素给结合内容信息的高效推荐建模带来了挑战，现有方法在内容信息的精细化建模以及模型训练效率方面仍存在不足。

具体而言，在结合特征信息方面，现有方法主要关注于通过不同方式建模特征之间的交互关系<sup>[37,86,131]</sup>。一类方法通过专家手动筛选组合特征，以显式的方式进行特征交叉，然后将交叉特征传入预测模型如逻辑回归模型<sup>[14]</sup>或深度神经网络<sup>[87]</sup>。显然这种方式需要大量的人工成本和领域知识，缺乏在不同领域应用的灵活性。第二类方法通过自动组合特征的方式进行特征交叉。代表性的方法包括自动机器学习<sup>[75]</sup>和因子分解机<sup>[28,47,132]</sup>。其中基于因子分解机的方法由于其灵活性和通用性得到了广泛的应用和发展。然而，目前关于因子分解机的研究通常侧重于对于高阶特征交互的挖掘，如引入非线性层<sup>[28,88]</sup>、注意力机制<sup>[34]</sup>、卷积神经网络<sup>[37,89]</sup>等。现有因子分解机方法在训练过程中仍需结合负采样策略进行学习<sup>[90-91]</sup>，这使得其表现容易收到负采样缺点的影响，无法达到最佳的效果。同时，高阶神经网络的引入也使得基于深度学习的因子分解机模型复杂度更高，限制了在大规模现实数据场景下的应用。针对以上问题，本章提出了一个基于非采样因子分解机的高效特征增强推荐模型 ENSFM (Efficient Non-Sampling Factorization Machine)，通过严谨的数学推理将因子分解机重构为广义矩阵分解的形式，并利用高效非采样学习来全面考虑用户偏好与特征信息之间的关联，提升推荐效果。在现实数据集上的实验表明，所提出的 ENSFM 模型相对于已有结合特征信息的推荐模型在精准

度和训练效率上均有显著提升。

在结合知识图谱信息方面, 现有研究中一类方法直接利用知识图谱的网络结构, 通过随机游走等方式来学习图谱上实体间的联系来改进推荐方法<sup>[92-95,133]</sup>, 另一类则是从知识图谱中学习用户或物品的额外特征, 然后把它结合到推荐模型中<sup>[15,29,96,134]</sup>。然而, 这两类方法均存在一定的局限性。第一类方法的效率和效果往往受到随机游走的不确定性限制, 且依赖于人工定义游走的路径规则; 第二类方法大都采用了图谱表示学习和推荐任务分别训练的策略, 如交替训练每个任务<sup>[96]</sup>或是训练完成一个任务后再训练另一个<sup>[15]</sup>, 这降低了每个任务之间的协同性。同时, 现有工作广泛依赖于负采样学习, 对算法的优化比较有限, 面临效率问题以及负采样的不稳定问题。针对以上问题, 本章提出了一个基于非采样联合学习的高效知识增强推荐模型 JNSKR (Jointly Non-Sampling learning model for Knowledge graph enhanced Recommendation)。为了应对非采样学习及知识增强推荐方法的效率挑战, 我们首先设计了针对知识图谱表示学习的优化算法, 以数学推导的方法实现了对于三元组表示学习的高效非采样优化。随后, 结合前文所设计的针对推荐算法的非采样优化训练方法, 将知识图谱表示和推荐任务进行联合学习, 以充分建模用户、物品和知识图谱实体之间的细粒度联系。在现实数据集上的实验表明, 所提出的 JNSKR 模型相对于已有知识增强的推荐模型在精准度和训练效率上均有显著提升。

本章的结构如下: 第5.2节介绍相关工作; 第5.3节介绍结合特征信息的高效推荐建模问题、所设计的 ENSFM 模型以及相关实验; 第5.4节介绍结合知识图谱的高效推荐建模问题、所设计的 JNSKR 模型以及相关实验; 最后, 第5.5节对本章进行小结。

## 5.2 相关工作

### 5.2.1 结合特征信息的推荐

结合特征信息的推荐旨在利用丰富的用户和物品属性特征如用户年龄、性别以及物品类别、价格等来提升推荐的精准度。对于特征信息的使用, 现有的研究主要是通过不同方式建模特征之间的交互关系, 可分为以下两种<sup>[37,86]</sup>: (1) 通过专家手动筛选组合特征, 以显式的方式进行特征交叉, 然后将交叉特征传入预测模型如逻辑回归模型<sup>[14]</sup>或深度神经网络<sup>[87]</sup>。显然这种方式需要大量的人工成本、专业工程技术和领域知识, 缺乏在不同领域应用的灵活性; (2) 通过自动组合特征的方式进行特征交叉。代表性的方法包括基于自动机器学习 (Auto Machine Learning)<sup>[75]</sup>和基于因子分解机的方式<sup>[28,47,135]</sup>。因子分解机<sup>[47]</sup>将特征两两交互到

统一的框架下来建模特征间的关系, 由于其灵活性和通用性得到了广泛的应用。随着近年来深度学习的发展, 出现了许多结合神经网络来增强因子分解的研究工作。如引入非线性层建模高阶特征交互的 NFM<sup>[28]</sup>和 DeepFM<sup>[88]</sup>, 使用注意力机制来自动学习交互特征权重的 AFM<sup>[34]</sup>、以及通过卷积神经网络来对特征外积进行交互的 CFM<sup>[37]</sup>和 xDeepFM<sup>[89]</sup>等。虽然有关因子分解机的研究工作很多, 但大都集中于对模型结构和特征交互方式的探索, 在推荐系统优化过程中仍需结合负采样策略进行学习<sup>[90-91]</sup>, 这使得这些方法的表现容易收到负采样缺点的影响, 无法达到最佳的效果。同时, 高阶神经网络的引入也使得基于深度学习的因子分解机模型复杂度更高, 限制了在大规模现实数据场景下的应用。

### 5.2.2 结合知识图谱的推荐

在推荐任务上, 知识图谱的引入能够帮助建立起用户和待推荐物品之间的额外联系, 从而更精准地刻画用户的偏好以及物品的属性, 提升推荐效果。现有的结合知识图谱的推荐方法主要可以分为以下两类: (1) 直接利用知识图谱的网络结构, 通过随机游走等策略来利用图谱上的元路径等实体间的联系来改进现有推荐方法<sup>[92-95]</sup>。如 Yu et al.<sup>[92]</sup>提出基于不同物品在知识图谱上的元路径相似度来计算物品相似度, 然后在推荐过程中优先考虑与用户已购买过物品相似的物品。Wang et al.<sup>[95]</sup>设计了结合长短期记忆网络 (Long Short-Term Memory, LSTM) 的结合元路径的知识增强推荐方法。Ma et al.<sup>[130]</sup>设计了基于规则的知识增强推荐算法, 能够与不同的推荐算法有效结合。这类方法主要优势在于保留了知识图谱上的结构信息, 可以很方便地生成可解释的推荐理由, 并且提升推荐算法的表现。其不足之处在于效率和效果往往受到随机游走的不确定性限制, 且依赖于人工定义游走的路径规则; (2) 从知识图谱中学习用户或物品的额外特征, 然后把它结合到推荐模型中<sup>[15,29,96,136]</sup>。如 Zhang et al.<sup>[137]</sup>使用 TransR 方法<sup>[138]</sup>来从知识图谱上学习物品表示, 然后应用到协同过滤方法中。Wang et al.<sup>[15]</sup>提出了一个基于多任务特征学习的知识增强推荐方法。此外, 还有不少工作在使用知识图谱的过程中结合了知识图谱上的高阶特征, 如 KGCCN<sup>[139]</sup>和 KGAT<sup>[29]</sup>分别使用图卷积神经网络和图注意力网络来改进知识图谱表示学习, 并应用到推荐任务中。这类方法的主要优势在于能够从知识图谱中直接获得物品和用户的表示, 可以方便的与现有的神经网络推荐方法相结合。不足之处是现有工作大都采用了图谱表示学习和推荐任务分别训练的策略, 如交替训练每个任务 (Alternative Learning)<sup>[96]</sup>或是训练完成一个任务后再训练另一个 (One-by-one Learning)<sup>[15]</sup>, 这降低了每个任务之间的协同性。同时, 现有工作广泛依赖于负采样学习, 对算法的优化比较有限, 面临效率问题以及负采样的不稳定问题。

表 5.1 结合特征信息的高效推荐建模中所用数学符号

符号	描述
$\mathbf{U}, \mathbf{V}$	分别代表用户集合和物品集合
$\mathbf{B}$	一批训练用户
$\mathbf{X}$	特征集合
$\mathbf{Y}$	用户—物品交互
$\mathcal{Y}$	正向用户—物品交互
$\mathbf{x}$	稀疏特征输入
$\mathbf{e}_i$	特征 $i$ 的向量表示
$\mathbf{h}$	预测层向量
$\mathbf{p}_u$	用户 $u$ 的辅助向量
$\mathbf{q}_v$	物品 $v$ 的辅助向量
$\mathbf{h}_{aux}$	预测层辅助向量
$w_i$	一阶特征权重
$w_0$	全局预测偏差
$c_{uv}$	训练样例 $y_{uv}$ 的权重
$m$	用户特征数量
$n$	物品特征数量
$d$	向量维度
$\Theta$	神经网络参数集合

### 5.3 结合特征信息的高效推荐建模

在本节对于结合特征信息的高效推荐建模进行阐述，包括问题定义、解决方案、以及相关实验。

#### 5.3.1 问题定义

表5.1中展示了本节中用到的符号和关键概念。假设数据集中有  $M$  个用户和  $N$  个物品，用户集合记为  $\mathbf{U}$ ，物品集合记为  $\mathbf{V}$ 。我们使用索引  $u$  表示用户，使用  $v$  表示物品。用户—物品交互矩阵记为  $\mathbf{Y} = [y_{uv}]_{M \times N} \in \{0, 1\}$ ，表示  $u$  是否与物品  $v$  有过交互。 $\mathbf{x}$  表示稀疏特征输入向量，如下所示：

$$\begin{array}{ccccccc}
 & \text{用户特征} & & & \text{物品特征} & & \\
 \underbrace{[0, 1, 0, \dots, 0]}_{\text{用户编号}} & \underbrace{[1, 0]}_{\text{性别}} & \underbrace{[0, 1, \dots, 0]}_{\text{所属机构}} & \underbrace{[0, 0, 1, \dots, 0]}_{\text{物品编号}} & \underbrace{[0, 1, 0, 1, \dots, 0]}_{\text{种类}} & & \\
 \end{array}$$

$m$  和  $n$  分别表示用户特征和物品特征的数量。在此设置下，结合特征信息的推荐任务定义如下：

**输入：** 用户集  $\mathbf{U}$ ，物品集  $\mathbf{V}$ ，用户—物品交互记录  $\mathbf{Y}$ ，和特征集合  $\mathbf{X}$ 。

**输出：** 给定一个用户，为其给出按照与物品交互概率排序得到的物品推荐列表。

### 5.3.2 预备知识

我们首先介绍一下因子分解机模型<sup>[47]</sup>，因子分解机既具有灵活的特征工程优势，又保留了隐语义模型的高精度预测。给定输入向量  $\mathbf{x}$ ，因子分解机通过对每两个特征之间的所有交互进行建模来预测目标分值：

$$\hat{y}_{FM}(\mathbf{x}) = w_0 + \sum_{i=1}^{m+n} w_i x_i + \sum_{i=1}^{m+n} \sum_{j=i+1}^{m+n} \mathbf{e}_i^T \mathbf{e}_j \cdot x_i x_j \quad (5.1)$$

其中  $w_0$  是全局偏差； $w_i$  是一阶特征权重； $\mathbf{e}_i^T \mathbf{e}_j$  项表示二阶特征交互； $\mathbf{e}_i \in \mathbb{R}^d$  是特征  $i$  的表示向量； $d$  表示向量维度。该计算公式可以重构为以下形式：

$$\hat{y}_{FM}(\mathbf{x}) = w_0 + \sum_{i=1}^{m+n} w_i x_i + \frac{1}{2} \sum_{f=1}^d \left( \left( \sum_{i=1}^{m+n} \mathbf{e}_{i,f} x_i \right)^2 - \sum_{i=1}^{m+n} \mathbf{e}_{i,f}^2 x_i^2 \right) \quad (5.2)$$

该形式下每一个训练样例具备线性的计算复杂度  $O((m+n)d)$ 。

如前文所述，负采样学习过程中可能会存在训练不足、选择有偏等问题，限制推荐系统的精准度。因此，本节提出以非采样学习的方式训练因子分解机模型。然而，传统的非采样损失函数（公式 (3.2)）优化因子分解机的复杂度为  $O((m+n)|\mathbf{U}||\mathbf{V}|d)$ ，不适用于现实大规模数据场景。而本文在第三章中所设计的高效非采样算法是针对矩阵分解形式下的，只考虑了用户和物品的编号交互，无法直接应用于因子分解机模型。本研究中我们新设计了一个新的高效非采样分解机模型来解决上述问题。

### 5.3.3 基于非采样因子分解机的高效特征增强推荐模型

本节详细阐述我们提出的高效非采样分解机模型 ENSFM，其同时具备因子分解机和非采样学习的优势。首先介绍模型概览，然后论证如何将因子分解机重构为广义矩阵分解的形式。在此基础上，我们提出高效分采样分解机学习算法。最后，对 ENSFM 的学习过程、泛化和复杂性进行了讨论。

#### 5.3.3.1 模型概览

ENSFM 的目标是在非采样情况下高效学习因子分解机模型，从而全面考虑用户偏好与特征信息之间的关联，提升推荐效果。ENSFM 形式上遵循广义的基于向

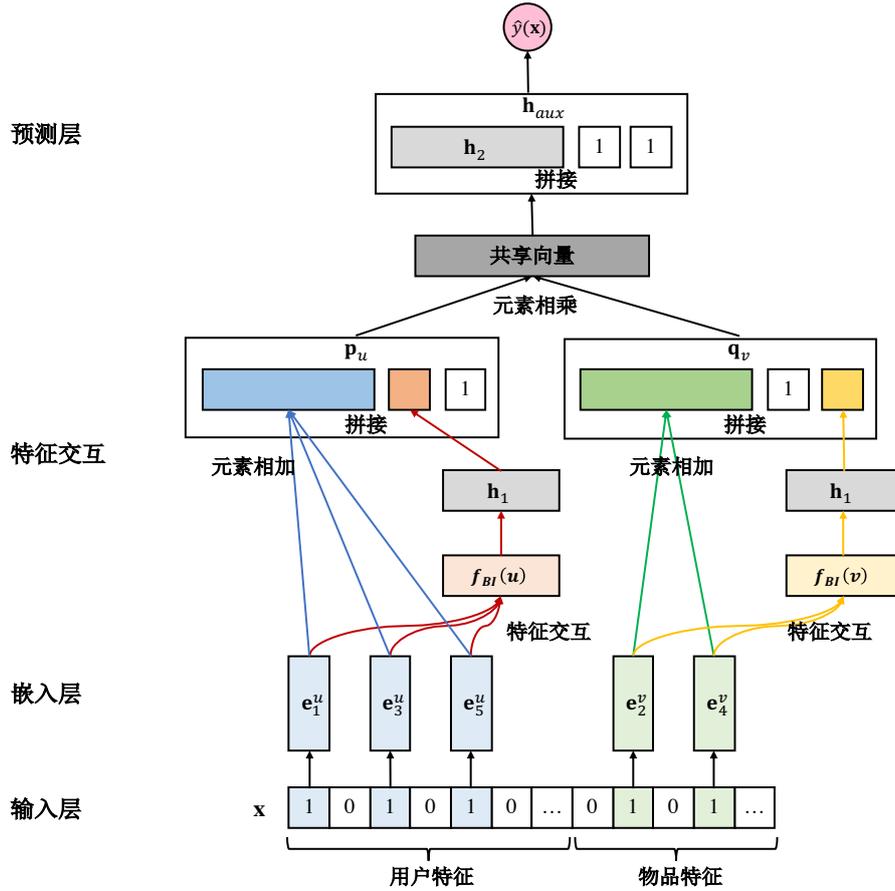


图 5.1 基于非采样因子分解机的高效特征增强推荐模型 (ENSFM) 示意图

量的因子分解机<sup>[28,34]</sup>：

$$\hat{y}_{FM}(\mathbf{x}) = w_0 + \sum_{i=1}^{m+n} w_i x_i + \underbrace{\mathbf{h}^T \sum_{i=1}^{m+n} \sum_{j=i+1}^{m+n} (x_i \mathbf{e}_i \odot x_j \mathbf{e}_j)}_{f(\mathbf{x})} \quad (5.3)$$

其中  $\odot$  表示两个向量之间的元素相乘， $\mathbf{h}$  表示预测层，为向量空间的不同维度分配相应的权重。

ENSFM 模型的示意图如图5.1所示。首先对模型整体做一简要介绍：

1. ENSFM 模型的输入首先通过嵌入层转换为稠密向量，具体来说，用户特征和物品特征分别表示为  $\mathbf{e}^u$  和  $\mathbf{e}^v$ 。输出  $\hat{y}_{FM}(\mathbf{x})$  是一个预测分数，表示用户  $u$  对物品  $v$  的偏好。
2. ENSFM 将公式 (5.3) 重构为广义矩阵分解的形式： $\hat{y}_{FM}(\mathbf{x}) = \mathbf{h}_{aux}^T (\mathbf{p}_u \odot \mathbf{q}_v)$ ，其中  $\mathbf{p}_u$ ， $\mathbf{q}_v$ ，和  $\mathbf{h}_{aux}$  为新构建的辅助向量。
3. ENSFM 模型采用高效非采样算法进行优化，在参数更新时考虑了所有训练样本。

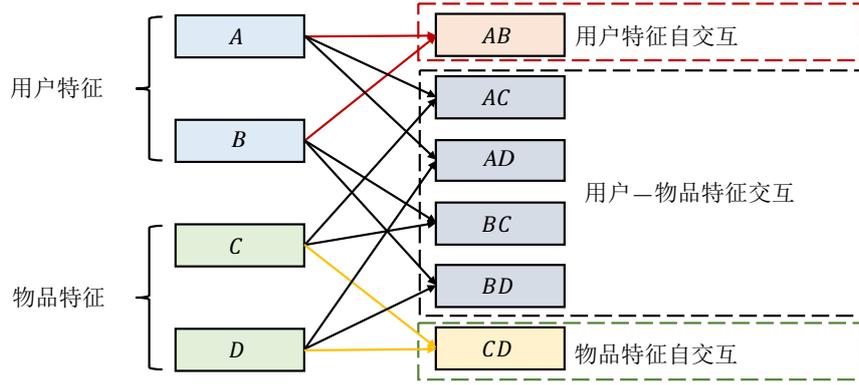


图 5.2 因子分解机中的特征交互示意图

### 5.3.3.2 理论分析

我们首先在本小节给出如下定理和证明：

**定理 5.1：** 广义因子分解机的预测函数（公式 (5.3)）可以重新表示为如下矩阵分解的形式：

$$\hat{y}_{FM}(\mathbf{x}) = \mathbf{h}^T (\mathbf{p}_u \odot \mathbf{q}_v) \quad (5.4)$$

其中  $\mathbf{p}_u$ ,  $\mathbf{q}_v$  分别表示用户  $u$  和物品  $v$  的特征向量。

定理5.1在两种最流行的推荐方法——矩阵分解（MF）和因子分解机（FM）之间建立了联系。接下来，在基于一个广义因子分解机模型证明该定理的同时，阐述我们所提出的 ENSFM 方法。

**证明：** 回顾公式 (5.3) 中的二阶特征交互  $f(\mathbf{x})$ ，其可以重新表示为：

$$f(\mathbf{x}) = \mathbf{h}_1^T \left( \underbrace{\sum_{i=1}^m \sum_{j=i+1}^m (x_i^u \mathbf{e}_i^u \odot x_j^u \mathbf{e}_j^u)}_{f_{BI}(u)} + \underbrace{\sum_{i=1}^n \sum_{j=i+1}^n (x_i^v \mathbf{e}_i^v \odot x_j^v \mathbf{e}_j^v)}_{f_{BI}(v)} \right) + \mathbf{h}_2^T \left( \sum_{i=1}^m x_i^u \mathbf{e}_i^u \odot \sum_{i=1}^n x_i^v \mathbf{e}_i^v \right) \quad (5.5)$$

其中  $f_{BI}(u)$  和  $f_{BI}(v)$  分别表示用户特征自交互和物品特征自交互（如图5.2所示）。注意公式 (5.3) 预测层参数  $\mathbf{h}$  可以扩展为  $\mathbf{h}_1$  和  $\mathbf{h}_2$ ，分别为特征自交互以及用户-物品特征交互分配不同的权重。该设置使得 ENSFM 具有更好的泛化能力。更进一步的扩展如使用注意力机制<sup>[4,34]</sup>等也同样适用。

如图5.2所示，用户特征自交互独立于物品特征自交互。即用户在跟不同物品交互时其特征自交互部分是保持一致的，物品特征自交互部分与之同理。因此这两部分可以预先表示出来以加速计算过程。具体地，构建三个辅助向量  $\mathbf{p}_u \in \mathbb{R}^{d+2}$ ,

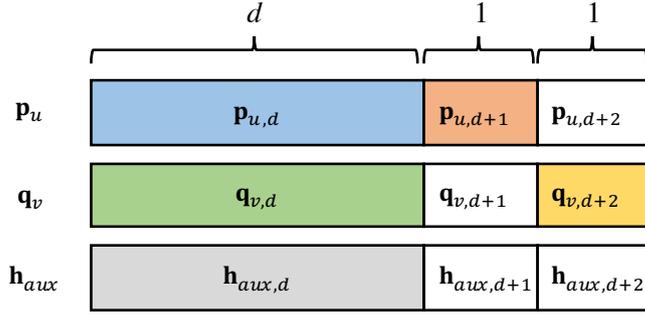


图 5.3 ENSFM 中辅助向量示意图

$\mathbf{q}_v \in \mathbb{R}^{d+2}$ , 和  $\mathbf{h}_{aux} \in \mathbb{R}^{d+2}$  来分别表示用户  $u$ , 物品  $v$ , 以及预测层参数 (如图5.3所示):

$$\mathbf{p}_u = \begin{bmatrix} \mathbf{p}_{u,d} \\ \mathbf{p}_{u,d+1} \\ \mathbf{p}_{u,d+2} \end{bmatrix}; \mathbf{q}_v = \begin{bmatrix} \mathbf{q}_{v,d} \\ \mathbf{q}_{v,d+1} \\ \mathbf{q}_{v,d+2} \end{bmatrix}; \mathbf{h}_{aux} = \begin{bmatrix} \mathbf{h}_{aux,d} \\ \mathbf{h}_{aux,d+1} \\ \mathbf{h}_{aux,d+2} \end{bmatrix} \quad (5.6)$$

其中

$$\mathbf{p}_{u,d} = \sum_{i=1}^m x_i^u \mathbf{e}_i^u; \mathbf{p}_{u,d+1} = \mathbf{h}_1^T f_{BI}(u) + w_0 + \sum_{i=1}^m w_i^u x_i^u; \mathbf{p}_{u,d+2} = 1 \quad (5.7)$$

$$\mathbf{q}_{v,d} = \sum_{i=1}^n x_i^v \mathbf{e}_i^v; \mathbf{q}_{v,d+1} = 1; \mathbf{q}_{v,d+2} = \mathbf{h}_1^T f_{BI}(v) + \sum_{i=1}^n w_i^v x_i^v \quad (5.8)$$

$$\mathbf{h}_{aux,d} = \mathbf{h}_2; \mathbf{h}_{aux,d+1} = 1; \mathbf{h}_{aux,d+2} = 1 \quad (5.9)$$

此时, 广义因子分解机的预测函数可以重新表示为如下矩阵分解的形式:

$$\hat{y}_{FM}(\mathbf{x}) = \mathbf{h}_{aux}^T (\mathbf{p}_u \odot \mathbf{q}_v) \quad (5.10)$$

其中  $\mathbf{p}_u$  仅取决于用户  $u$  的特征, 而  $\mathbf{q}_v$  仅取决于物品  $v$  的特征。当设置  $\mathbf{h}_1 = \mathbf{h}_2 = \mathbf{h}$  时, 公式 (5.10) 的结果与公式 (5.3) 结果完全相同。 ■

### 5.3.3.3 高效非采样训练

在本小节介绍 ENSFM 的高效训练过程, 首先进行如下分析:

公式 (5.5) 中的  $f_{BI}(u)$  和  $f_{BI}(v)$  可以重写为以下形式<sup>[28,47]</sup> 以实现线性复杂度, 以  $f_{BI}(u)$  为例:

$$f_{BI}(u) = \frac{1}{2} \left( \left( \sum_{i=1}^m x_i^u \mathbf{e}_i^u \right)^2 - \sum_{i=1}^m (x_i^u \mathbf{e}_i^u)^2 \right) \quad (5.11)$$

$f_{BI}(u)$  时间复杂度为  $O(md)$ , 而  $f_{BI}(v)$  为  $O(nd)$ 。

**算法 5.1** ENSFM 训练流程

**Require:** 训练集  $\{\mathbf{Y}, \mathbf{U}, \mathbf{V}, \mathbf{X}\}$ ; 训练样例权重  $c$ ; 学习率  $\eta$ ; 向量长度  $d$

**Ensure:** 神经网络模型参数  $\Theta$

- 1: 随机初始化模型参数  $\Theta$
- 2: **while** 结束条件尚未满足 **do**
- 3:   选出一批训练物品及相应训练样例  $\{\mathbf{Y}_{\mathbf{B}}, \mathbf{B}, \mathbf{V}, \mathbf{X}\}$
- 4:   为用户构建辅助向量  $\mathbf{P}_{\mathbf{B}}$  (公式 (5.6), (5.7))
- 5:   为物品构建辅助向量  $\mathbf{Q}$  (公式 (5.6), (5.8))
- 6:   构建辅助向量  $\mathbf{h}_{aux}$  (公式 (5.6), (5.9))
- 7:   计算损失函数  $\tilde{\mathcal{L}}(\Theta)$  (公式 (5.12))
- 8:   更新模型参数
- 9: **end while**
- 10: **return**  $\Theta$

同时, 定理5.1的证明表明, 公式 (5.10) 中的  $\mathbf{p}_u$  和  $\mathbf{q}_v$  彼此独立。因此可以通过预计算辅助向量来避免大量重复计算以实现加速。

最后, 在辅助向量的帮助下, ENSFM 的预测被重新表示为矩阵分解的形式, 它满足定理3.1的要求。因此, 我们对一批用户的高效非采样损失函数可以通过如下方式计算:

$$\begin{aligned} \tilde{\mathcal{L}}(\Theta) = & \sum_{u \in \mathbf{B}} \sum_{v \in \mathbf{V}^+} ((c_v^+ - c_v^-) \hat{y}(\mathbf{x})^2 - 2c_v^+ \hat{y}(\mathbf{x})) \\ & + \sum_{i=1}^d \sum_{j=1}^d \left( (h_{aux,i} h_{aux,j}) \left( \sum_{u \in \mathbf{B}} p_{u,i} p_{u,j} \right) \left( \sum_{v \in \mathbf{V}} c_v^- q_{v,i} q_{v,j} \right) \right) \end{aligned} \quad (5.12)$$

其中  $\mathbf{B}$  表示一批用户,  $c_v$  表示训练样本  $y_{uv}$  的权重。具体的训练流程参见算法5.1

此外, 为了更好地训练模型, 我们采用 Adagrad<sup>[105]</sup> 作为模型的优化器, 它的主要优势在于会自适应地调整训练过程中的学习率, 选择合理的步长来对解空间进行搜索, 从而避免了对于学习率这个超参数过多的调参过程。去除函数 (Dropout) 也被加入到模型中用来避免训练出现过拟合现象<sup>[123]</sup>, 我们的随机去除主要是针对特征交互后的共享向量进行的, 去除操作之后, 保留比例为  $\rho$  的向量参数。

### 5.3.3.4 讨论

首先对 ENSFM 的复杂度进行分析。ENSFM 的计算复杂度可分为两部分, 第一部分构建辅助向量, 第二部分是高效非采样学习。对一批训练数据, 构建辅助向量  $\mathbf{P}_{\mathbf{B}}$  花费  $O(m|\mathbf{B}|d)$ ,  $\mathbf{Q}$  花费  $O(n|\mathbf{V}|d)$ 。这里  $\mathbf{P}_{\mathbf{B}}$  和  $\mathbf{Q}$  可以随着损失函数同步更新, 因此, 整体一批训练数据的计算复杂度为  $O((m|\mathbf{B}| + n|\mathbf{V}|)d^2 + |\mathcal{Y}_{\mathbf{B}}|d)$ 。其中  $\mathcal{Y}_{\mathbf{B}}$  表示每批训练数据中实际存在的用户-物品正向交互。对于传统非采样方法的复杂度为  $O((m+n)|\mathbf{B}||\mathbf{V}|d)$ , 考虑到现实场景中  $|\mathcal{Y}_{\mathbf{B}}| \ll |\mathbf{B}||\mathbf{V}|$ , 且  $d \ll |\mathbf{B}|$ , 优

表 5.2 评估 ENSFM 模型的数据集统计信息

数据集	用户数量	物品数量	特征数量	用户—物品交互数量	特征种类数量
<i>Frappe</i>	957	4,082	5,382	96,203	10
<i>Last.fm</i>	1,000	20,301	37,358	214,574	4
<i>Movielens</i>	6,040	3,706	10,021	1,000,209	6

化后的 ENSFM 较传统的非采样训练方法有了数量级的效率提升，使得通过非采样方法训练因子分解机成为可能。

其次，ENSFM 对几种现有的结合特征的推荐方法有很好的泛化性<sup>[28,47,140]</sup>。具体地，通过将  $\mathbf{h}_1$  和  $\mathbf{h}_2$  固定为常数向量  $(1, \dots, 1)$ ，可以转换为传统因子分解机模型 FM<sup>[47]</sup>；通过设置  $\mathbf{h}_1 = \mathbf{h}_2$ ，可以转换为一层深度的 NFM 模型<sup>[28]</sup>；通过将  $\mathbf{h}_1$  固定到  $(0, \dots, 0)$ ， $\mathbf{h}_2$  固定为  $(1, \dots, 1)$ ，可以转换为 SVDFeature 方法<sup>[140]</sup>。除了上述差异，ENSFM 的关键在于提出了一种高效的非采样优化算法，提供了一种新的改进因子分解机的思路。

### 5.3.4 实验

本实验所用代码已开源<sup>①</sup>。

#### 5.3.4.1 数据集介绍

我们选择三个被广泛使用的推荐系统公开数据集进行实验，分别是 *Frappe*<sup>②</sup>，*Last.fm*<sup>③</sup>和 *Movielens*<sup>④</sup>。对三个数据集的简单介绍如下：

- ***Frappe***：Frappe 是一个手机应用（App）发现工具。该数据集由<sup>[141]</sup>建立，包含 96,203 个手机应用记录以及用户和应用的 10 个特征数据，如用户的编号、年龄等信息。
- ***Last.fm***：Last.fm 是一个常用的音乐推荐的数据集，我们实验中使用了包含有 1,000 个用户的数据集版本。所包含的用户特征包括用户的编号以及最后一个听的音乐编号，包含的物品特征为音乐编号和创作者。
- ***Movielens***：Movielens 是一个常用的电影推荐数据集，我们实验中使用了包含有一百万个交互的数据集版本，且相应的评分也被转换为隐式数据。所包含的用户特征有编号、性别、年龄等，物品特征包括电影编号以及种类。

① <https://github.com/chenchongthu/ENSFM>

② <http://baltrunas.info/research-menu/frappe>

③ <https://grouplens.org/datasets/hetrec-2011/>

④ <https://grouplens.org/datasets/movielens/1m/>

表 5.3 ENSFM 与对比方法的特点比较

模型特点	MP	ONCF	ENMF	FM	NFM	DeepFM	CFM	ENSFM
用户—物品交互	\	√	√	√	√	√	√	√
特征信息	\	\	\	√	√	√	√	√
神经网络模型	\	√	√	\	√	√	√	√
非采样学习	\	\	√	\	\	\	\	√

三个数据集的基本统计信息见表5.2。

### 5.3.4.2 基准方法

为了验证所提出的 ENSFM 模型的有效性，我们与以下现有的推荐算法进行了比较：

- **Most Popular (MP)**: 最简单的非个性化推荐方法，根据物品的流行度对物品进行排序。
- **ONCF**<sup>[36]</sup>: 外积神经网络协同过滤模型，通过外积和卷积神经网络建模用户和物品交互信息。
- **ENMF**<sup>[49]</sup>: 本文在第三章所提出的高效神经网络矩阵模型，通过高效非采样学习从整体数据上优化模型参数。
- **FM**<sup>[47]</sup>: 原始的因子分解机模型，既具有灵活的特征工程优势，又保留了隐语义模型的高精度预测。
- **NFM**<sup>[28]</sup>: 神经网络因子分解机是当前最优的融合用户和物品特征的推荐方法之一，使用多层神经网络来建模不同特征的高阶交互来提升算法表现。
- **DeepFM**<sup>[88]</sup>: 深度因子分解机模型，使用因子分解机替代 Wide&Deep 模型<sup>[14]</sup>中的 Deep 部分来建模不同特征的高阶交互。
- **CFM**<sup>[37]</sup>: 卷积分解机模型，通过外积和卷积神经网络建模高阶特征交互，是当前因子分解机中最先进的拓展模型之一。

除了 MP 以外的模型均使用 TensorFlow<sup>①</sup>实现。对于 FM, NFM, ONCF 和 CFM, 我们使用工作<sup>[37]</sup>中所公开的代码<sup>②</sup>。对于 DeepFM, 我们使用原作者公开的代码<sup>[88]</sup><sup>③</sup>。

① <https://www.tensorflow.org/>

② <https://github.com/chenboability/CFM>

③ <https://github.com/ChenglongChen/tensorflow-DeepFM>

### 5.3.4.3 评价方式

在模型评价过程中，我们将使用不同方法为每个用户对所有的候选物品进行排序（除了训练集和验证集已知的有正向交互的物品）<sup>[107]</sup>。同时，我们采用了在推荐系统评价中被广泛使用的留一法来进行方法评价<sup>[13,20]</sup>。具体地，对于每个用户对于物品的交互记录，我们保留最后一个交互作为测试数据，其他的所有交互用做训练。在评价指标上，我们采用命中率（HR）和归一化的累积增益（NDCG）。HR@K 主要评价所预测的正例是否在推荐列表中的前 K 个，而 NDCG@K 则还会考虑所预测的正例在推荐列表中的具体位置，越靠前则得分越高。评价指标的计算方式如公式（3.12）所示。实验过程中我们对所有方法使用不同的随机数种子重复运行 5 次，然后汇报各个方法取得的平均表现。

### 5.3.4.4 超参数设置

所有基准算法的参数设定都是根据对应论文的初始化要求设置的，且也都进行了调参以确保其达到最优的效果。我们对超参数的主要搜索空间包括：学习率的搜索范围为 [0.005, 0.01, 0.02, 0.05]，为了避免过拟合，网络留存率（Dropout）的搜索范围为 [0.0, 0.1, ..., 0.9]，特征向量长度  $d$  的搜索范围为 [8, 16, 32, 64]，训练样本权重  $c_0$  的搜索范围为 [0.005, 0.01, 0.05, 0.1, 0.2, 0.5, 1]。最终，根据预实验结果，每个批的大小被设置成 512，学习率被设置成 0.05，特征向量的长度被设置成 64。基于卷积神经网络的模型 ONCF 和 CFM 的输出大小设为 32，NFM 的层数设为 1，DeepFM 的层数设为 3，网络留存率比例设为 0.9，训练样本权重  $c_0$  对 Frappe, Last.fm 和 Movielens 分别设为 0.05, 0.005 以及 0.5。

### 5.3.4.5 整体表现分析

本文所提出的 ENSFM 方法与其他基准方法在三个数据集上的推荐表现见表 5.4，为了对比不同长度推荐列表的效果，我们汇报了推荐列表长度为 5、10 和 20 的结果。值得一提的是，对于 Frappe 和 Last.fm 数据集，FM, DeepFM, NFM, ONCF, 和 CFM 的实验结果与在工作<sup>[37]</sup>中汇报的完全一致，因为我们采用了相同的数据集和实验设置。从实验结果中我们有以下发现：

首先，我们所提出的 ENSFM 在三个数据集上均表现出最好的效果，显著优于所有基线方法。具体而言，与 CFM（最近提出的最先进的基于深度学习的因子分解机模型）相比，ENSFM 在三个数据集上的相对提升分别为 9.15%、48.05% 和 20.22%。作为基于浅层神经网络的 ENSFM 参数量显著少于 CFM，但是非采样训练方法的应用能够让 ENSFM 在整个数据集上进行优化，因此取得了更好的效果。

表 5.4 ENSFM 与不同推荐方法的效果比较, \*\* 表示较最好的基准方法有显著提升, 显著性是  $p < 0.01$ 。“RI” 表示 ENSFM 相较对应方法的相对提升

<i>Frappe</i>	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20	RI
MP	0.2539	0.3493	0.4136	0.1595	0.1898	0.2060	+143.3%
FM	0.4204	0.5486	0.6590	0.3054	0.3469	0.3750	+39.86%
DeepFM	0.4632	0.6035	0.7322	0.3308	0.3765	0.4092	+27.77%
NFM	0.4798	0.6197	0.7382	0.3469	0.3924	0.4225	+23.64%
ONCF	0.5359	0.6531	0.7691	0.3940	0.4320	0.4614	+13.24%
CFM	0.5462	0.6720	0.7774	0.4153	0.4560	0.4859	+9.15%
ENMF	0.5682	0.6833	0.7749	0.4314	0.4642	0.4914	+6.95%
<b>ENSFM</b>	<b>0.6094**</b>	<b>0.7118**</b>	<b>0.7889**</b>	<b>0.4771**</b>	<b>0.5105**</b>	<b>0.5301**</b>	–
<i>Last.fm</i>	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20	RI
MP	0.0013	0.0023	0.0032	0.0007	0.0011	0.0013	+26566%
FM	0.1658	0.2382	0.3537	0.1142	0.1374	0.1665	+108.4%
DeepFM	0.1773	0.2612	0.3799	0.1204	0.1473	0.1772	+94.59%
NFM	0.1827	0.2678	0.3783	0.1235	0.1488	0.1765	+91.76%
ONCF	0.2183	0.3208	0.4611	0.1493	0.1823	0.2176	+58.11%
CFM	0.2375	0.3538	0.4841	0.1573	0.1948	0.2277	+48.05%
ENMF	0.3188	0.4254	0.5279	0.2256	0.2531	0.2894	+15.94%
<b>ENSFM</b>	<b>0.3683**</b>	<b>0.4729**</b>	<b>0.5793**</b>	<b>0.2744**</b>	<b>0.3082**</b>	<b>0.3352**</b>	–
<i>Movielens</i>	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20	RI
MP	0.0084	0.0308	0.0763	0.0041	0.0111	0.0227	+388.9%
FM	0.0377	0.0687	0.1164	0.0234	0.0334	0.0453	+52.32%
DeepFM	0.0413	0.0754	0.1351	0.0247	0.0365	0.0503	+38.43%
NFM	0.0421	0.0775	0.1334	0.0268	0.0381	0.0521	+33.91%
ONCF	0.0491	0.0801	0.1368	0.0301	0.0402	0.0543	+24.70%
CFM	0.0514	0.0812	0.1398	0.0318	0.0419	0.0567	+20.22%
ENMF	0.0534	0.0867	0.1523	0.0332	0.0448	0.0606	+13.10%
<b>ENSFM</b>	<b>0.0601**</b>	<b>0.1024**</b>	<b>0.1690**</b>	<b>0.0373**</b>	<b>0.0508**</b>	<b>0.0674**</b>	–

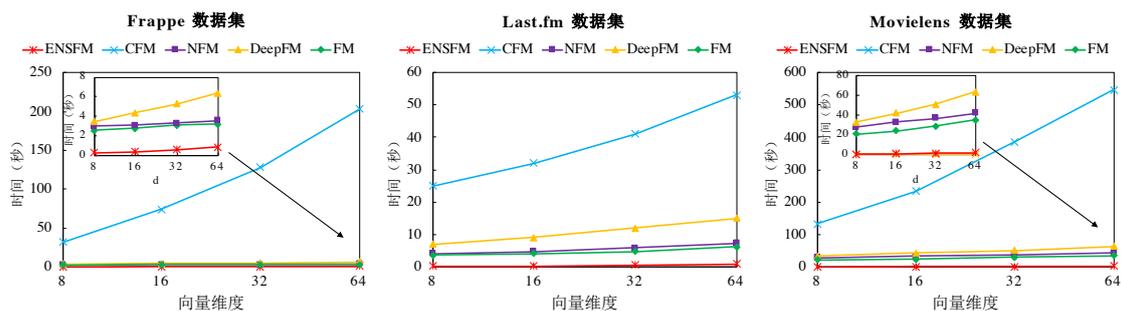


图 5.4 ENSFM, CFM, NFM, DeepFM 和 FM 在不同向量维度下的训练耗时

基于负采样的方法 (FM、DeepFM、NFM、ONCF、CFM) 只使用了一小部分采样的训练数据, 可能会忽略重要的训练样例从而降低模型表现。实验结果还显示了通过更好的学习方式来改进传统的浅层方法的潜力。同时, 实验结果中 ENSFM 的表现显著优于 ENMF, 这表明特征信息有助于帮助提升推荐效果<sup>[21,37,142]</sup>。

其次, 使用非采样学习策略的方法通常比基于负采样的方法表现更好。例如, 在表5.4中, ENMF(不使用特征信息)和 ENSFM 都比其他基线方法(NFM、DeepFM、ONCF 和 CFM) 表现更好。这与之前的工作结论一致<sup>[43,67]</sup>, 即: 非采样学习方法能够取得更优秀的模型表现。

最后, 虽然在采用相同的基于负采样的学习策略时, 基于深度学习的因子分解机方法确实比普通因子分解机方法取得了更好的性能, 但与我们的非采样的 ENSFM 相比, 所取得的提升相对较小。这显示了在 Top-N 推荐任务上, 更深的模型和更多的参数不一定会带来最好结果, 学习策略甚至比先进的神经网络结构更重要。

#### 5.3.4.6 训练效率分析

在本小节中, 我们进行实验探索所设计的 ENSFM 模型与基线模型的训练效率。所有的实验都是在同一台服务器上运行的, 其配置为: 英特尔 Xeon9 核心 CPU (2.4GHz), 使用了一张 NVIDIA GeForce GTX TITAN X GPU。

我们首先研究了不同向量维度  $d$  下 FM、DeepFM、NFM、CFM 和 ENSFM 每一轮训练的时间消耗, 结果如图5.4所示。从图中我们可以看出, ENSFM 的训练时间成本远低于其他因子分解机模型。随着向量维度  $d$  的增加, 基线方法的成本显著增加, 而 ENSFM 仍然保持着非常快的训练过程。例如, 在 Movielens 数据集上, 维度为 64 的 ENSFM 每轮训练仅需 2 秒。进一步地, 我们比较了 ENSFM 和其他因子分解机模型 FM, NFM, DeepFM, CFM 的整体训练效率, 所有方法的向量维度都设置为 64, 实验结果如表5.5所示。从表中可以明显地观察到, ENSFM 的整体训练时间比基线模型快了几个数量级。举例来说, 对于最大的数据集 Movielens,

表 5.5 ENSFM 与不同推荐模型的训练耗时情况，其中 s/m/h/d 分别代表秒/分钟/小时/天

模型	Frappe			Last.fm			Movielens		
	每轮时长	训练轮数	总训练时长	每轮时长	训练轮数	总训练时长	每轮时长	训练轮数	总训练时长
FM	3.2s	500	27m	6.2s	500	52m	35s	500	5h
NFM	3.6s	500	30m	7.3s	500	61m	42s	500	6h
DeepFM	6.4s	500	54m	15s	500	324m	64s	500	9h
CFM	203s	500	28h	54s	500	125m	9m	500	3d
ENSFM	<b>0.9s</b>	200	<b>3m</b>	<b>1.1s</b>	500	<b>10m</b>	<b>2s</b>	200	<b>7m</b>

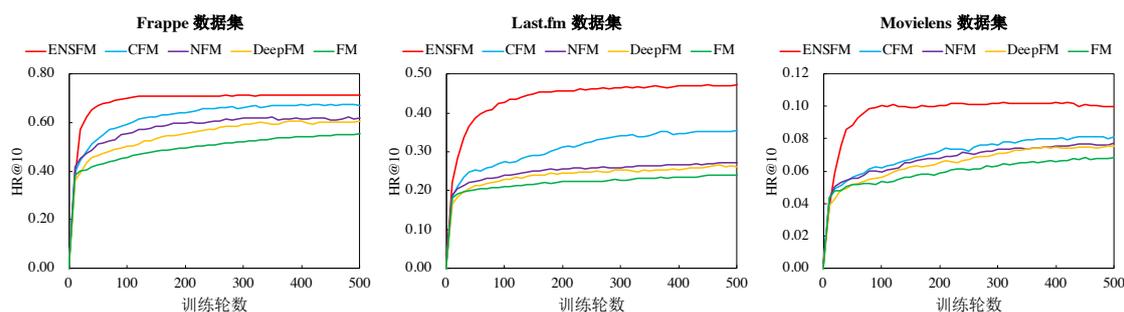


图 5.5 ENSFM, CFM, NFM, DeepFM 和 FM 随训练轮数的收敛情况

我们的 ENSFM 只需要 7 分钟就可以达到最佳性能，而基线方法 NFM、DeepFM 和 CFM 分别需要 6 小时、9 小时和 3 天才能训练完毕。加速是 NFM 的 50 倍以上，是 CFM 的 600 倍以上，体现出 ENSFM 绝对的训练优势，对于其他数据集，ENSFM 的结果同样显著。在真实的应用场景中，模型训练时间也是需要考的一个重要因素。ENSFM 方法在训练效率方面显示出明显的优势，表明了算法具有很强的实用性。

我们也通过实验展示了 ENSFM, CFM, NFM, DeepFM 和 FM 随训练轮数的收敛情况，实验结果如图 5.5 所示。为了方便，图 5.5 中以 HR@10 评价指标为例进行展示，其余评价指标上的结果与之类似。从图中，我们有如下观察：首先，我们的方法 ENSFM 的收敛速度比其他因子分解机方法快得多，并且始终保持更好的表现。原因在于其他因子分解机方法是基于负采样进行训练的，通常需要更多的训练轮数，并且很难将模型优化到最优的状态。

#### 5.3.4.7 超参数影响分析

在本节中，我们进行了实验探索不同向量维度  $d$  和不同负样本权重  $c_0$  对 ENSFM 方法的影响。值得一提的是，我们的 ENSFM 在实际应用中很容易调参，因为：(1) ENSFM 的整体训练过程非常快；(2) 与大多数现有的深度学习因子分解机方法不同，ENSFM 不需要使用 FM 进行预训练；(3) 通常只有一个超参数——负样本权重  $c_0$  需要针对不同的数据集进行调参。

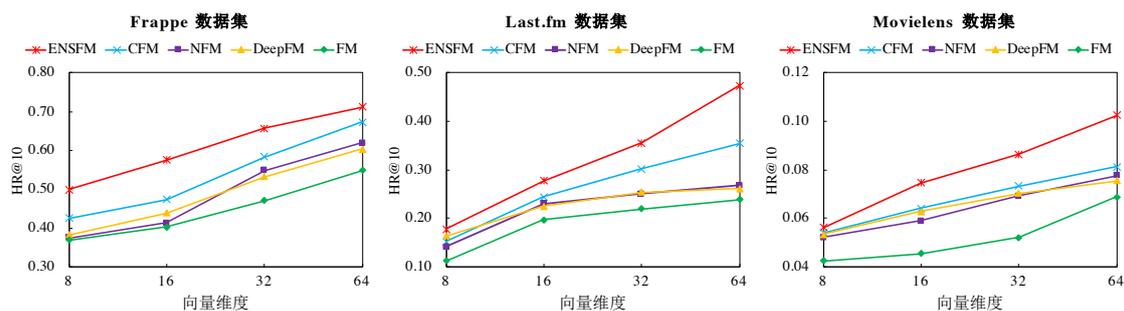


图 5.6 ENSFM, CFM, NFM, DeepFM 和 FM 随向量维度变化的表现情况

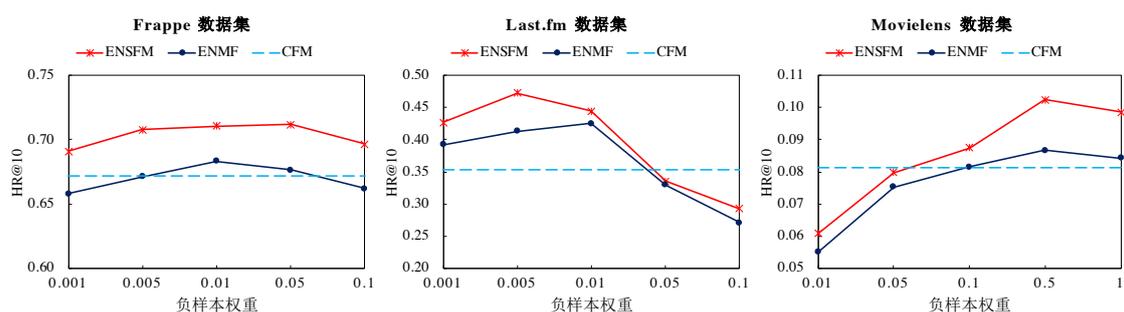


图 5.7 ENSFM, ENMF 和 CFM 随负样本权重变化的表现情况

我们首先进行实验测试向量维度  $d$  对于结果的影响。图5.6显示了 HR@10 指标随向量维度变化的表现情况。对于其他指标，观察结果一致。从该图可以看出：我们的 ENSFM 方法在不同向量维度设置下均优于其他模型。值得注意的是，向量维度为 32 的 ENSFM 甚至比向量维度为 64 的 CFM 模型表现效果更好。这进一步验证了非采样学习所带来的推荐性能优势。此外，随着向量维数的增加，所有模型的性能都会提高。这表明，更大的维度可以增加建模能力，从而学到更多用户和物品的隐含信息，提升推荐效果。这一观察结果也与之前的工作保持一致<sup>[12,20,39]</sup>。然而，对于大多数深度学习和神经网络方法，更大的向量维度也意味着需要更多的训练时间。因此，高效的神经网络学习算法对于提升推荐模型实际应用中的效率和效果意义重大。

接下来，我们进行实验测试负样本权重  $c_0$  对于基于非采样学习的推荐模型 ENMF 和 ENSFM 的效果影响，结果展示于图5.7中。CFM 的结果也添加在图中以展示 ENSFM 的效果。值得一提的是，在本节中为了方便，我们将负样本权重设置为一个固定的常数，并且已经在之前的实验中显示了优于现有推荐方法的性能。在实际使用中，也可以将权重设置为跟物品流行度相关的数值来进一步提升模型的效果。据图，我们有如下的观察：首先，对 Frappe, Last.fm, 和 Movielens 数据集,  $c_0$  的最佳设置分别为 0.05, 0.005, 和 0.5。当  $c_0$  相对峰值更小或太大时，ENMF 和 ENSFM 的性能都会有所降低。这突出了非采样学习策略中负样本权重的调节的重

表 5.6 结合知识图谱的高效推荐建模中所用数学符号

符号	描述
$\mathbf{U}, \mathbf{V}$	分别代表用户集合和物品集合
$\mathbf{E}, \mathbf{R}$	分别代表知识图谱节点和关系集合
$\mathbf{B}$	一批训练物品
$\mathbf{Y}$	用户—物品交互
$\mathcal{Y}$	正向用户—物品交互
$\mathbf{G}$	知识图谱
$\mathcal{G}$	知识图谱上所有正向三元组的集合
$c_{uv}$	训练样例 $y_{uv}$ 的权重
$w_{hrt}$	训练样例 $g_{hrt}$ 的权重
$\mathbf{p}_u, \mathbf{q}_v$	分别代表用户 $u$ 和物品 $v$ 的隐语意表示向量
$\mathbf{e}_h, \mathbf{e}_t$	分别代表实体 $h$ 和 $t$ 的隐语意表示向量
$\mathbf{r}_k$	关系 $k$ 的隐语意表示向量
$d$	向量维度
$\Theta$	神经网络参数集合

要性。其次，ENSMF 方法对负样本权重的选择具有稳定性。例如，在 Movielens 上的  $c_0$  介于 0.1 到 1 之间时始终优于最佳基线方法 CFM。最后，考虑到每个数据集的统计信息，我们发现负样本的最佳权重跟数据集的稠密度具有一定的相关性。如之前的工作<sup>[8-9]</sup>所示，稀疏数据中大部分物品未被用户交互，更可能是因为用户未看到而不是用户对其不感兴趣。因此，为稀疏数据分配更小的负样本权重是更加合理的。

## 5.4 结合知识图谱的高效推荐建模

在本节对于结合知识图谱的高效推荐建模进行阐述，包括问题定义、解决方案、以及相关实验。

### 5.4.1 问题定义

表5.6中展示了本节中用到的符号和关键概念。假设数据集中有  $M$  个用户和  $N$  个物品，用户集合记为  $\mathbf{U}$ ，物品集合记为  $\mathbf{V}$ 。我们使用索引  $u$  表示用户，使用  $v$  表示物品。用户—物品交互矩阵记为  $\mathbf{Y} = [y_{uv}]_{M \times N} \in \{0, 1\}$ ，表示  $u$  是否与物品  $v$  有过交互。此外，除了推荐系统上的用户、物品交互，我们还有知识图谱信

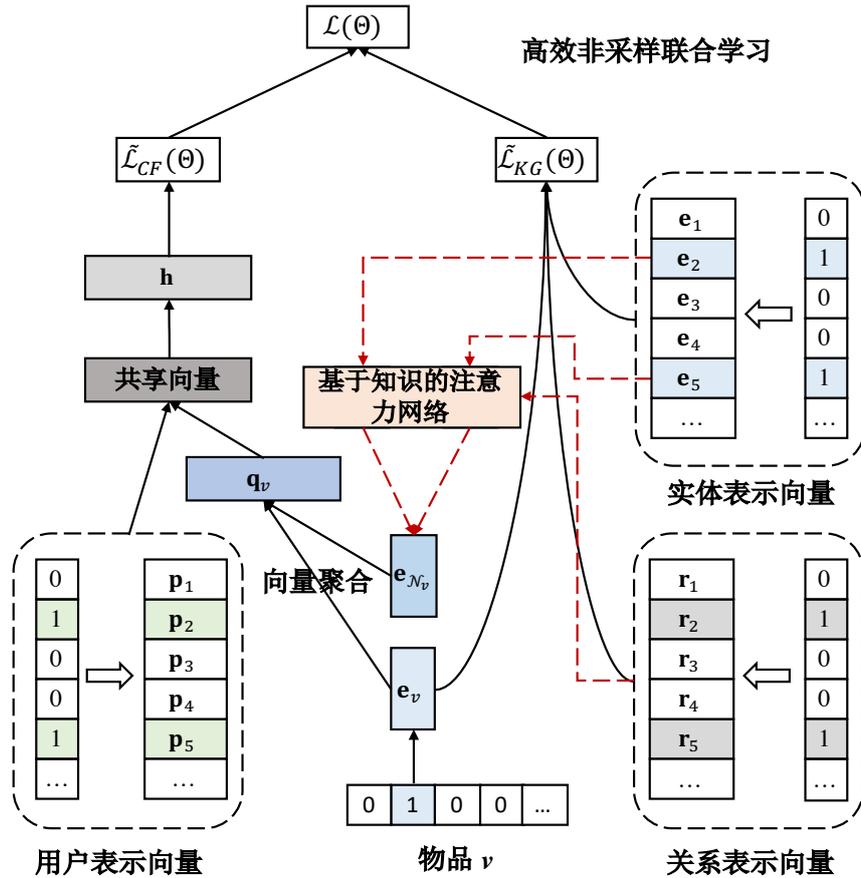


图 5.8 基于非采样联合学习的高效知识增强推荐模型 (JNSKR) 示意图

息，其中包含了实体与物品的各种关联关系。使用  $\mathbf{G} = (\mathbf{E}, \mathbf{R})$  来表示知识图谱， $\{(h, r, t) | h, t \in \mathbf{E}, r \in \mathbf{R}\}$  表示实体  $h$  和实体  $t$  间有关系为  $r$  的连边。在此设置下，结合知识图谱的推荐任务定义如下：

**输入：** 用户集  $\mathbf{U}$ ，物品集  $\mathbf{V}$ ，用户—物品交互记录  $\mathbf{Y}$ ，知识图谱  $\mathbf{G}$ 。

**输出：** 给定一个用户，为其给出按照与物品交互概率排序得到的物品推荐列表。

## 5.4.2 基于非采样联合学习的高效知识增强推荐模型

### 5.4.2.1 模型概览

本节我们将介绍提出的基于非采样联合学习的高效知识增强推荐模型 JNSKR，图5.8为模型的示意图。我们首先对模型的整体做一个简要的介绍：

1. JNSKR 模型的主要目标是在效率和效果两方面提升知识增强的推荐方法。与已有知识增强推荐方法将表示学习和推荐模型训练分为两阶段训练不同，我们提出对这两部分同时进行非采样策略下的联合优化，以确保训练过程中用户和物品的表示学习能够最大程度地服务于最终的推荐效果。
2. JNSKR 模型的输入为用户和物品的交互行为、知识图谱，我们首先会将它们

通过表示学习方法转化为使用稠密向量表示。物品作为连接推荐系统和知识图谱表示的桥梁来实现联合学习。最终的模型输出  $\hat{y}_{uv}$  是一个预测分数，表示用户  $u$  对物品  $v$  的偏好程度。

3. JNSKR 模型主要包含三个结构：A. 知识图谱表示学习模块，我们设计了高效非采样知识图谱学习算法；B. 推荐模块，计算用户对候选物品的兴趣情况，使用注意力机制来对物品的表示进行推荐和图谱信息的融合；C. 联合学习模块，以端到端的方式同时对上述两个模块进行训练优化，以得到最佳的推荐结果。

#### 5.4.2.2 高效非采样知识图谱表示学习

知识图谱的表示学习是将图谱上实体和关系转换成向量表示的重要途径，已被广泛应用于知识增强的推荐方法<sup>[29,93,143]</sup>。现有的知识图谱表示学习大都采用了负采样的方法进行模型训练<sup>[127,138,144]</sup>，但近期研究表明负采样学习并不稳定，使得学习过程很难收敛到最优的状态<sup>[43,51]</sup>。因此，在本工作中我们首先将前文所介绍的高效非采样学习算法拓展到三元组的知识图谱学习任务中，以获得更好的实体和关系表示。

若直接对知识图谱上的所有关系进行非采样学习，在使用平方损失函数的情况下，其损失函数将为如下形式：

$$\begin{aligned}\mathcal{L}_{KG}(\Theta) &= \sum_{h \in \mathbf{B}} \sum_{t \in \mathbf{E}} \sum_{r \in \mathbf{R}} w_{hrt} (g_{hrt} - \hat{g}_{hrt})^2 \\ &= \sum_{h \in \mathbf{B}} \sum_{t \in \mathbf{E}} \sum_{r \in \mathbf{R}} w_{hrt} (g_{hrt}^2 - 2g_{hrt}\hat{g}_{hrt} + \hat{g}_{hrt}^2)\end{aligned}\quad (5.13)$$

其中  $w_{hrt}$  表示训练样例  $g_{hrt}$  的权重； $g_{hrt} = 1$  表示在实体  $h$  和  $t$  间存在关系  $r$ ， $g_{hrt} = 0$  则表示不存在。显然，直接按照该损失函数来进行非采样学习的计算复杂度会非常高。考虑到  $g_{hrt} \in \{0, 1\}$  且信息是确定的在训练过程中不会发生变化，可以对公式 (5.13) 展开后去除常数项：

$$\begin{aligned}\tilde{\mathcal{L}}_{KG}(\Theta) &= -2 \sum_{h \in \mathbf{B}} \sum_{t \in \mathbf{E}^+} \sum_{r \in \mathbf{R}^+} w_{hrt}^+ \hat{g}_{hrt} + \sum_{h \in \mathbf{B}} \sum_{t \in \mathbf{E}} \sum_{r \in \mathbf{R}} w_{hrt} \hat{g}_{hrt}^2 \\ &= \underbrace{\sum_{h \in \mathbf{B}} \sum_{t \in \mathbf{E}^+} \sum_{r \in \mathbf{R}^+} ((w_{hrt}^+ - w_{hrt}^-) \hat{g}_{hrt}^2 - 2w_{hrt}^+ \hat{g}_{hrt})}_{\mathcal{L}_{KG}^P(\Theta)} + \underbrace{\sum_{h \in \mathbf{B}} \sum_{t \in \mathbf{E}} \sum_{r \in \mathbf{R}} w_{hrt}^- \hat{g}_{hrt}^2}_{\mathcal{L}_{KG}^A(\Theta)}\end{aligned}\quad (5.14)$$

其中，消去了与梯度计算无关的常数项，并对公式进行了进一步拆项化简。 $\mathcal{L}_{KG}^P(\Theta)$  记录了对于所有正例交互需要计算的内容， $\mathcal{L}_{KG}^A(\Theta)$  记录了需要对所有数据进行的

计算。因为真实场景下知识图谱也会相对稀疏，因此该损失函数的主要计算瓶颈在于  $\mathcal{L}_{KG}^A(\Theta)$  部分。

为了对公式 (5.14) 中的  $\mathcal{L}_{KG}^A(\Theta)$  进行优化，首先要确定用于知识图谱表示学习的具体方法，也就是  $\hat{g}_{hrt}$  的计算方法。在已有表示学习方法中，基于度量学习的方法如 TransR<sup>[138]</sup> 会引入新的迁移矩阵参数，将进一步增加  $\hat{g}_{hrt}$  的计算复杂度。因此，我们最终选择了 DistMult<sup>[144]</sup> 方法来对图谱进行表示学习，这是另一类基于分解机制的有代表性的知识图谱表示方法，其计算方式如下：

$$\hat{g}_{hrt} = \mathbf{e}_h^T \cdot \text{diag}(\mathbf{r}) \cdot \mathbf{e}_t = \sum_i^d e_{h,i} r_i e_{t,i} \quad (5.15)$$

其中， $\text{diag}(\mathbf{r})$  是一个对角矩阵，对角线上的值与关系特征向量  $\mathbf{r}$  是一一对应的。可以看到，这样的计算方式除了实体和关系特征向量外并不会引入任何新的待学习参数。

接下来，将 DistMult 的计算方式代入到  $\hat{g}_{hrt}^2$ ，并对向量的内积进行了拆项，将原有  $\mathbf{e}_h$ ， $\mathbf{e}_t$  和  $\mathbf{r}$  点乘计算过程改写为：

$$\begin{aligned} \hat{g}_{hrt}^2 &= \sum_i^d e_{h,i} r_i e_{t,i} \sum_j^d e_{h,j} r_j e_{t,j} \\ &= \sum_i^d \sum_j^d (e_{h,i} e_{h,j}) (r_i r_j) (e_{t,i} e_{t,j}) \end{aligned} \quad (5.16)$$

将公式 (5.16) 的化简结果代回到  $\mathcal{L}_{KG}^A(\Theta)$  中，同时，如果给每个三元组的权重  $w_{hrt}$  设置为仅与头实体相关<sup>[8-9]</sup>，那么  $e_{h,i}$ ， $e_{t,i}$  和  $r_i$  的计算过程可以被完全拆分开来单独计算。在此基础上，对于  $\sum_{h \in \mathbf{B}} w_h^- e_{h,i} e_{h,j}$ ， $\sum_{t \in \mathbf{E}} e_{t,i} e_{t,j}$  和  $\sum_{r \in \mathbf{R}} r_i r_j$  的优化也将变得相互独立。这样就可以通过提前将这几项的值计算出来以加速模型的训练过程。最终的优化后非采样知识图谱表示学习损失函数如下：

$$\tilde{\mathcal{L}}_{KG}(\Theta) = \mathcal{L}_{KG}^P(\Theta) + \sum_{i=1}^d \sum_{j=1}^d \left( \left( \sum_{r \in \mathbf{R}} r_i r_j \right) \left( \sum_{h \in \mathbf{B}} w_h^- e_{h,i} e_{h,j} \right) \left( \sum_{t \in \mathbf{E}} e_{t,i} e_{t,j} \right) \right) \quad (5.17)$$

总结来看，对于  $\mathcal{L}_{KG}^A(\Theta)$  函数中运算单元的重新组合是提升算法效率的关键。改进前的计算复杂度为  $O(|\mathbf{B}||\mathbf{E}||\mathbf{R}|d)$ ，改进后的计算复杂度则为  $O((|\mathbf{B}|+|\mathbf{E}|+|\mathbf{R}|)d^2)$ 。由于实际模型中使用的向量长度一般为 64-128，而关系和实体数都可能上万，因此较原有的计算方式将会有很大的效率提升。

## 5.4.2.3 知识增强的推荐模块

上一小节中对于知识图谱的表示学习过程进行了优化，接下来对知识增强的推荐模块进行介绍。我们使用了基于神经网络形式的矩阵分解类推荐方法<sup>[20]</sup>作为基本的推荐框架，每个用户-物品对的推荐得分计算方式如下：

$$\hat{y}_{uv} = \mathbf{h}^T (\mathbf{p}_u \odot \mathbf{q}_v) \quad (5.18)$$

其中  $\mathbf{p}_u \in \mathbb{R}^d$  和  $\mathbf{q}_v \in \mathbb{R}^d$  表示用户  $u$  和物品  $v$  的隐语义嵌入向量， $\mathbf{p}_u$  是基于用户的编号随机初始化表示，在训练的过程中会逐渐根据用户的已有交互学习得到其不同的特征向量；而待推荐物品的特征表示  $\mathbf{q}_v$  则将融合推荐场景下物品的编号信息和知识图谱上它作为实体节点的特征表示； $\odot$  符号表示对于两个向量进行逐项相乘； $\mathbf{h}$  表示预测层，为向量空间的不同维度分配相应的权重。

在对物品的特征计算过程中，我们使用神经网络中的注意力机制来融合推荐场景下的物品特征和知识图谱上的物品特征。具体来说，对于每个物品  $v$ ，它最终的表示  $\mathbf{q}_v$  不仅由它在推荐系统中的编号信息决定，也受到它在知识图谱上的表示以及其邻居节点和关联关系的影响。在知识图谱上，关联关系的类型和实体的含义都将影响邻居实体的表示，例如：用户在电影推荐场景下可能会更关注影片类型（关系类型），而且通常不同用户对动作片、剧情片等不同类型的影片有不同程度的偏好。因此，我们选择了注意力机制来根据用户的特征向量动态地调整对于物品在知识图谱上的表示的融合权重，从而为物品学习到更好的特征表示。使用集合  $\mathcal{N}_v = \{(v, r, t) | g_{vrt} = 1\}$  记录以物品  $v$  为首节点的所有知识图谱三元组，计算  $v$  表示的具体方式如下：

$$\begin{aligned} \mathbf{q}_v &= \mathbf{e}_v + \mathbf{e}_{\mathcal{N}_v} \\ &= \mathbf{e}_v + \sum_{(v,r,t) \in \mathcal{N}_v} \alpha_{(r,t)} \mathbf{e}_t \end{aligned} \quad (5.19)$$

其中  $\alpha_{(r,t)}$  是注意力权重，代表了在信息传播过程中实体节点  $t$  对物品  $v$  在关系  $r$  下造成了多大的影响。 $\mathbf{e}_v$  是根据物品编号在推荐模型中得到的物品基本表示，而  $\mathbf{e}_{\mathcal{N}_v}$  则是根据  $v$  在知识图谱的结构信息确定。 $\alpha_{(r,t)}$  的计算过程如下：

$$\begin{aligned} \alpha_{(r,t)}^* &= \mathbf{h}_\alpha^T \sigma(\mathbf{W}_1 \mathbf{e}_t + \mathbf{W}_2 \mathbf{r} + \mathbf{b}) \\ \alpha_{(r,t)} &= \frac{\exp(\alpha_{(r,t)}^*)}{\sum_{(v,r',t') \in \mathcal{N}_v} \exp(\alpha_{(r',t')}^*)} \end{aligned} \quad (5.20)$$

其中  $\mathbf{W}_1 \in \mathbb{R}^{k \times d}$ ， $\mathbf{W}_2 \in \mathbb{R}^{k \times d}$ ， $\mathbf{b} \in \mathbb{R}^k$ ，和  $\mathbf{h}_\alpha \in \mathbb{R}^k$  是注意力网络中的主要参数； $k$  代表的是注意力网络向量维度； $\sigma$  是非线性的激活函数  $ReLU$ <sup>[122]</sup>。注意力的最终权重通过 softmax 函数的归一化得到。

**算法 5.2** JNSKR 训练流程

**Require:** 训练集  $\{Y, U, V, G, E, R\}$ ; 训练样例权重  $c, w$ ; 学习率  $\eta$ ; 向量长度  $d$

**Ensure:** 神经网络模型参数  $\Theta$

- 1: 随机初始化模型参数  $\Theta$
- 2: **while** 结束条件尚未满足 **do**
- 3:   选出一批训练物品及相应训练样例  $\{Y_B, G_B\}$
- 4:   计算知识图谱表示学习的损失函数  $\tilde{\mathcal{L}}_{KG}(\Theta)$  (公式 (5.17))
- 5:   计算推荐模块的损失函数  $\tilde{\mathcal{L}}_{CF}(\Theta)$  (公式 (5.21))
- 6:    $\mathcal{L}(\Theta) \leftarrow \tilde{\mathcal{L}}_{CF}(\Theta) + \mu \tilde{\mathcal{L}}_{KG}(\Theta)$
- 7:   更新模型参数
- 8: **end while**
- 9: **return**  $\Theta$

至此，我们完成了对于物品  $v$  的向量化建模。推荐模块最终将通过公式 (5.18) 来计算得分，注意到推荐模块计算过程满足定理3.1的需求，因此对于推荐任务的预测损失函数可采用如下形式：

$$\begin{aligned} \tilde{\mathcal{L}}_{CF}(\Theta) = & \sum_{u \in U^+} \sum_{v \in B} ((c_v^+ - c_v^-) \hat{y}_{uv}^2 - 2c_v^+ \hat{y}_{uv}) \\ & + \sum_{i=1}^d \sum_{j=1}^d \left( (h_i h_j) \left( \sum_{u \in U} p_{u,i} p_{u,j} \right) \left( \sum_{v \in B} c_v^- q_{v,i} q_{v,j} \right) \right) \end{aligned} \quad (5.21)$$

#### 5.4.2.4 联合学习

为了能够更有效地将知识图谱上的物品特征融入到知识增强推荐方法中，我们提出使用联合学习的训练方式同时优化知识图谱表示和推荐结果，而不是采用现有工作广泛选择的两步式训练。具体优化目标函数如下：

$$\mathcal{L}(\Theta) = \tilde{\mathcal{L}}_{CF}(\Theta) + \mu \tilde{\mathcal{L}}_{KG}(\Theta) + \lambda \|\Theta\|_2^2 \quad (5.22)$$

其中，推荐模块的损失函数  $\tilde{\mathcal{L}}_{CF}(\Theta)$  计算方式见公式 (5.21)，知识图谱表示学习损失函数  $\tilde{\mathcal{L}}_{KG}(\Theta)$  计算方式见公式 (5.17)。为了能够控制两个任务对于最终训练的贡献，超参数  $\mu$  被用来控制比例情况。 $L_2$  正则化项被用于限制  $\Theta$ ，避免出现过拟合现象。

传统的知识增强推荐方法<sup>[29,95-96,139,145]</sup>大都在模型训练中使用了负采样的策略，且负采样同时存在于知识图谱表示学习和推荐模型的训练过程，引入了较大的模型训练随机性。JNSKR 模型利用前面介绍的针对两个场景下的高效非采样损失函数计算形式实现了高效的非采样联合学习。具体的训练流程参见算法流程5.2。

此外，为了更好地训练模型，我们采用 Adagrad<sup>[105]</sup>作为模型的优化器，它的主要优势在于会自适应地调整训练过程中的学习率，选择合理的步长来对解空间进行搜索，从而避免了对于学习率这个超参数过多的调参过程。去除函数 (Dropout)

表 5.7 评估 JNSKR 模型的数据集统计信息

		亚马逊图书	Yelp2018
用户-物品交互	用户数量	70,679	45,919
	物品数量	24,915	45,538
	交互数量	847,733	1,185,068
知识图谱	实体数量	88,572	90,961
	关系数量	39	42
	三元组数量	2,557,746	1,853,704

也被加入到模型中用来避免训练出现过拟合现象<sup>[123]</sup>，我们的随机去除主要是针对作为桥梁的物品向量  $\mathbf{q}_v$  进行的，去除操作之后，保留比例为  $\rho$  的向量参数。

#### 5.4.2.5 讨论

对整个 JNSKR 模型的计算时间复杂度进行分析。由于 JNSKR 的训练过程中同时对知识图谱表示和推荐模块进行优化，因此复杂度也将分为两部分。首先对于知识图谱表示部分（公式 (5.17)），每一批训练数据的计算复杂度为  $O((|\mathbf{B}| + |\mathbf{E}| + |\mathbf{R}|)d^2 + |\mathcal{G}_{\mathbf{B}}|d)$ ，其中  $\mathcal{G}_{\mathbf{B}}$  表示每批训练数据中实际存在的三元组关系。对于推荐模块（公式 (5.21)），每一批训练数据的计算复杂度为  $O((|\mathbf{B}| + |\mathbf{U}|)d^2 + |\mathcal{Y}_{\mathbf{B}}|d)$ （注意力机制的计算复杂度相对要小很多，因此被忽略了），其中  $\mathcal{Y}_{\mathbf{B}}$  表示每批训练数据中实际存在的用户-物品正向交互。两部分相加即得整体的计算复杂度为  $O((|\mathbf{B}| + |\mathbf{E}| + |\mathbf{R}|)d^2 + |\mathcal{G}_{\mathbf{B}}|d + (|\mathbf{B}| + |\mathbf{U}|)d^2 + |\mathcal{Y}_{\mathbf{B}}|d)$ 。如果不使用高效非采样这两个任务下传统非采样方法的复杂度为  $O((|\mathbf{B}||\mathbf{E}||\mathbf{R}| + |\mathbf{B}||\mathbf{U}|)d)$ 。考虑到实际场景中  $|\mathcal{G}_{\mathbf{B}}| \ll |\mathbf{B}||\mathbf{E}||\mathbf{R}|$ ， $|\mathcal{Y}_{\mathbf{B}}| \ll |\mathbf{B}||\mathbf{U}|$ ，且  $d \ll |\mathbf{B}|$ ，优化后的 JNSKR 较传统的非采样训练方法有了数量级的效率提升。

### 5.4.3 实验

本实验所用代码已开源<sup>①</sup>。

#### 5.4.3.1 数据集介绍

我们选择了两个被广泛使用的含知识图谱的推荐模型用于实验：亚马逊数据集中的图书推荐场景**亚马逊图书**<sup>②</sup>和 Yelp 平台发布的数据集 **Yelp2018**<sup>③</sup>。这两个

① <https://github.com/chenchongthu/JNSKR>

② <http://jmcauley.ucsd.edu/data/amazon>

③ <https://www.yelp.com/dataset/challenge>

表 5.8 JNSKR 与对比方法的特点比较

模型特点	NCF	ENMF	NFM	CKE	CFKG	RippleNet	KGAT	JNSKR
用户—物品交互	√	√	√	√	√	√	√	√
知识图谱	\	\	√	√	√	√	√	√
联合学习	\	\	\	\	\	\	\	√
非采样学习	\	√	\	\	\	\	\	√

数据也在其他的知识增强推荐方法中也被使用过<sup>[29]</sup><sup>①</sup>，在这里我们简要介绍两个数据集的基本情况：

- **亚马逊图书**：亚马逊数据集是被各类推荐算法广为使用的集合。在本研究工作中的实验中，我们选择了亚马逊数据集场景下的图书推荐数据集，知识图谱是通过将图书物品与 Freebase 知识图谱上的实体进行精准匹配后得到的。
- **Yelp2018**：该数据集是 2018 年 Yelp 举办的推荐比赛中发布的，它包含用户对商户和餐厅的评分信息。对应的知识图谱信息是通过将商户和餐厅的类别、位置和属性信息构建成异质信息网络。

为了保证算法中知识图谱的质量，两个数据集中连边数量少于 10 的实体将会从数据集中被移除，保留的关联关系则需要出现至少 50 次以上。对这些数据的预处理操作均与已有知识增强的推荐方法研究<sup>[29]</sup>保持一致。两个数据集的一些基本统计信息见表 5.7。

#### 5.4.3.2 基准方法

为了对比 JNSKR 算法的推荐表现和效率，我们主要选择了如下几类基准算法：协同过滤类（NCF 和 ENMF）、结合内容特征的（NFM）和多个知识增强的推荐方法（基于表示学习的 CKE 和 CFKG 方法、基于路径的 RippleNet 方法和基于图神经网络的 KGAT 方法），下面简要介绍这些基准算法：

- **NCF<sup>[20]</sup>**：神经网络协同过滤模型，是一个有代表性的基于神经网络的推荐模型，通过将矩阵分解与多层感知机相结合对隐式反馈数据进行建模。
- **ENMF<sup>[49]</sup>**：本文在第三章所提出的高效神经网络矩阵模型，通过高效非采样学习从整体数据上优化模型参数。
- **NFM<sup>[28]</sup>**：神经网络因子分解机是当前最优的融合用户和物品特征的推荐方法之一，使用多层神经网络来建模不同特征的高阶交互来提升算法表现。
- **CKE<sup>[137]</sup>**：有代表性的融合知识图谱上物品表征的知识增强推荐方法，它主

<sup>①</sup> [https://github.com/xiangwang1223/knowledge\\_graph\\_attention\\_network](https://github.com/xiangwang1223/knowledge_graph_attention_network)

要将物品在图谱上的语义信息通过 TransR<sup>[138]</sup> 方法融入到矩阵分解算法中。

- **CFKG**<sup>[143]</sup>: 知识增强的推荐方法, 主要使用了 TransE<sup>[127]</sup> 来学习融合知识图谱的异质信息网络上的用户、物品和实体表征, 然后将推荐任务转化为在异质信息网络上预测用户和物品间是否存在购买的连边;
- **RippleNet**<sup>[93]</sup>: 基于路径的知识增强的推荐方法, 使用循环神经网络建模知识图谱上物品之间的联系路径, 从而对用户进行更精准的表征, 也是当前最优的方法之一。
- **KGAT**<sup>[29]</sup>: 当前效果最好的知识增强的推荐方法之一, 主要使用了图神经网络和注意力机制来建模知识图谱上实体之间的高阶联系, 并据此优化推荐方法。

#### 5.4.3.3 评价方式

对每个数据集, 我们随机选择 80% 的交互历史数据来构建训练集, 10% 作为验证集用于调整超参数, 最后 10% 用来做测试集。在模型评价过程中, 我们将使用不同算法为每个用户对所有的候选物品进行排序 (除了训练集和验证集已知的有正向交互的物品)。为了评价前  $K$  个推荐结果的表现, Recall@ $K$  和 NDCG@ $K$  将作为主要的评价指标, 其中 Recall@ $K$  关注于用户交互过的正例是否被排在前  $K$  项中, 而 NDCG@ $K$  则会细粒度的考虑前  $K$  项中正例交互的具体位置, 这两个指标都是越高越好。

#### 5.4.3.4 超参数设置

所有基准算法的参数设定都是根据对应论文的初始化要求设置的, 且也都在验证集上进行了调参以确保其达到最优的效果。我们对超参数的主要搜索空间包括: 学习率的搜索范围为  $[0.005, 0.01, 0.02, 0.05]$ ,  $L_2$  正则化权重的搜索范围为  $[10^{-5}, 10^{-4}, \dots, 10^{-1}, 1]$ 。为了避免过拟合, 网络留存率 (Dropout) 的搜索范围为  $[0.0, 0.1, \dots, 0.9]$ , 特征向量长度  $d$  和  $k$  的搜索范围为  $[16, 32, 64]$ 。最终, 根据预实验结果, 每个训练批的大小被设置成 512, 学习率被设置成 0.05, 特征向量的长度被设置成 64, 而注意力机制的向量长度则为 32。对 NFM 方法, 神经网络的层数被设置成 1 层且具有 64 个神经元 (与原始论文一致<sup>[28]</sup>); 对于 RippleNet 方法, 我们将路径的长度和存储的数量分别设置为 2 和 8<sup>[93]</sup>。对 KGAT 方法, 根据原文的建议<sup>[29]</sup>, 我们将图神经网络层数设置成 3, 且对应的向量长度分别为 64, 32 和 16。ENMF 和我们提出的 JNSKR 使用了高效非采样的训练策略, 对于不同样例的权重设置我们参照了已有工作<sup>[9,51]</sup>, 将其设为与物品频率相关的值。此外, 在 JNSKR 的联合训练过程中, 超参数  $\mu$  被调为 0.01。

表 5.9 JNSKR 与不同推荐方法的效果比较, \*\* 表示较最好的基准方法有显著提升, 显著性是  $p < 0.01$ 。“RI”表示 JNSKR 相较对应方法的相对提升

模型	亚马逊图书						
	Recall@10	Recall@20	Recall@40	NDCG@10	NDCG@20	NDCG@40	RI
NCF	0.0874	0.1319	0.1924	0.0724	0.0895	0.1111	+17.03%
ENMF	0.1002	0.1472	0.2085	0.0797	0.0998	0.1215	+5.49%
NFM	0.0891	0.1366	0.1975	0.0723	0.0913	0.1152	+14.44%
CKE	0.0875	0.1343	0.1946	0.0705	0.0885	0.1114	+17.14%
CFKG	0.0769	0.1142	0.1901	0.0603	0.077	0.0985	+32.62%
RippleNet	0.0883	0.1336	0.2008	0.0747	0.0910	0.1164	+13.99%
KGAT	<u>0.1017</u>	<u>0.1489</u>	<u>0.2094</u>	<u>0.0814</u>	<u>0.1006</u>	<u>0.1225</u>	+4.31%
<b>JNSKR</b>	<b>0.1056**</b>	<b>0.1558**</b>	<b>0.2178**</b>	<b>0.0842**</b>	<b>0.1068**</b>	<b>0.1271**</b>	–
模型	Yelp2018						
	Recall@10	Recall@20	Recall@40	NDCG@10	NDCG@20	NDCG@40	RI
NCF	0.0389	0.0653	0.1060	0.0603	0.0802	0.1087	+14.28%
ENMF	0.0403	0.0711	0.1109	0.0611	<u>0.0877</u>	0.1097	+9.15%
NFM	0.0396	0.0660	0.1082	0.0603	0.0810	0.1094	+13.03%
CKE	0.0399	0.0657	0.1074	0.0608	0.0805	0.1091	+13.13%
CFKG	0.0288	0.0522	0.0904	0.0450	0.0644	0.0897	+44.27%
RippleNet	0.0402	0.0664	0.1088	0.0613	0.0822	0.1097	+11.90%
KGAT	<u>0.0418</u>	<u>0.0712</u>	<u>0.1128</u>	<u>0.0630</u>	0.0867	<u>0.1129</u>	+7.26%
<b>JNSKR</b>	<b>0.0456**</b>	<b>0.0749**</b>	<b>0.1209**</b>	<b>0.0687**</b>	<b>0.0917**</b>	<b>0.1211**</b>	–

#### 5.4.3.5 整体表现分析

JNSKR 方法与其他基准方法在两个数据集上的推荐表现见表 4.3, 为了对比不同长度推荐列表的效果, 我们汇报了推荐列表长度为 10、20 和 40 的结果。由于使用的数据集和划分方式与 KGAT 完全一致, 这里汇报 KGAT 的表现也与其原文一致<sup>[29]</sup>。从实验结果中我们有如下发现:

首先, 我们提出的 JNSKR 方法在两个数据集上都取得了最好的表现, 且显著优于现有的所有基准方法 ( $p$  值小于 0.01)。即便是与当前使用了最领先且需要耗费大量计算资源的 KGAT 方法, JNSKR 在亚马逊图书和 Yelp2018 数据集上也分别取得了平均 4.31% 和 7.26% 的相对提升。作为基于浅层神经网络的 JNSKR 的参数量较 KGAT 方法显著更少, 但是因为非采样训练方法的应用能够让 JNSKR 在整个数据集上找到最优解。其他推荐方法, 如: NCF, NFM, CKE, CFKG, RippleNet

和 KGAT 都是使用了负采样的训练方法，因此表现欠佳。此外，与仅基于用户-物品交互的协同过滤方法 NCF 和 ENMF 相比，JNSKR 的提升说明了知识信息的引入对于提升推荐算法的表现起到了关键性作用。

其次，使用了非采样学习的 ENMF 和 JNSKR 方法较其他基于负例采样的方法效果在大部分场景下都要更好，例如，在表中 ENMF 的结果较 NCF 有较大提升，而 JNSKR 击败了所有的对比算法。这一结果与已有研究工作中对非采样学习和负采样训练方法的比较结论是一致的<sup>[9,43,51,67]</sup>，即：非采样学习方法能够取得更优秀的模型表现。

最后，近期大量工作开始致力于研究知识增强推荐方法，也有很多先进的神经网络结构被应用于不同知识增强方法中，他们虽然较不使用知识信息的推荐方法相比有较大提升，但是他们取得的推荐表现其实相比没有使用任何交互外特征，但是使用了非采样训练的 ENMF 方法并没有太大优势。ENMF 就能够击败融合了知识或其他内容特征的 NFM, CKE, CFKG 和 RippleNet 方法，仅比 KGAT 方法略差。这也再次说明了至少对于推荐算法来说，使用合适的训练策略可能比引入更多的额外信息和设计更复杂的网络结构更加有用。

#### 5.4.3.6 训练效率分析

很多深度学习方法仅关注模型是否能够提供更好的结果，但是忽略了模型要实际应用所面临的效率挑战，因而也很少汇报模型的训练时间<sup>[11]</sup>。实际上如果模型训练或者输出的时间花费过大，很可能会限制它在真实场景下的应用。本节中，我们将提出的 JNSKR 算法与其他知识增强的推荐算法进行了效率比较，包括 CKE、CFKG、RippleNet 和 KGAT。所有的实验都是在同一台服务器上运行的，其配置为：英特尔 Xeon9 核心 CPU (2.4GHz)，使用了一张 NVIDIA GeForce GTX TITAN X GPU。实验得到的不同方法的训练效率见表5.10。

从表格中可以看到，本文提出的 JNSKR 较其他的知识增强推荐算法有数量级上的训练效率提升。以亚马逊图书数据集为例，JNSKR 仅需要 47 分钟就实现模型收敛达到最优，然而现有的表现最好的知识增强推荐方法 RippleNet 和 KGAT 方法分别需要 50 小时和 45 小时，加速了 20 倍左右，让知识增强的推荐学习方法更有可能被应用于真实的推荐场景中。再加上前文的实验验证了 JNSKR 方法在效果上也能显著优于现有的各类知识增强的推荐方法，表明了算法具有很强的可用性，能够被用于真实系统。

表 5.10 JNSKR 与不同推荐模型的训练耗时情况, 其中 s/m/h 分别代表秒/分钟/小时

模型	亚马逊图书			Yelp2018		
	每轮时长	训练轮数	总训练时长	每轮时长	训练轮数	总训练时长
CKE	66s	200	220m	75s	200	250m
CFKG	27s	200	90m	45s	200	150m
RippleNet	15m	200	50h	11m	200	37h
KGAT	9m	300	45h	7m	300	35h
<b>JNSKR</b>	<b>14s</b>	<b>200</b>	<b>47m</b>	<b>16s</b>	<b>200</b>	<b>54m</b>

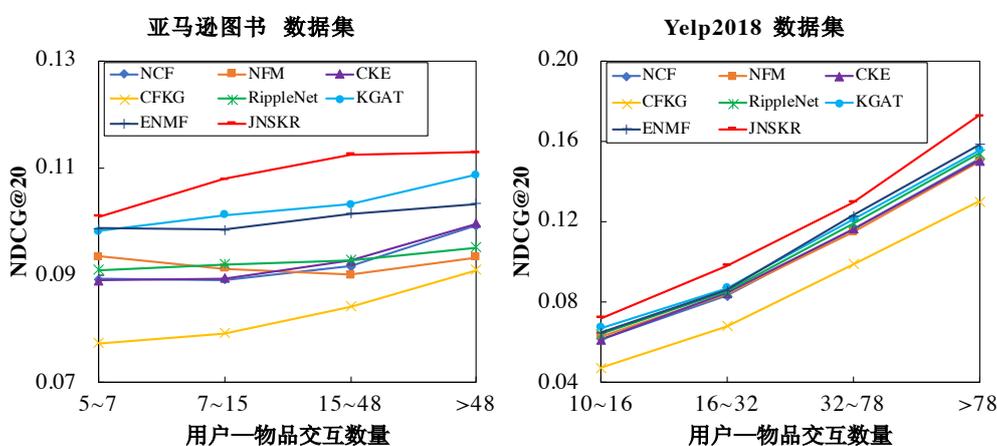


图 5.9 JNSKR 与不同推荐模型在不同交互行为数量下的表现情况

#### 5.4.3.7 稀疏数据表现分析

数据稀疏是真实推荐系统面临的一个主要挑战<sup>[146]</sup>, 其难点主要在于如何对仅有少量交互的用户或物品进行准确的特征建模。而知识图谱增强的推荐方法因为引入了外部信息, 能够将原本孤立的物品节点间建立关联, 因此能够在用户仅有少量交互的前提下对用户进行较为有效的兴趣建模。为了验证 JNSKR 模型在对仅有少量交互历史的用户推荐时的表现, 我们将两个数据集上的用户分成了人数相等的四个子群体, 对亚马逊图书数据集来说, 四个用户子集的交互次数分别为: 5-7 次、7-15 次、15-48 次和多于 48 次; 对 Yelp2018 来说, 四个用户子集的交互次数分别为 10-16 次、16-32 次、32-78 次和多于 78 次。图 5.9 中展示了以 NDCG@20 为评价指标时各个推荐方法在不同用户子集上的推荐表现。据此, 我们有如下的观察:

首先, 大部分知识增强的推荐方法较仅使用用户-物品交互的方法来说表现会更好一些。这些提升不仅体现在交互数量较少的用户身上, 对于有着丰富交互历史

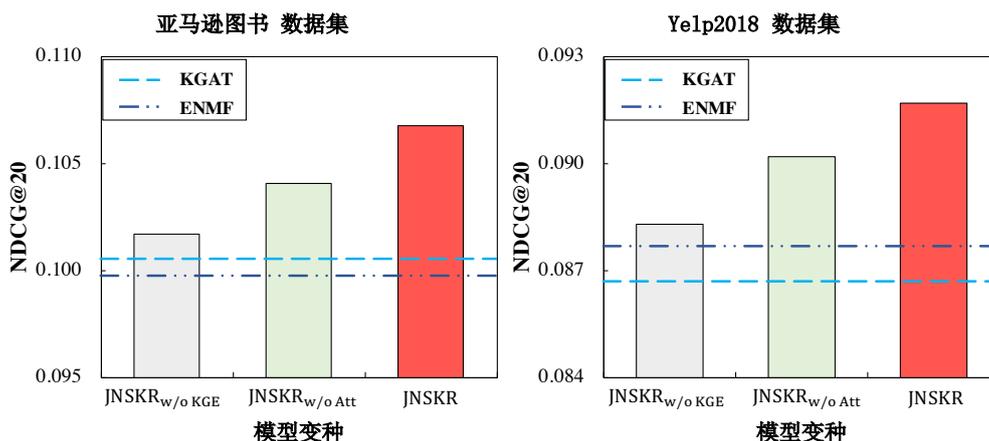


图 5.10 JNSKR 和它的两个变种的实验表现

的用户来说，知识图谱的引入也能帮助给他们带来更好的推荐结果。这很可能是因为在知识图谱上的物品间联系帮助更精准地建模了用户对于不同物品的兴趣程度。但是值得注意的是在 Yelp2018 数据集上，即便是结合知识图谱的 RippleNet 和 KGAT 方法也仅比最简单的 CF 方法要略好，可能的原因是在这个数据集上使用的知识图谱关系强度有限，中间也有一定的噪声，且用户的交互数量已经相对较多了，导致并不能带来太高的推荐结果提升<sup>[29]</sup>。其次，可以看到我们提出的 JNSKR 方法在各个场景下均取得了较现有知识增强推荐方法 RippleNet 和 KGAT 更优的表现，表明 JNSKR 能为不同交互数量的用户都带来一致的推荐效果提升，体现了非采样学习方法的稳定性。

#### 5.4.3.8 消融实验分析

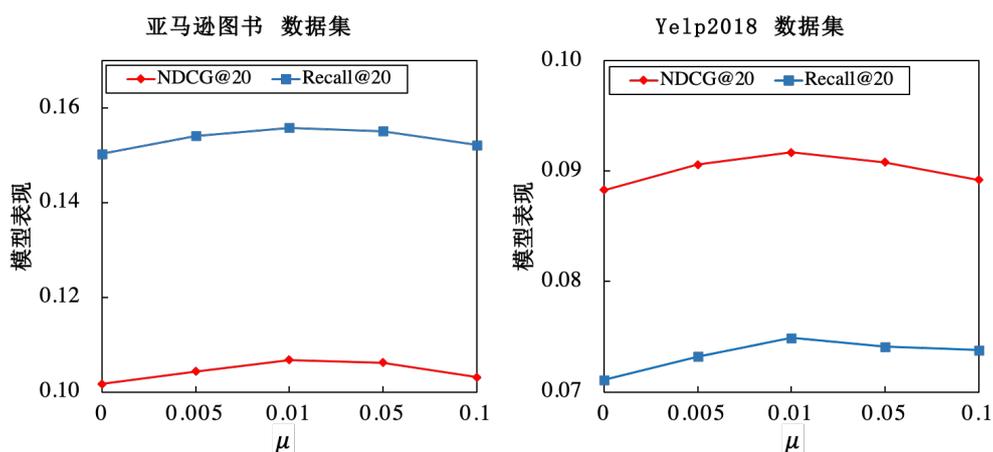
本节设计的 JNSKR 方法使用了非采样的学习策略来训练知识图谱表示，同时使用了注意力机制来融合物品在推荐系统上的表征和知识图谱上的特征，本小中我们首先验证这两个模块的有效性，因此设计了如下的模型变种：

- **JNSKR<sub>w/o KGE</sub>**: 该 JNSKR 变种去除了知识图谱表示学习部分（即在训练的损失函数中移除了公式 (5.17) 部分）。
- **JNSKR<sub>w/o Att</sub>**: 该 JNSKR 变种去掉了物品特征表示的注意力机制，改为计算知识图谱上的三元组对物品表征影响时使用固定的权重（见公式 (5.19)）。

图5.10中展示了两个变种在实验数据集上的推荐结果，为了直观地进行比较，我们在图中也标出了 ENMF（表现最好的仅使用用户-物品交互的推荐方法）和 KGAT（表现最好的结合知识图谱信息的推荐方法）的效果。有关实验结果的主要分析和结论如下：一方面，如果不使用知识图谱信息，JNSKR<sub>w/o KGE</sub> 的表现显著的弱于完整的 JNSKR 方法 ( $p < 0.01$ )。如果使用了知识图谱信息，即便是使用了

表 5.11 联合学习和交替学习两种训练策略在不同数据集上的结果比较, \* 表示有统计显著的结果差异  $p < 0.05$ 

	亚马逊图书		Yelp2018	
	Recall@20	NDCG@20	Recall@20	NDCG@20
JNSKR <sub>Alt</sub>	0.1532	0.1041	0.0731	0.0896
JNSKR	<b>0.1558*</b>	<b>0.1068*</b>	<b>0.0749*</b>	<b>0.0917*</b>

图 5.11 联合学习超参数  $\mu$  敏感性分析

最简单的通过固定的权重进行特征融合的  $JNSKR_{w/o Att}$  方法也能较不使用图谱的  $JNSKR_{w/o KGE}$  方法取得显著提升 ( $p < 0.05$ )。这表明两个模块对于 JNSKR 都有显著的正向作用。另一方面, 即便是不使用知识图谱信息的最弱的 JNSKR 的变种  $JNSKR_{w/o KGE}$ , 它的推荐结果也要比最好的基准算法 KGAT 要更好, 这说明我们设计的非采样学习方法能够显著提升对于用户的建模效果。

现有的不少推荐方法<sup>[29,96]</sup>在训练过程中都选择了对知识图谱表示  $\mathcal{L}_{KG}$  和推荐模块  $\mathcal{L}_{CF}$  交替优化训练。以 KGAT 为例, 它在训练过程中首先会固定推荐模块  $\mathcal{L}_{CF}$  部分的所有参数, 仅使用知识图谱表示的损失函数  $\mathcal{L}_{KG}$  来更新模型参数, 然后再将  $\mathcal{L}_{KG}$  部分的参数固定, 仅使用  $\mathcal{L}_{CF}$  来进行训练。我们认为这样的交替训练其实需要耗费更多的计算资源, 而且很可能在训练过程中导致模型更慢收敛, 因此在提出的 JNSKR 模型中选择了联合学习的方式。为了验证联合训练的有效性, 我们开展了如下的对比实验来比较使用联合训练和交替训练策略时 JNSKR 方法的表现, 实验结果如表 5.11, 其中,  $JNSKR_{Alt}$  就是使用交替学习策略的 JNSKR 变种。

从表中我们可以看到, 使用交替学习策略的  $JNSKR_{Alt}$  方法较使用联合学习方法的 JNSKR 有显著差异, 这说明交替学习的训练方式确实难以细粒度地建模推荐场景下用户对物品的兴趣程度, 因此取得的结果也要欠佳。

此外，我们也对联合学习策略下的超参数  $\mu$  开展了参数敏感性分析（见公式(5.22)）。它控制了两个子任务：图谱表示学习和推荐模块对于整个模型更新的贡献度。由于推荐为主任务，因此我们对  $\mu$  分别取值为 [0, 0.005, 0.01, 0.05, 0.1] 开展了对比实验，结果如图5.11所示。从图中我们可以看到，随着  $\mu$  的取值升高，推荐结果的表现呈现了先上升后下降的趋势，这说明确保主任务在损失函数在主导地位非常重要。同时，在验证集上找到一个最优的联合学习策略也非常重要，如果超参数选择不当，甚至可能出现比不使用联合学习还要更差的情况。

## 5.5 小结

本章继续面向高效神经网络推荐系统中作为核心的“推荐模型”问题，重点研究了结合内容信息的高效推荐建模。具体地，本章研究的内容信息包括用户和物品的特征属性以及知识图谱信息。在结合特征信息方面，本章提出了基于非采样因子分解机的高效特征增强推荐模型 ENSFM，不仅从理论上在因子分解机和矩阵分解之间建立了转换关系，且利用高效非采样学习来全面考虑用户偏好与特征信息之间的关联，提升推荐效果；在结合知识图谱数据方面，本章提出了基于非采样联合学习的高效知识增强推荐模型 JNSKR，首先以数学推导的方法实现了对于三元组表示学习的高效非采样优化，然后在推荐模块中使用注意力机制来融合物品在推荐场景的基本表示和在知识图谱上与其他实体间的关联关系情况，最后通过联合学习的方式同时对知识图谱表示和推荐模型开展训练。在多个数据集上的丰富实验和分析表明，我们所提出的 ENSFM 和 JNSKR 模型较已有方法在推荐效果和训练效率上均取得了显著的提升。实验结果同时也表明，在学习策略上的改进甚至能比使用更先进的神经网络结构带来更大的提升。

本章相关成果发表于 CCF A 类会议论文 TheWebConf 2020<sup>[54]</sup>和 SIGIR 2020<sup>[55]</sup>。

## 第6章 高效可遗忘推荐系统框架研究

### 6.1 引言

本文前面部分从高效神经网络推荐中的学习方法及模型设计的角度进行了研究。在本章中，我们将开始从“系统平台”层面进行研究。推荐系统通过从训练数据中学习来为用户提供个性化的产品和服务。然而，考虑到所收集的交互数据可能包含着敏感信息或是噪声点，推荐系统在一些场景也需要快速遗忘某些数据以及从这些数据中学到的模型参数<sup>[16,44,147]</sup>。首先，最近一些关于数据隐私保护的法案颁布，如海外的 GDPR (General Data Protection Regulation)<sup>①</sup>和 CCPA (California Consumer Privacy Act)<sup>②</sup>赋予数据“被遗忘的权利”，“被遗忘的权利”要求系统能够消除数据所有者要求遗忘的数据及其带来的影响（如从数据中学习到的用户偏好）。国内的《互联网信息服务算法推荐管理规定》<sup>③</sup>也对数据遗忘做了不同程度的规定。其次，在模型稳定性上，如果模型的表现效果被一些噪声数据<sup>[148-149]</sup>所降低时，需要遗忘这些脏数据对于模型参数的影响。此外，用户的偏好通常是动态变化的<sup>[150]</sup>。例如，想要购买手机的用户会乐于看到有关手机的推荐。但在购买之后，她/他将在一段时间内对新手机的推荐不感兴趣。在这种情况下，用户将希望删除手机相关的交互数据，以便系统能够提供更有用的推荐。

最直接的数据遗忘方法是用去除要删除数据后的训练集重新训练模型。然而在大规模现实数据场景下，重训模型需要巨大的计算开销。因此，高效的、可遗忘推荐系统有着重要的实际需求和研究价值。机器遗忘 (Machine Unlearning) 作为人工智能和机器学习的新生领域，现有的方法仍处于刚起步阶段。目前关于数据遗忘的研究主要集中在计算机视觉和自然语言处理的相关任务上<sup>[44-46,151-152]</sup>。例如，SISA 方法<sup>[16]</sup>将训练数据随机分割成几个不相交的子集，然后在每个子集上训练子模型。最终的预测结果通过多数投票 (Majority Voting) 或平均的方式将子模型聚合得到分数。当收到特定数据遗忘的请求时，只需要将相应的子模型重新训练即可。然而，通过随机的方式划分为子集可能会严重损害结构化的信息。同时，现有机器遗忘方法中的聚合部分通常为每个子模型分配一个静态权重，无法在预测不同的用户—物品交互时动态改变。因此，直接将已有的机器遗忘方法应用于推荐任务上时会造成较大的精度损失。

针对上述问题，本章提出了一个通用的高效可遗忘推荐系统框架 RecEraser，可

① <https://gdpr-info.eu/>

② <https://oag.ca.gov/privacy/ccpa>

③ [http://www.cac.gov.cn/2022-01/04/c\\_1642894606364259.htm](http://www.cac.gov.cn/2022-01/04/c_1642894606364259.htm)

表 6.1 高效可遗忘推荐系统研究中所用数学符号

符号	描述
$U, V$	分别代表用户集合和物品集合
$Y$	用户—物品交互
$K$	划分子数据集的数量
$S_i$	第 $i$ 个子数据集
$M_i$	在子数据集 $S_i$ 上训练的子模型
$\bar{p}_u, \bar{q}_v$	预训练的用户 $u$ 和物品 $v$ 的向量表示, 用于数据划分
$p_u^i, q_v^i$	由子模型 $M_i$ 训练得到的用户 $u$ 和物品 $v$ 的向量表示
$p_u, q_v$	聚合后的用户 $u$ 和物品 $v$ 的向量表示

以在保持高推荐性能的同时实现高效的数据遗忘。RecEraser 同样是将训练集划分为多个子集, 并训练子模型, 随后进行模型聚合得到最终的预测结果。与现有工作不同的是, 为了保持数据的结构信息, 我们新设计了三种分别基于用户、物品和交互行为相似性的数据划分策略。同时, 考虑到推荐系统的预测要面向独立的用户和物品, 不同子模型对不同的用户—物品的预测过程应是动态变化的。为了进一步提高推荐性能, 我们也提出了一种基于注意力机制的自适应模型聚合方法。值得一提的是, RecEraser 作为一个通用的上层框架, 可以和各种推荐系统模型配合使用。为了评估 RecEraser 的效果, 我们在三个现实数据集上基于三个有代表性的推荐系统模型 BPR<sup>[6]</sup>, WMF<sup>[7,52]</sup>, 和 LightGCN<sup>[42]</sup>进行了实验。实验结果表明, RecEraser 不仅可以实现高效的数据遗忘, 而且在性能上优于当前最先进的机器遗忘框架, 如 SISA<sup>[16]</sup>和 GraphEraser<sup>[53]</sup>。进一步的实验也验证了所提出的数据划分策略和自适应聚合方法的有效性。

本章的结构安排如下, 第6.2节介绍预备知识; 第6.3节介绍所提出的高效可遗忘推荐框架, 包括数据平衡分割算法以及基于注意力机制的模型聚合; 第6.4节展示了实验结果; 最后, 第6.5节对本章进行小结。

## 6.2 预备知识

我们首先介绍符号表示, 然后定义推荐系统中的数据遗忘任务, 并讨论其与现有机器遗忘任务的差异和挑战。

### 6.2.1 问题定义

表6.1描述了本章中使用的数学符号和相应描述。假设数据集中有  $M$  个用户和  $N$  个物品，用户集合记为  $\mathbf{U}$ ，物品集合记为  $\mathbf{V}$ 。我们使用索引  $u$  表示用户，使用  $v$  表示物品。用户—物品交互矩阵记为  $\mathbf{Y} = [y_{uv}]_{M \times N} \in \{0, 1\}$ ，表示  $u$  是否与物品  $v$  有过交互。推荐任务是给定目标用户  $u$ ，预测  $u$  可能感兴趣的物品列表。

对于推荐中的数据遗忘，如果用户  $u$  申请遗忘有关物品  $v$  的交互记录（即， $y_{uv}$ ），推荐系统需要完全去除  $y_{uv}$  对模型参数的影响，即等同于在训练集  $\mathbf{Y} \setminus y_{uv}$  上训练模型。具体地，推荐中的数据遗忘的任务要实现以下三个目标：

- **理论遗忘保证**：推荐中数据遗忘的基本要求是能够证明完全去除要遗忘数据对模型参数的影响。
- **高遗忘效率**：推荐中数据遗忘的过程应尽可能快。
- **保持推荐准确度**：推荐中应用的遗忘框架应尽可能保持原有的推荐准确度。

#### 6.2.1.1 推荐中数据遗忘的挑战

目前关于机器遗忘的研究主要集中在计算机视觉和自然语言处理的相关任务上<sup>[44-46]</sup>，直接应用于推荐任务上时会造成较大的精度损失。例如，目前最先进的机器遗忘方法 SISA<sup>[16]</sup>使用随机方法将训练数据划分为多个子集。然而，推荐中的训练数据是用户—物品交互记录，其中包含丰富的协同信息。随机分割训练数据会破坏数据的结构，从而影响推荐性能。一种容易想到的解决方案是使用社区发现或聚类方法<sup>[153-155]</sup>来分割数据。但是这些方法可能会导致高度不平衡的数据划分，从而影响数据遗忘的效率。因此，需要设计新的可以保持用户—物品结构化信息的平衡数据分割算法。此外，现有机器遗忘方法中的聚合部分通常为每个子模型分配一个静态权重，这不适用于在预测时要面向独立的用户和物品的推荐系统任务。虽然最近的 GraphEraser<sup>[53]</sup>使用基于学习的方法来分配权重，但权重无法在分配后随着不同的预测目标而动态改变。综上所述，如何在保证推荐精准度的同时实现通用且高效率的数据遗忘需要进一步的研究和探索。

## 6.3 高效可遗忘推荐框架

在本节中，我们首先概述 RecEraser 框架，然后详细介绍所提出的 RecEraser 中两个关键部分：(1) 平衡数据分割算法和 (2) 基于注意力机制的自适应聚合模块。

### 6.3.1 方法概览

RecEraser 的示意图如图6.1所示。首先对其做一简要介绍：

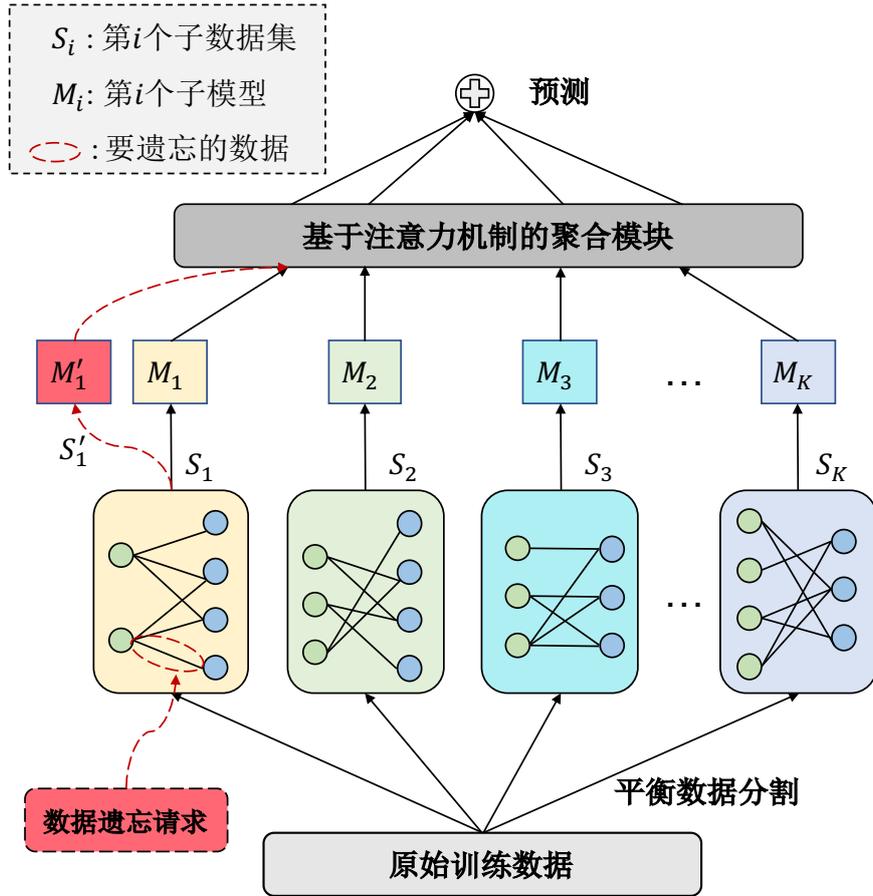


图 6.1 高效可遗忘推荐框架 (RecEraser) 示意图

1. RecEraser 框架由三个部分组成，包括平衡数据分割、子模型训练和基于注意力的自适应聚合模块。
2. 首先平衡数据分割部分用于在保留结构化信息的同时将训练数据划分为平衡的子集。然后为每个子数据集 ( $S_i$ ) 训练一个子模型 ( $M_i$ )。在预测阶段，采用基于注意力的自适应聚合策略动态地为不同子模型分配权重以得到最终的预测分数。
3. 当收到特定数据遗忘的请求时，只需要将相应的子模型和最后的聚合部分重新训练即可，相比于重训整个模型效率有了极大提升。

### 6.3.2 数据平衡分割算法

正如之前提到的，推荐中的训练数据是用户—物品交互记录，其中包含丰富的协同信息。为了保留这些结构化信息，我们提出了三种新的平衡数据分割算法，即基于用户的平衡分割 (UBP, User-based Balanced Partition)、基于物品的平衡分割 (IBP, Item-based Balanced Partition) 和基于交互的平衡分割 (InBP, Interaction-based Balanced Partition)。实现平衡分割的整体思路是设置最大的子数据集大小  $t$ ，并且

**算法 6.1** 基于用户的平衡分割算法 (UBP)

---

**Require:** 预训练用户向量  $\bar{\mathbf{P}} = \{\bar{\mathbf{p}}_1, \bar{\mathbf{p}}_2, \dots, \bar{\mathbf{p}}_M\}$ ; 用户—物品交互  $\mathbf{Y}$ ; 子数据集数量  $K$ ; 每个子数据集最大值  $t$

**Ensure:** 分隔好的子数据集  $\mathbf{S} = \{S_1, S_2, \dots, S_K\}$

- 1: 随机从  $\mathbf{U}$  中选择  $K$  个中心点  $\mathbf{A} = \{a_1, a_2, \dots, a_K\}$
- 2: **while** 结束条件尚未满足 **do**
- 3:   **for** 每个  $\mathbf{A}$  中的  $a_i$  **do**
- 4:     **for** 每个  $\mathbf{U}$  中的  $u$  **do**
- 5:       计算数据点距离  $E(a_i, u) = \text{dist}(a_i, u)$  (公式 (6.1))
- 6:     **end for**
- 7:   **end for**
- 8:   将距离  $E$  按照从大到小排序, 获得  $E_s$
- 9:   清空  $\mathbf{S}$
- 10:  **for** 每个  $E_s$  中的  $a_i$  和  $u$  **do**
- 11:    **if**  $|S_i| < t$  且用户  $u$  还未分配 **then**
- 12:      $S_i \leftarrow S_i \cup Y_u$
- 13:    **end if**
- 14:  **end for**
- 15:  通过公式 (6.2) 更新  $\mathbf{A}$
- 16: **end while**
- 17: **return**  $\mathbf{S}$

---

只在子数据集大小小于  $t$  时分配新的数据点。

UBP 算法根据用户的相似度划分训练数据, 如算法6.1所示。给定用户集  $\mathbf{U}$ , 首先为每个子数据集随机选择一个用户作为中心点, 标记为  $\mathbf{A} = \{a_1, a_2, \dots, a_K\}$ 。对于每个中心点用户, 通过计算相对于所有其他用户的距离来为子数据集分配数据。本文采用欧几里得距离<sup>[155]</sup>, 计算方式如下:

$$\text{dist}(a_i, u) = \|\bar{\mathbf{p}}_{a_i} - \bar{\mathbf{p}}_u\|_2 = \sqrt{\sum_{j=1}^n (\bar{p}_{a_i,j} - \bar{p}_{u,j})^2} \quad (6.1)$$

其中  $\bar{\mathbf{p}}_{a_i}$  和  $\bar{\mathbf{p}}_u$  分别是用户  $a_i$  和用户  $u$  的预训练向量, 在本工作中由模型 WMF<sup>[7]</sup> 学习得到。这里的用户向量也可以用其他推荐模型学习。

之后, 计算好的距离分数按照降序进行排序得到  $E_s$ 。对于  $E_s$  中的每一个用户—中心点对, 如果  $S_i$  的大小小于最大值  $t$ , 则将用户  $u$  的所有交互数据分配给该子数据集。最后, 新的中心点被计算为其相应子数据集中所有用户表示向量的平均值:

$$a_i = \frac{\sum_{j \in S_i} \bar{\mathbf{p}}_j}{|S_i|} \quad (6.2)$$

UBP 算法重复上述步骤, 直到满足停止标准 (达到最大迭代或中心点不变)。

IBP 方法基于预训练物品表示向量的相似性, 整体过程与 UBP 相似, 如算

**算法 6.2** 基于物品的平衡分割算法 (IBP)

**Require:** 预训练物品向量  $\bar{\mathbf{Q}} = \{\bar{\mathbf{q}}_1, \bar{\mathbf{q}}_2, \dots, \bar{\mathbf{q}}_N\}$ ; 用户—物品交互  $\mathbf{Y}$ ; 子数据集数量  $K$ ; 每个子数据集最大值  $t$

**Ensure:** 分隔好的子数据集  $\mathbf{S} = \{S_1, S_2, \dots, S_K\}$

```

1: 随机从  $\mathbf{V}$  中选择  $K$  个中心点  $\mathbf{A} = \{a_1, a_2, \dots, a_K\}$ 
2: while 结束条件尚未满足 do
3:   for 每个  $\mathbf{A}$  中的  $a_i$  do
4:     for 每个  $\mathbf{V}$  中的  $v$  do
5:       计算数据点距离  $E(a_i, v) = \text{dist}(a_i, v)$ 
6:     end for
7:   end for
8:   将距离  $E$  按照从大到小排序, 获得  $E_s$ 
9:   清空  $\mathbf{S}$ 
10:  for 每个  $E_s$  中的  $a_i$  和  $v$  do
11:    if  $|S_i| < t$  且物品  $v$  还未分配 then
12:       $S_i \leftarrow S_i \cup Y_v$ 
13:    end if
14:  end for
15:  更新  $\mathbf{A}$ 
16: end while
17: return  $\mathbf{S}$ 
    
```

法6.2所示。

UBP 和 IBP 分别基于用户和物品来分割数据, 这可能会忽略用户—物品交互的全局特性。因此, 我们进一步提出了一种基于交互的平衡分割算法 InBP, 如算法6.3所示。

InBP 与 UBP 和 IBP 的区别在于: (1) InBP 同时用到预训练的用户和物品向量; (2) InBP 从用户—物品交互中选择中心点  $\mathbf{A} = \{a_1, a_2, \dots, a_K\}$ , 其中  $a_i = (\mathbf{p}_i, \mathbf{q}_i)$ ; (3) 中心点  $a_i$  和交互  $y_{uv}$  之间的距离通过如下方式计算:

$$\begin{aligned} \text{dist}(a_i, y_{uv}) &= \|\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_u\|_2 \times \|\bar{\mathbf{q}}_i - \bar{\mathbf{q}}_v\|_2 \\ &= \sqrt{\sum_{j=1}^n (\bar{p}_{i,j} - \bar{p}_{u,j})^2} \times \sqrt{\sum_{j=1}^n (\bar{q}_{i,j} - \bar{q}_{v,j})^2} \end{aligned} \quad (6.3)$$

(4) 新的中心点计算方式如下:

$$a_i = \left( \frac{\sum_{j \in S_i} \bar{\mathbf{p}}_j}{|S_i|}, \frac{\sum_{j \in S_i} \bar{\mathbf{q}}_j}{|S_i|} \right) \quad (6.4)$$

### 6.3.3 基于注意力机制的模型聚合

在对训练数据进行划分之后, 在每个子数据集  $S_i$  上训练一个子模型  $M_i$ 。考虑到推荐系统的预测要面向独立的用户和物品, 不同子模型对不同的用户—物品

**算法 6.3** 基于交互的平衡分割算法 (InBP)

**Require:** 预训练用户向量  $\bar{\mathbf{P}} = \{\bar{\mathbf{p}}_1, \bar{\mathbf{p}}_2, \dots, \bar{\mathbf{p}}_M\}$ ; 预训练物品向量  $\bar{\mathbf{Q}} = \{\bar{\mathbf{q}}_1, \bar{\mathbf{q}}_2, \dots, \bar{\mathbf{q}}_N\}$ ; 用户—物品交互  $\mathbf{Y}$ ; 子数据集数量  $K$ ; 每个子数据集最大值  $t$

**Ensure:** 分隔好的子数据集  $\mathbf{S} = \{S_1, S_2, \dots, S_K\}$

- 1: 随机从  $\mathbf{Y}$  中选择  $K$  个中心点  $\mathbf{A} = \{a_1, a_2, \dots, a_K\}$
- 2: **while** 结束条件尚未满足 **do**
- 3:   **for** 每个  $\mathbf{A}$  中的  $a_i$  **do**
- 4:     **for** 每个  $\mathbf{Y}$  中的  $y_{uv}$  **do**
- 5:       计算数据点距离  $E(a_i, y_{uv}) = \text{dist}(a_i, y_{uv})$  (公式 (6.3))
- 6:     **end for**
- 7:   **end for**
- 8:   将距离  $E$  按照从大到小排序, 获得  $E_s$
- 9:   清空  $\mathbf{S}$
- 10:  **for** 每个  $E_s$  中的  $a_i$  和  $v$  **do**
- 11:    **if**  $|S_i| < t$  且交互数据  $y_{uv}$  还未分配 **then**
- 12:      $S_i \leftarrow S_i \cup y_{uv}$
- 13:    **end if**
- 14:  **end for**
- 15:  通过公式 (6.4) 更新  $\mathbf{A}$
- 16: **end while**
- 17: **return**  $\mathbf{S}$

的预测过程应是动态变化的。例如, 如果子数据集  $S_i$  中包含的用户  $u$  的训练交互多于子数据集  $S_j$ , 那么在预测  $u$  的偏好时,  $M_i$  的权重更有可能大于  $M_j$ 。为了进一步提高推荐性能, 我们提出了一种基于注意力机制的自适应聚合方法。

考虑到不同子模型学习到的用户和物品的表示可能不在同一向量空间, 我们首先将它们转换到相同的表示空间中。具体而言, 对于子模型  $M_i$  学习到的用户向量  $\mathbf{P}^i$  和物品向量  $\mathbf{Q}^i$ , 如下所示:

$$\mathbf{P}_{tr}^i = \mathbf{W}^i \mathbf{P}^i + \mathbf{b}^i; \quad \mathbf{Q}_{tr}^i = \mathbf{W}^i \mathbf{Q}^i + \mathbf{b}^i \quad (6.5)$$

其中  $\mathbf{W}^i \in \mathbb{R}^{d \times d}$  用于将  $\mathbf{P}^i$  和  $\mathbf{Q}^i$  转移到相同的向量空间;  $\mathbf{b}^i \in \mathbb{R}^d$  是向量偏差。

然后, 用户和物品的最终聚合向量表示计算如下:

$$\mathbf{P} = \sum_{i=1}^K \alpha_i \mathbf{P}_{tr}^i; \quad \mathbf{Q} = \sum_{i=1}^K \beta_i \mathbf{Q}_{tr}^i \quad (6.6)$$

其中  $\alpha_i$  和  $\beta_i$  分别代表用户和物品聚合时子模型  $M_i$  的注意力权重, 计算方式如下:

$$\begin{aligned} \alpha_i^* &= \mathbf{h}_1^\top \sigma(\mathbf{W}_1 \mathbf{p}_{tr}^i + \mathbf{b}_1); & \alpha_i &= \frac{\exp(\alpha_i^*)}{\sum_{j=1}^K \exp(\alpha_j^*)} \\ \beta_i^* &= \mathbf{h}_2^\top \sigma(\mathbf{W}_2 \mathbf{p}_{tr}^i + \mathbf{b}_2); & \beta_i &= \frac{\exp(\beta_i^*)}{\sum_{j=1}^K \exp(\beta_j^*)} \end{aligned} \quad (6.7)$$

其中  $\mathbf{W}_1 \in \mathbb{R}^{k \times d}$ ,  $\mathbf{b}_1 \in \mathbb{R}^k$ , 和  $\mathbf{h}_1 \in \mathbb{R}^k$  是计算用户注意力的参数;  $\mathbf{W}_2 \in \mathbb{R}^{k \times d}$ ,  $\mathbf{b}_2 \in \mathbb{R}^k$ , 和  $\mathbf{h}_2 \in \mathbb{R}^k$  是计算物品注意力的参数;  $k$  是注意力网络向量维度;  $\sigma$  代表非线性激活函数  $\text{ReLU}$  [122]。之后, 注意力的最终权重通过 softmax 函数归一化得到。

基于注意力的自适应聚合方法基于以下目标函数优化:

$$\min_{\Theta} \mathcal{L}(\mathbf{P}, \mathbf{Q}, \mathbf{Y}) + \lambda \|\Theta\|_2^2 \quad (6.8)$$

其中  $\mathbf{P}$  和  $\mathbf{Q}$  分别代表用户和物品聚合后的向量表示;  $\mathbf{Y}$  是用户—物品交互的真实标签;  $\mathcal{L}$  表示基模型的损失函数 (如 BPR [6]);  $\Theta$  表示注意模型的参数;  $L_2$  正则化防止模型过拟合。注意, 只有聚合模块的参数通过公式 (6.8) 进行优化, 而子模型学习的向量表示  $\mathbf{P}^i$  和  $\mathbf{Q}^i$  是固定的。

由于在聚合模块优化时使用了全部的训练数据  $\mathbf{Y}$ , 其中可能也会包含需要遗忘的数据。因此, 该聚合方法在遗忘数据后需要重新进行训练以彻底遗忘特定数据对于模型参数的影响。幸运的是, 在实际应用中该聚合方法只需要很少的轮数 (大约 10 轮) 即可训练完毕, 并且可以有效地提高推荐系统性能。

### 6.3.4 训练

RecEraser 框架是和模型无关的, 适用于多种推荐系统模型。在本工作中, 我们将其跟三个有代表性的推荐系统模型相结合, 分别是 BPR [6], WMF [7], 和 LightGCN [42]。这三个模型的评分函数均基于向量内积。形式上, 对于每个子模型  $M_i$ , 评分函数被定义为:

$$\hat{y}_{uv} = \mathbf{p}_u^i \top \mathbf{q}_v^i \quad (6.9)$$

其中  $\mathbf{p}_u^i$  和  $\mathbf{q}_v^i$  分别代表由子模型  $M_i$  训练得到的用户  $u$  和物品  $v$  的向量表示。

BPR [6] 和 LightGCN [42] 使用相同的负采样学习策略。具体地, 子模型  $M_i$  的目标函数是:

$$\mathcal{L}(\mathbf{P}^i, \mathbf{Q}^i, S_i) = - \sum_{(u,v) \in S_i} \ln \sigma(\hat{y}_{uv} - \hat{y}_{uj}) \quad (6.10)$$

其中  $\sigma(x) = 1/(1 + \exp(-x))$  是 sigmoid 函数, 对于每个正向用户—物品对  $(u, v)$ , 从用户  $u$  的未交互物品中采样负例  $(u, j)$ 。BPR 和 LightGCN 之间的区别在于, BPR 仅从用户物品编号学习表示向量  $(\mathbf{P}^i, \mathbf{Q}^i)$ , 而 LightGCN 则通过图神经网络从编号和邻居节点中学习向量表示。

表 6.2 评估 RecEraser 的数据集统计信息

数据集	用户数量	物品数量	用户—物品交互数量	稠密度
<b>Yelp2018</b>	31,668	38,048	1,561,406	0.13%
<b>Movielens-1m</b>	6,940	3,706	1,000,209	3.89%
<b>Movielens-10m</b>	71,567	10,681	10,000,054	1.31%

WMF<sup>[7]</sup>是一种基于非采样的推荐方法，子模型  $M_i$  目标函数为：

$$\mathcal{L}(\mathbf{P}^i, \mathbf{Q}^i, S_i) = \sum_{u \in \mathbf{U}_i} \sum_{v \in \mathbf{V}_i} c_{uv} (y_{uv} - \hat{y}_{uv})^2 \quad (6.11)$$

其中  $c_{uv}$  表示训练样例  $y_{uv}$  的权重； $\mathbf{U}_i$  和  $\mathbf{V}_i$  分别表示子模型  $M_i$  中的用户集合和物品集合。根据定理3.1，上述目标函数符合本文所提出的高效非采样学习的形式，因此本文中 WMF 方法同样采用高效非采样目标函数进行优化。

此外，为了更好地训练模型，我们采用 Adagrad<sup>[105]</sup>作为优化器，它的主要优势在于会自适应地调整训练过程中的学习率，选择合理的步长来对解空间进行搜索，从而避免了对于学习率这个超参数过多的调参过程。RecEraser 采用了两步训练的流程。第一步是子模型训练，由于子模型互相独立，因此可以并行训练不同的子模型以进一步加速。第二步是固定子模型参数，并通过公式 (6.8) 训练基于注意力机制的模型聚合部分。

## 6.4 实验

本实验所用代码已开源<sup>①</sup>。

### 6.4.1 实验设置

#### 6.4.1.1 数据集介绍

我们选择三个被广泛使用的推荐系统公开数据集进行实验，分别是 **Yelp2018**<sup>②</sup>，**Movielens-1m**<sup>③</sup> 和 **Movielens-10m**<sup>④</sup>。对三个数据集的简单介绍如下：

- **Yelp2018**：该数据集是 2018 年 Yelp 举办的推荐比赛中发布的，它包含用户对商户和餐厅的评分信息。由于本文重点关注隐式数据，因此具体的评分被转换为 0 或 1 的值，表示用户是否对某个物品进行了评分。
- **Movielens-1m**：Movielens-1m 是一个电影评分数据集，广泛被用于评估推荐

① <https://github.com/chenchongthu/Recommendation-Unlearning>

② <https://www.yelp.com/dataset/challenge>

③ <https://grouplens.org/datasets/movielens/1m/>

④ <https://grouplens.org/datasets/movielens/10m/>

系统算法的性能。该版本是包含一百万个评分的数据集，其中每个用户至少有 20 个评分。

- **Movielens-10m**: Movielens-10m 是包含一千万个电影评分的数据集，也是我们实验中所用最大的数据集。

三个数据集的基本统计信息见表6.2。

#### 6.4.1.2 基准方法

为了评估所提出的 RecEraser 的效果，我们基于 3 个有代表性的推荐系统模型与当前最先进的数据遗忘方法进行了对比，实验中所用到的推荐系统模型是：

- **BPR**<sup>[6]</sup>: 贝叶斯个性化排序 (Bayesian Personalized Ranking)，通过优化正负样本的偏序损失来训练矩阵分解类的推荐方法。
- **WMF**<sup>[7,52]</sup> 基于矩阵分解和非采样学习的推荐模型，其中训练样例权重设置为固定值。值得一提的是，实验中 WMF 使用了本文在第三章中所提出的高效非采样学习算法进行优化。
- **LightGCN**<sup>[42]</sup>: 轻量化图卷积推荐模型，通过简化图卷积神经网络的设计使其更适合推荐任务，是当前基于图神经网络的表现最好的推荐方法之一。

实验中所对比的数据遗忘方法包括：

- **Retrain**: 最直接的数据遗忘方法，在训练数据中去除要遗忘的数据点并重新训练模型。
- **SISA**<sup>[16]</sup>: 当前最先进的数据遗忘方法之一，将训练数据随机分割成几个不相交的子集，然后在每个子集上训练子模型。最终的预测结果通过平均的方式将子模型聚合得到分数。
- **GraphEraser**<sup>[53]</sup>: 同样是当前最先进的数据遗忘方法之一，在 SISA 的基础上提出了基于图上节点的平衡分割方法，并在聚合时采用了基于学习的方式分配子模型权重，但是子模型权重不会随着不同的预测项而动态改变。

#### 6.4.1.3 评价方式

对每个数据集，我们随机选择 80% 的交互历史数据来构建训练集，10% 作为验证集用于调整超参数，最后 10% 用来做测试集。在模型评价过程中，我们将使用不同算法为每个用户对所有的候选物品进行排序（除了训练集和验证集已知的有正向交互的物品）。为了评价前 K 个推荐结果的表现，Recall@K 和 NDCG@K 将作为评价指标，其中 Recall@K 关注于用户交互过的正例是否被排在前 K 项中，而 NDCG@K 则会细粒度的考虑前 K 项中正例交互的具体位置，这两个指标都是越高越好。

表 6.3 RecEraser 与不同机器遗忘方法的效果比较, \*\* 表示较最好的基准方法有显著提升, 显著性是  $p < 0.01$ 

Yelp2018	BPR				WMF				LightGCN			
	Retrain	SISA	GraphEraser	RecEraser	Retrain	SISA	GraphEraser	RecEraser	Retrain	SISA	GraphEraser	RecEraser
Recall@10	0.0302	0.0151	0.0224	<b>0.0249**</b>	0.0368	0.0235	0.0250	<b>0.0335**</b>	0.0367	0.0221	0.0278	<b>0.0328**</b>
Recall@20	0.0521	0.0272	0.0397	<b>0.0439**</b>	0.0619	0.0401	0.0431	<b>0.0572**</b>	0.0637	0.0392	0.0479	<b>0.0568**</b>
Recall@50	0.1028	0.0566	0.0798	<b>0.0889**</b>	0.1189	0.0785	0.0825	<b>0.1105**</b>	0.1216	0.0768	0.0924	<b>0.1112**</b>
NDCG@10	0.0344	0.0181	0.0254	<b>0.0282**</b>	0.0422	0.0278	0.0295	<b>0.0385**</b>	0.0424	0.0259	0.0319	<b>0.0377**</b>
NDCG@20	0.0423	0.0224	0.0319	<b>0.0353**</b>	0.0512	0.0336	0.0360	<b>0.0471**</b>	0.0524	0.0321	0.0393	<b>0.0465**</b>
NDCG@50	0.0612	0.0334	0.0468	<b>0.0523**</b>	0.0723	0.0478	0.0508	<b>0.0669**</b>	0.0735	0.0461	0.0557	<b>0.0669**</b>
Movielens-1m	BPR				WMF				LightGCN			
	Retrain	SISA	GraphEraser	RecEraser	Retrain	SISA	GraphEraser	RecEraser	Retrain	SISA	GraphEraser	RecEraser
Recall@10	0.1510	0.1080	0.1190	<b>0.1393**</b>	0.1592	0.0759	0.1127	<b>0.1435**</b>	0.1544	0.1089	0.1174	<b>0.1425**</b>
Recall@20	0.2389	0.1737	0.1906	<b>0.2254**</b>	0.2507	0.1266	0.1823	<b>0.2254**</b>	0.2428	0.1743	0.1914	<b>0.2266**</b>
Recall@50	0.4005	0.2970	0.3271	<b>0.3822**</b>	0.4165	0.2309	0.3081	<b>0.3818**</b>	0.4076	0.3084	0.3322	<b>0.3839**</b>
NDCG@10	0.3412	0.2774	0.2844	<b>0.3208**</b>	0.3546	0.2058	0.2575	<b>0.3317**</b>	0.3509	0.2767	0.3832	<b>0.3276**</b>
NDCG@20	0.3368	0.2679	0.2827	<b>0.3186**</b>	0.3509	0.1998	0.2557	<b>0.3253**</b>	0.3447	0.2686	0.2791	<b>0.3229**</b>
NDCG@50	0.3669	0.2840	0.3105	<b>0.3483**</b>	0.3838	0.2174	0.2807	<b>0.3542**</b>	0.3754	0.2905	0.3043	<b>0.3520**</b>
Movielens-10m	BPR				WMF				LightGCN			
	Retrain	SISA	GraphEraser	RecEraser	Retrain	SISA	GraphEraser	RecEraser	Retrain	SISA	GraphEraser	RecEraser
Recall@10	0.1951	0.1620	0.1556	<b>0.1798**</b>	0.2094	0.1332	0.1135	<b>0.1994**</b>	0.1994	0.1221	0.1055	<b>0.1881**</b>
Recall@20	0.3016	0.2507	0.2470	<b>0.2808**</b>	0.3177	0.2142	0.1766	<b>0.3031**</b>	0.3059	0.2033	0.1676	<b>0.2899**</b>
Recall@50	0.4688	0.3906	0.3979	<b>0.4436**</b>	0.4838	0.3515	0.2838	<b>0.4655**</b>	0.4748	0.3314	0.2632	<b>0.4537**</b>
NDCG@10	0.3425	0.3002	0.2688	<b>0.3223**</b>	0.3704	0.2499	0.2209	<b>0.3575**</b>	0.3555	0.2289	0.2139	<b>0.3369**</b>
NDCG@20	0.3534	0.3053	0.2813	<b>0.3319**</b>	0.3784	0.2579	0.2243	<b>0.3642**</b>	0.3635	0.2476	0.2141	<b>0.3352**</b>
NDCG@50	0.3946	0.3366	0.3219	<b>0.3719**</b>	0.4169	0.2918	0.2484	<b>0.4015**</b>	0.4037	0.2808	0.2288	<b>0.3844**</b>

#### 6.4.1.4 超参数设置

所有基准算法的参数设定都是根据对应论文的初始化要求设置的, 且也都在验证集上进行了调参以确保其达到最优的效果。我们对超参数的主要搜索空间包括: 学习率的搜索范围为  $[0.005, 0.01, 0.02, 0.05]$ ,  $L_2$  正则化权重的搜索范围为  $[10^{-5}, 10^{-4}, \dots, 10^{-1}, 1]$ 。特征向量长度  $d$  和  $k$  的搜索范围为  $[16, 32, 64]$ 。最终, 根据预实验结果, 每个训练批的大小被设置成 512, 学习率被设置成 0.05, 特征向量的长度被设置成 64, 而注意力机制的向量长度则为 32。在实验中, 我们还采用了早停的策略, 即当 Recall@10 在验证集上的表现连续 10 轮不再增加时停止训练。

### 6.4.2 性能评估

#### 6.4.2.1 推荐表现分析

我们首先评估了不同数据遗忘方法下的推荐性能。在本小节中, RecEraser 采用了 InBP 方法进行数据分割, 且所有数据遗忘方法的子数据集数设置为 10。为了对比不同长度推荐列表的效果, 我们汇报了推荐列表长度为 10、20 和 50 的结果。实验结果展示于表 6.3 中。

表 6.4 RecEraser 的数据遗忘效率

	Yelp2018			Movielens-1m			Movielens-10m		
	BPR	WMF	LightGCN	BPR	WMF	LightGCN	BPR	WMF	LightGCN
<b>Retrain</b>	400m	75m	1090m	330m	13m	545m	3250m	292m	25330m
子模型训练时长	39m	2.5m	52m	12m	0.9m	17m	240m	14m	610m
<b>RecEraser</b> 聚合模块训练时长	13m	1.3m	34m	10m	0.2m	14m	80m	4m	330m
总时长	52m	3.8m	86m	22m	1.1m	31m	320m	16m	940m

首先可以看到，Retrain 方法通常可以获得最佳的推荐表现，因为它直接在训练数据中去除要遗忘的数据点并重新训练模型，避免了数据分割带来的信息损失。然而，如下一节所示，它的效率较低，尤其是当模型复杂且训练数据集较大时，这降低了其在现实场景中的实用性。

其次，我们提出的 RecEraser 在三个数据集上的表现显著优于最先进的数据遗忘方法 SISA 和 GraphEraser ( $p < 0.01$ )。具体地，在 Yelp2018 数据集上，当使用基本模型 BPR、WMF 和 LightGCN 时，RecEraser 相比于 SISA 的相对提升率分别为 59%、41% 和 46%。与 GraphEraser 相比，相对提升率分别为 11%、32% 和 19%。在其他数据集上，RecEraser 的结果也很显著。在推荐效果上的显著提升可以归因于两个原因：(1) RecEraser 通过所提出的 InBP 方法对数据进行分割，可以更好地保留数据中的用户和物品的结构化信息；(2) RecEraser 在预测时使用所提出的基于注意力的聚合方法自适应地聚合子模型，这可以进一步提高模型的全局表现。

第三，考虑到每个方法在三个数据集上的表现，可以观察到数据遗忘方法的表现与数据集的稀疏度相关。例如，RecEraser 在 Movielens-1m 和 Movielens-10m 上的表现与 Retrain 方法相比损失更小，因为这两个数据集相比 Yelp2018 更加密集。这同时也表明从稀疏的用户-物品交互中更难学习用户偏好，因此应该仔细设置子数据集的数量，以在计算效率和推荐表现上取得平衡。

#### 6.4.2.2 数据遗忘效率分析

在本小节中，我们进行了实验来探索 RecEraser 的数据遗忘效率。遵循之前工作<sup>[53]</sup>的设置，我们从训练数据中随机选择 100 个用户-物品交互进行遗忘，记录相应子模型和聚合部分的训练时间，并计算平均训练时间。在实验中，子模型数量被设置为 10，服务器配置为：英特尔 Xeon9 核心 CPU (2.4GHz)，使用了一张 NVIDIA GeForce GTX TITAN X GPU。结果如表 6.4 所示。

实验结果表明，与 Retrain 方法相比，我们提出的 RecEraser 可以显著提高数据遗忘的效率。例如，在 Movielens-10m 数据集上，RecEraser 在 BPR、WMF 和

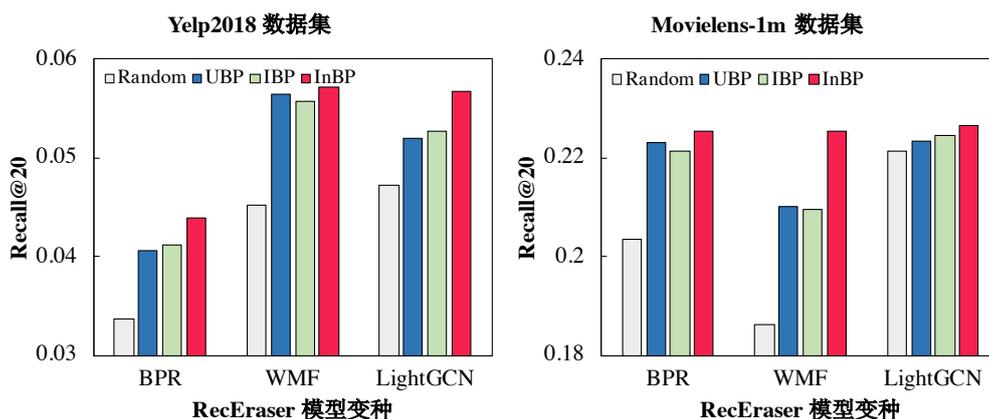


图 6.2 RecEraser 中不同数据分割方式对推荐结果的影响

LightGCN 模型上分别只需要 320 分钟、16 分钟和 920 分钟就可以实现数据遗忘，而 Retrain 方法则分别需要 3250 分钟、292 分钟和 25330 分钟。相比于 Retrain，RecEraser 的加速超过 10 倍。Retrain 方法在大规模数据上非常耗时。而 RecEraser 只需要重新训练相应的子模型和聚合部分，就可以实现快速的数据遗忘。此外，对于较大的数据集，可以分割成更多份的数据集，以进一步在保持推荐性能的同时提高数据遗忘效率。

#### 6.4.2.3 消融实验分析

在本章中，RecEraser 包含三个分别基于用户、物品和交互相似度的平衡数据分割方法 UBP、IBP、和 InBP。我们首先进行消融实验探究不同分割方法对推荐结果的影响，实验结果展示于图6.2中。从图中，我们可以看到，首先本章所提出的数据分割方法，UBP、IBP 和 InBP，与随机方法相比，可以实现更好的推荐效果。以在 Yelp2018 数据集上训练的 LightGCN 为例，对于 InBP，Recall@20 的结果为 0.0568，而对于 Random，结果仅为 0.0472。其次，从总体上看，InBP 的表现通常优于 UBP 和 IBP，这表明基于用户-物品交互的推荐数据集划分比仅基于用户或物品更加有效。

进一步地，为了说明所提出的基于注意力的聚合方法（简记为 AttAgg）的有效性，我们与 MeanAgg 和 StaticAgg 方法进行了比较。其中 MeanAgg 是 SISA<sup>[16]</sup>使用的方法，通过平均所有子模型的预测值来获得聚合分数。StaticAgg 是 GraphEraser<sup>[53]</sup>使用的方法，这是一种基于学习为不同的子模型分配权重的方式，但是子模型权重不会随着不同的预测项而动态改变。图6.3展示了上述方法的表现。可以看到，与 MeanAgg 和 StaticAgg 相比，我们所提出的 AttAgg 方法可以有效地提高推荐的表现。例如，基于 WMF 方法在 Yelp2018 上的表现中，AttAgg 的 Recall@20 分数为 0.0572，而 MeanAgg 和 StaticAgg 的相应结果分别为 0.0475 和 0.0482。这

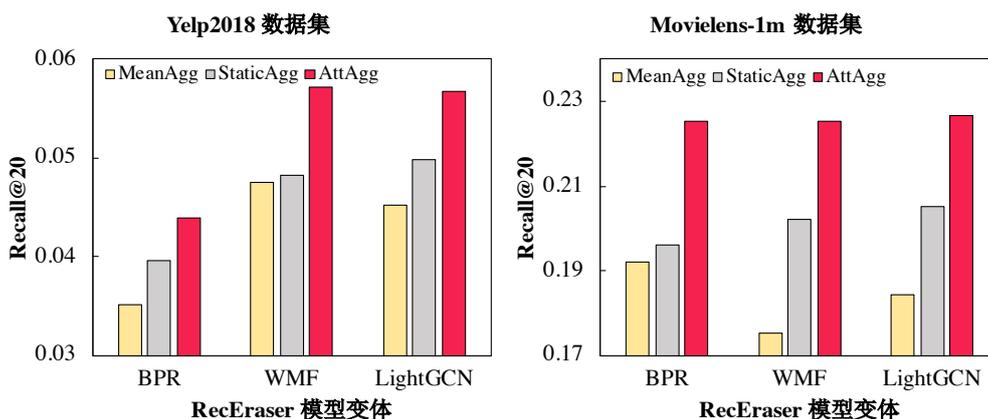


图 6.3 RecEraser 中不同聚合方式对推荐结果的影响

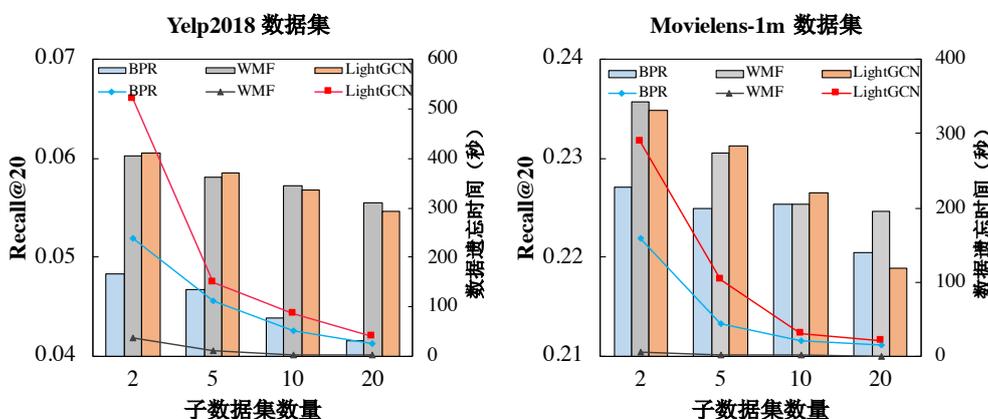


图 6.4 RecEraser 中不同子模型数量对结果的影响

是比较合理的，因为推荐系统在预测时需要面向不同的用户和物品，每个子模型学习到的用户和物品的特征在不同的预测场景时应该有不同的贡献。在预测不同的用户—物品交互时，我们所提出的 AttAgg 可以自适应地为不同的子模型分配权重，从而进一步提升推荐效果。

#### 6.4.2.4 子模型数量影响分析

我们也在本小节进行了实验探索子模型数量对结果的影响。结果展示于图6.4中。对于所有方法，当子模型数量增加时，平均的数据遗忘时间均会减少。这同样是合理的，因为更多的子模型数量意味着每个子模型训练所用的数据规模更小，这会提升数据遗忘的效率。然而，整体数据的分散也会在一定程度上影响推荐模型的表现。因此，在实际使用中应仔细调整子模型数量，以在数据遗忘效率和推荐性能之间寻求平衡。

## 6.5 小结

本章从“系统平台”层面对高效神经网络推荐系统进行了研究。具体地，针对近年来现实应用场景对于数据可遗忘的需求，我们研究了高效可遗忘推荐系统问题，并提出了据我们所知第一个面向推荐系统任务的高效数据遗忘框架 **RecEraser**。为了保持数据的结构化信息，本章首先设计了三种新的平衡数据分割算法。同时，本章也提出了一种基于注意力机制的自适应聚合方法来进一步提高模型效果。为了评估 **RecEraser** 的效果，我们在三个现实数据集上基于三个有代表性的推荐系统模型 **BPR**，**WMF**，和 **LightGCN** 进行了实验。实验结果表明，**RecEraser** 不仅可以实现高效的数据遗忘，而且在推荐表现上优于当前最先进的机器遗忘框架。同时，我们认为本章设计的 **RecEraser** 方法也对未来关于数据遗忘问题的进一步研究具有一定的启发性。本章中对于推荐系统中数据遗忘的设置仍具有一定的局限性，如未考虑批量化遗忘数据的场景。举例来说，系统可以将几个连续的数据遗忘请求聚合成一个批次，然后在一个时间点对这一批数据同时遗忘。目前，基于 **SISA** 的方法如何高效地批量遗忘数据仍然是一个有待解决的问题，我们将在未来对其进行更进一步的探索和研究。

本章相关成果发表于 CCF A 类会议论文 **TheWebConf 2022**<sup>[56]</sup>。

## 第7章 总结与展望

### 7.1 研究工作总结

本文主要研究了高效神经网络推荐问题，以实现精准且高效的个性化推荐系统为目标，从“学习方法”、“推荐模型”、与“系统平台”三个不同的角度，对高效非采样隐式数据学习、高效多场景推荐建模、以及高效可遗忘推荐框架三个问题展开研究，并取得了一系列研究成果。本文的主要贡献和创新点可以总结如下：

**高效非采样隐式数据学习。**推荐系统所常用的隐式数据中负反馈信息是缺失的，且其对用户兴趣建模至关重要。如何精准且高效的获取负反馈信号一直是困扰推荐系统学术界和产业界的难题之一。目前广泛使用的负采样策略稳定性差，使得模型很难收敛到最优的状态。而传统非采样学习方法虽在模型训练效果上更有优势，却因高昂的计算复杂度无法应用于神经网络模型的优化。为了解决上述问题，本文在第3章中通过严谨的数学推理，重构了常用的平方损失函数，并新设计了一系列高效的、可用于神经网络训练的非采样学习方法。将从整体数据中学习的时间复杂度降低了一个数量级，突破了长期以来依赖“采样”的方式训练神经网络模型的瓶颈。成功使用“非采样”获得比传统“采样”方式更快更有效的模型参数学习，从而极大地提高了神经网络模型的实用性能。进一步地，基于一个简单的神经网络矩阵分解模型，本文将所提出的高效非采样学习算法在推荐系统任务上实例化，并命名为高效神经网络矩阵分解模型 ENMF (Efficient Neural Matrix Factorization)。在现实数据集上的实验表明，所提出的 ENMF 方法相对于已有方法推荐精准度提升 5% 以上，训练速度提升 5-30 倍。同时，与现有神经网络推荐系统研究往往聚焦于复杂和先进的模型结构不同，我们从“学习方法”的角度提出了一套新的研究思路来提升推荐效果，所设计的高效非采样算法填补了非采样神经网络推荐模型研究的空白，扩展了神经网络推荐系统研究的广度。这部分内容也为本文随后所设计的高效神经网络推荐模型打下了技术基础。

**高效多场景推荐建模。**基于所设计的高效非采样学习算法，并针对推荐系统在现实应用场景中包含的不同数据特点，本文分别于第4章和第5章研究如何高效且精细化建模结合异质交互行为（社交网络、多类型交互行为）与内容信息（特征信息、知识图谱）的神经网络推荐模型，包括：(1) 在结合社交网络的推荐场景下，本文提出了一个基于自适应迁移学习的高效社交推荐模型 EATNN (Efficient Adaptive Transfer Neural Network) 来联合学习用户社交行为和物品兴趣偏好，实现社交行为的充分利用；(2) 在结合如点击、加入购物车、购买等多类型交互行为的

推荐系统场景下,本文提出了基于因果推理的高效多行为推荐模型 EHCF (Efficient Heterogeneous Collaborative Filtering) 和基于异质图神经网络的高效多行为推荐模型 GHCF (Graph Heterogeneous Collaborative Filtering), 挖掘不同类型行为隐含的用户偏好, 实现对于用户不同交互行为的精准建模; (3) 在结合特征信息的推荐系统场景下, 本文首先通过理论分析建立了因子分解机和矩阵分解之间的转换关系, 进而提出了一个高效非采样分解机模型 ENSFM (Efficient Non-Sampling Factorization Machine) 来全面考虑用户偏好与特征信息之间的关联, 提升推荐效果; (4) 在结合知识图谱的推荐系统场景下, 本文首先通过理论推导将在第一部分所提出的非采样学习算法扩展到三元知识图谱表示学习任务, 然后提出了一个基于非采样联合学习的高效知识增强推荐模型 JNSKR (Jointly Non-Sampling learning model for Knowledge graph enhanced Recommendation)。该模型可以很好地融合知识图谱中的特征信息, 帮助对用户偏好进行充分刻画。本文所提出的以上推荐模型在多个现实数据集上相比已有基准方法在模型表现和训练效率上均取得了显著的提升效果。例如, 在结合多行为数据的推荐任务上, 推荐精准度提升 40% 以上, 训练速度快 10 倍以上。

**高效可遗忘推荐框架。**在许多场景下, 基于隐私保护和坏数据筛查等因素, 推荐系统需要能够高效遗忘某些敏感数据以及从这些数据中学到的参数。针对这一问题, 本文于第6章提出了据我们所知第一个通用的高效可遗忘推荐系统框架 RecEraser, 可以在保持高推荐性能的同时实现高效的数据遗忘。RecEraser 将训练数据划分为多个子集, 并训练子模型, 随后进行模型聚合得到最终的预测结果。为了保持数据的结构化信息, 我们新设计了三种分别基于用户、物品和交互行为相似性的平衡数据划分算法。同时, 考虑到推荐系统的预测要面向独立的用户和物品, 不同子模型对不同的用户—物品的预测过程应是动态变化的。为了进一步提高推荐性能, 我们也提出了一个基于注意力机制的自适应模型聚合方法。RecEraser 作为一个通用的上层框架, 可以和各种推荐系统模型配合使用。我们在三个现实数据集上基于三个有代表性的推荐系统模型 BPR, WMF, 和 LightGCN 进行了实验。实验结果表明, RecEraser 不仅可以实现高效的数据遗忘, 而且在推荐表现上优于当前最先进的机器遗忘框架。机器遗忘作为人工智能的新生领域, 现有的方法仍处于刚起步阶段。目前关于数据遗忘的研究主要集中在计算机视觉和自然语言处理的相关任务上。本文设计的 RecEraser 方法也对未来关于数据遗忘问题的进一步研究具有一定的启发性。

上述工作从“学习方法”、“推荐模型”、与“系统平台”三个不同的角度对高效神经网络推荐问题进行了研究, 对神经网络推荐系统的发展有着基础性的意义,

其创新点也能够应用于其他相关机器学习任务和模型中，具有较强的技术创新且能够创造较好的社会效益。

## 7.2 未来工作展望

本文认为，未来可以继续开展以下研究：

**高效非采样方法的深入研究。**本文的研究表明虽然神经网络结构的改进能够帮助提升推荐算法的表现，但是模型的训练和采样策略对于最终的推荐表现也有十分重要的影响。近年来大部分工作聚焦于如何改进负采样的样例选择过程，对于非采样学习和无负例的训练方法关注较少。这类训练方法的优化可以从根本上提升推荐算法的效果和运行效率，也值得相关领域的学者给予更多关注。虽然本文所提出的高效非采样算法在训练效率和模型表现上展示了非常不错的效果，但其也存在一定的局限性，主要表现在损失函数推导的过程中限制了最终输出层只能为线性预测层，而不能是非线性输出层。针对非线性神经网络模型的通用高效非采样学习算法是一个非常有前景的未来研究方向。同时，本文所研究的高效非采样学习算法是基础性的，本质上，从只有正向交互的数据中学习的机器学习任务和模型都可以使用这一解决方案。未来工作也可以进一步关注其在知识图谱学习、网络表示学习、图表示学习等方向上的应用。另一方面，近年来关于对比学习 (Contrastive Learning) 和自监督学习 (Self-supervised Learning) 的相关研究也为非采样学习提供了一个很好的思路<sup>[156-158]</sup>，其思想是通过让有交互的用户和物品的表示向量相似，同时采用不同的编码方式或限制条件防止模型坍塌（即所有的用户物品向量表示相似），以此来避免训练过程所需要的负样例。目前该方向上的研究仍处于起步阶段，未来工作可以拓展至多场景推荐模型的学习中，如结合异质交互行为和内容特征等。

**可信与安全的推荐系统研究。**最近，数据科学的相关政策使得对于用户隐私的保护越来越受到重视。推荐系统是和用户数据强相关的，因此在隐私保护和安全方面同样具有有较高的需求。本文在第6章提出了据我们所知第一个面向推荐系统任务的高效数据遗忘框架。当前，机器遗忘作为人工智能的新生领域，现有的方法仍处于刚起步阶段。目前关于数据遗忘的研究主要集中在计算机视觉和自然语言处理的相关任务上，在推荐上的研究还比较少。同时，本文中对于推荐系统中数据遗忘的设置仍具有一定的局限性，如未考虑批量化遗忘数据的场景。举例来说，系统可以将几个连续的数据遗忘请求聚合成一个批次，然后在一个时间点对这一批数据同时遗忘。目前，基于 SISA 的方法如何高效地批量遗忘数据仍然是一个有待解决的问题，需要未来对其进行更进一步的探索和研究。另外，本文认为未来工作

也应多关注具有隐私保护的推荐系统模型，如无法从所设计的推荐系统结果中反推出用户的性别，年龄等隐私信息。在技术方面，未来可探索采用对抗网络，多目标优化等方式来在保证推荐系统准确性的同时，也不泄露用户的隐私信息。最后，在可信的推荐系统方面，本文认为也应设计具备准确性以外的评价因素的推荐系统模型，在未来工作中进一步考虑多元性，公平性，可解释性等模型优化目标。

## 参考文献

- [1] Warren J, Marz N. Big data: Principles and best practices of scalable realtime data systems[M]. Simon and Schuster, 2015.
- [2] 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.
- [3] Zhang S, Yao L, Sun A, et al. Deep learning based recommender system: A survey and new perspectives[J]. ACM Computing Surveys (CSUR), 2019, 52(1): 1-38.
- [4] Chen C, Zhang M, Liu Y, et al. Neural attentional rating regression with review-level explanations[C]//Proceedings of the 2018 World Wide Web Conference. 2018: 1583-1592.
- [5] Ding J, Quan Y, Yao Q, et al. Simplify and robustify negative sampling for implicit collaborative filtering[C]//In Advances in Neural Information Processing Systems. 2020: 1094-1105.
- [6] Rendle S, Freudenthaler C, Gantner Z, et al. Bpr: Bayesian personalized ranking from implicit feedback[C]//Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. 2009: 452-461.
- [7] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets[C]//2008 Eighth IEEE International Conference on Data Mining. Ieee, 2008: 263-272.
- [8] Liang D, Charlin L, McInerney J, et al. Modeling user exposure in recommendation[C]//Proceedings of the 25th international conference on World Wide Web. 2016: 951-961.
- [9] He X, Zhang H, Kan M Y, et al. Fast matrix factorization for online recommendation with implicit feedback[C]//Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. 2016: 549-558.
- [10] Goodfellow I, Bengio Y, Courville A, et al. Deep learning: volume 1[M]. MIT press Cambridge, 2016.
- [11] Schwartz R, Dodge J, Smith N A, et al. Green ai[J]. Communications of the ACM, 2020, 63(12): 54-63.
- [12] Chen C, Zhang M, Liu Y, et al. Social attentional memory network: Modeling aspect-and friend-level differences in recommendation[C]//Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. 2019: 177-185.
- [13] Gao C, He X, Gan D, et al. Neural multi-task recommendation from multi-behavior data[C]//2019 IEEE 35th international conference on data engineering (ICDE). IEEE, 2019: 1554-1557.
- [14] Cheng H T, Koc L, Harmsen J, et al. Wide & deep learning for recommender systems[C]//Proceedings of the 1st workshop on deep learning for recommender systems. 2016: 7-10.
- [15] Wang H, Zhang F, Xie X, et al. Dkn: Deep knowledge-aware network for news recommendation [C]//Proceedings of the 2018 world wide web conference. 2018: 1835-1844.
- [16] Bourtole L, Chandrasekaran V, Choquette-Choo C A, et al. Machine unlearning[C]//2021 IEEE Symposium on Security and Privacy (SP). IEEE, 2021: 141-159.

- 
- [17] Chen M, Zhang Z, Wang T, et al. Graph unlearning[J]. arXiv preprint arXiv:2103.14991, 2021.
- [18] Jagielski M, Oprea A, Biggio B, et al. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning[C]//2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018: 19-35.
- [19] Devooght R, Kourtellis N, Mantrach A. Dynamic matrix factorization with priors on unknown values[C]//Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2015: 189-198.
- [20] He X, Liao L, Zhang H, et al. Neural collaborative filtering[C]//Proceedings of the 26th international conference on world wide web. 2017: 173-182.
- [21] Rendle S, Freudenthaler C. Improving pairwise learning for item recommendation from implicit feedback[C]//Proceedings of the 7th ACM international conference on Web search and data mining. 2014: 273-282.
- [22] Chen J, Wang C, Zhou S, et al. Fast adaptively weighted matrix factorization for recommendation with implicit feedback[C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 34. 2020: 3470-3477.
- [23] Pilászy I, Zibriczky D, Tikk D. Fast als-based matrix factorization for explicit and implicit feedback datasets[C]//Proceedings of the fourth ACM conference on Recommender systems. 2010: 71-78.
- [24] 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.
- [25] Zheng L, Noroozi V, Yu P S. Joint deep modeling of users and items using reviews for recommendation[C]//Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, 2017: 425-434.
- [26] Zhao T, McAuley J, King I. Leveraging social connections to improve personalized ranking for collaborative filtering[C]//Proceedings of the 23rd ACM international conference on conference on information and knowledge management. 2014: 261-270.
- [27] Tang L, Long B, Chen B C, et al. An empirical study on recommendation with multiple types of feedback[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016: 283-292.
- [28] He X, Chua T S. Neural factorization machines for sparse predictive analytics[C]//Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. 2017: 355-364.
- [29] Wang X, He X, Cao Y, et al. Kgat: Knowledge graph attention network for recommendation [C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 950-958.
- [30] Xiao L, Min Z, Yongfeng Z, et al. Learning and transferring social and item visibilities for personalized recommendation[C]//Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 2017: 337-346.

- 
- [31] Gao C, He X, Gan D, et al. Learning to recommend with multiple cascading behaviors[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2019.
- [32] Zhao Z, Hong L, Wei L, et al. Recommending what video to watch next: a multitask ranking system[C]//*Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, 2019: 43-51.
- [33] Chen J, Zhang H, He X, et al. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention[C]//*Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 2017: 335-344.
- [34] Xiao J, Ye H, He X, et al. Attentional factorization machines: learning the weight of feature interactions via attention networks[C]//*Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2017: 3119-3125.
- [35] Ebesu T, Shen B, Fang Y. Collaborative memory network for recommendation systems[C]//*The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018: 515-524.
- [36] He X, Du X, Wang X, et al. Outer product-based neural collaborative filtering[C]//*Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2018: 2227-2233.
- [37] Xin X, Chen B, He X, et al. Cfm: Convolutional factorization machines for context-aware recommendation.[C]//*IJCAI: volume 19*. 2019: 3926-3932.
- [38] Sun F, Liu J, Wu J, et al. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer[C]//*Proceedings of the 28th ACM international conference on information and knowledge management*. 2019: 1441-1450.
- [39] He X, He Z, Du X, et al. Adversarial personalized ranking for recommendation[C]//*The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 2018: 355-364.
- [40] Wang J, Yu L, Zhang W, et al. Irgan: A minimax game for unifying generative and discriminative information retrieval models[C]//*Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 2017: 515-524.
- [41] Wang X, He X, Wang M, et al. Neural graph collaborative filtering[C]//*Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 2019: 165-174.
- [42] He X, Deng K, Wang X, et al. Lightgcn: Simplifying and powering graph convolution network for recommendation[C]//*Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 2020: 639-648.
- [43] Xin X, Yuan F, He X, et al. Batch is not heavy: Learning word representations from all samples [C]//*Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018: 1853-1862.
- [44] Cao Y, Yang J. Towards making systems forget with machine unlearning[C]//*2015 IEEE Symposium on Security and Privacy*. IEEE, 2015: 463-480.
- [45] Ginart A A, Guan M Y, Valiant G, et al. Making ai forget you: data deletion in machine learning [C]//*Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 2019: 3518-3531.

- 
- [46] Gupta V, Jung C, Neel S, et al. Adaptive machine unlearning[J]. arXiv preprint arXiv:2106.04378, 2021.
- [47] Rendle S. Factorization machines[C]//2010 IEEE International conference on data mining. IEEE, 2010: 995-1000.
- [48] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [49] Chen C, Zhang M, Zhang Y, et al. Efficient neural matrix factorization without sampling for recommendation[J]. ACM Transactions on Information Systems (TOIS), 2020, 38(2): 1-28.
- [50] Chen C, Ma W, Zhang M, et al. Revisiting negative sampling vs. non-sampling in implicit recommendation[J]. ACM Transactions on Information Systems (TOIS), 2022.
- [51] Chen C, Zhang M, Wang C, et al. An efficient adaptive transfer neural network for social-aware recommendation[C]//Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019: 225-234.
- [52] Chen C, Zhang M, Zhang Y, et al. Efficient heterogeneous collaborative filtering without negative sampling for recommendation[C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 34. 2020: 19-26.
- [53] Chen C, Ma W, Zhang M, et al. Graph heterogeneous multi-relational recommendation[C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 35. 2021: 3958-3966.
- [54] Chen C, Zhang M, Ma W, et al. Efficient non-sampling factorization machines for optimal context-aware recommendation[C]//Proceedings of The Web Conference 2020. 2020: 2400-2410.
- [55] Chen C, Zhang M, Ma W, et al. Jointly non-sampling learning for knowledge graph enhanced recommendation[C]//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020: 189-198.
- [56] Chen C, Sun F, Zhang M, et al. Recommendation unlearning[C]//Proceedings of The Web Conference 2022. 2022.
- [57] Ji G, He S, Xu L, et al. Knowledge graph embedding via dynamic mapping matrix[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015: 687-696.
- [58] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems. 2013: 3111-3119.
- [59] Lian D, Liu Q, Chen E. Personalized ranking with importance sampling[C]//Proceedings of The Web Conference 2020. 2020: 1093-1103.
- [60] Zhang W, Chen T, Wang J, et al. Optimizing top-n collaborative filtering via dynamic negative item sampling[C]//Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. 2013: 785-788.
- [61] Chae D K, Kang J S, Kim S W, et al. Cfgan: A generic collaborative filtering framework based on generative adversarial networks[C]//Proceedings of the 27th ACM international conference on information and knowledge management. 2018: 137-146.

- 
- [62] Jin B, Lian D, Liu Z, et al. Sampling-decomposable generative adversarial recommender[J]. *Advances in Neural Information Processing Systems*, 2020, 33.
- [63] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//*Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*. 2014: 2672-2680.
- [64] Ying R, He R, Chen K, et al. Graph convolutional neural networks for web-scale recommender systems[C]//*Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018: 974-983.
- [65] Yang Z, Ding M, Zhou C, et al. Understanding negative sampling in graph representation learning[C]//*Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020: 1666-1676.
- [66] Huang T, Dong Y, Ding M, et al. Mixgcf: An improved training method for graph neural network-based recommender systems[J]. 2021.
- [67] Yuan F, Xin X, He X, et al. fbgd: Learning embeddings from positive unlabeled data with bgd [J]. 2018.
- [68] Cremonesi P, Koren Y, Turrin R. Performance of recommender algorithms on top-n recommendation tasks[C]//*Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010: 39-46.
- [69] Herlocker J L, Konstan J A, Borchers A, et al. An algorithmic framework for performing collaborative filtering[C]//*Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 1999: 230-237.
- [70] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//*Proceedings of the 10th international conference on World Wide Web*. 2001: 285-295.
- [71] Zhang S, Yao L, Xu X. Autosvd++: An efficient hybrid collaborative filtering model via contractive auto-encoders[C]//*Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2017: 957-960.
- [72] Koren Y. Collaborative filtering with temporal dynamics[C]//*Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009: 447-456.
- [73] Xue F, He X, Wang X, et al. Deep item-based collaborative filtering for top-n recommendation [J]. *ACM Transactions on Information Systems (TOIS)*, 2019, 37(3): 33.
- [74] Chen X, Xu H, Zhang Y, et al. Sequential recommendation with user memory networks[C]//*Proceedings of the eleventh ACM international conference on web search and data mining*. ACM, 2018: 108-116.
- [75] Wang T H, Hu X, Jin H, et al. Autorec: An automated recommender system[C]//*Fourteenth ACM Conference on Recommender Systems*. 2020: 582-584.
- [76] Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks[C]//*Proceedings of the fourth ACM conference on Recommender systems*. 2010: 135-142.

- 
- [77] Ma H, Zhou D, Liu C, et al. Recommender systems with social regularization[C]//Proceedings of the fourth ACM international conference on Web search and data mining. 2011: 287-296.
- [78] Du L, Li X, Shen Y D. User graph regularized pairwise matrix factorization for item recommendation[C]//International Conference on Advanced Data Mining and Applications. Springer, 2011: 372-385.
- [79] Rafailidis D, Crestani F. Joint collaborative ranking with social relationships in top-n recommendation[C]//Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. 2016: 1393-1402.
- [80] Singh A P, Gordon G J. Relational learning via collective matrix factorization[C]//Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 2008: 650-658.
- [81] Krohn-Grimberghe A, Drumond L, Freudenthaler C, et al. Multi-relational matrix factorization using bayesian personalized ranking for social network data[C]//Proceedings of the fifth ACM international conference on Web search and data mining. 2012: 173-182.
- [82] Pan W, Xiang E, Liu N, et al. Transfer learning in collaborative filtering for sparsity reduction [C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 24. 2010.
- [83] Ding J, Yu G, He X, et al. Improving implicit recommender systems with view data.[C]//IJCAI. 2018: 3343-3349.
- [84] Loni B, Pagano R, Larson M, et al. Bayesian personalized ranking with multi-channel user feedback[C]//Proceedings of the 10th ACM Conference on Recommender Systems. 2016: 361-364.
- [85] Rendle S, Gantner Z, Freudenthaler C, et al. Fast context-aware recommendations with factorization machines[C]//Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. 2011: 635-644.
- [86] Wang R, Fu B, Fu G, et al. Deep & cross network for ad click predictions[M]//Proceedings of the ADKDD'17. 2017: 1-7.
- [87] Wang X, He X, Feng F, et al. Tem: Tree-enhanced embedding model for explainable recommendation[C]//Proceedings of the 2018 World Wide Web Conference. 2018: 1543-1552.
- [88] Guo H, Tang R, Ye Y, et al. Deepfm: a factorization-machine based neural network for ctr prediction[C]//Proceedings of the 26th International Joint Conference on Artificial Intelligence. 2017: 1725-1731.
- [89] 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 & Data Mining. 2018: 1754-1763.
- [90] Qiang R, Liang F, Yang J. Exploiting ranking factorization machines for microblog retrieval [C]//Proceedings of the 22nd ACM international conference on Information & Knowledge Management. 2013: 1783-1788.
- [91] Yuan F, Guo G, Jose J M, et al. Lambdafm: learning optimal ranking with factorization machines using lambda surrogates[C]//Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. 2016: 227-236.

- 
- [92] Yu X, Ren X, Sun Y, et al. Recommendation in heterogeneous information networks with implicit user feedback[C]//Proceedings of the 7th ACM conference on Recommender systems. 2013: 347-350.
- [93] Wang H, Zhang F, Wang J, et al. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems[C]//Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018: 417-426.
- [94] Zhao H, Yao Q, Li J, et al. Meta-graph based recommendation fusion over heterogeneous information networks[C]//Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017: 635-644.
- [95] Wang X, Wang D, Xu C, et al. Explainable reasoning over knowledge graphs for recommendation[C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 33. 2019: 5329-5336.
- [96] Wang H, Zhang F, Zhao M, et al. Multi-task feature learning for knowledge graph enhanced recommendation[C]//The World Wide Web Conference. 2019: 2000-2010.
- [97] Golatkar A, Achille A, Soatto S. Eternal sunshine of the spotless net: Selective forgetting in deep networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 9304-9312.
- [98] Izzo Z, Smart M A, Chaudhuri K, et al. Approximate data deletion from machine learning models[C]//International Conference on Artificial Intelligence and Statistics. PMLR, 2021: 2008-2016.
- [99] Guo C, Goldstein T, Hannun A, et al. Certified data removal from machine learning models [C]//International Conference on Machine Learning. PMLR, 2020: 3832-3842.
- [100] Neel S, Roth A, Sharifi-Malvajerdi S. Descent-to-delete: Gradient-based methods for machine unlearning[C]//Algorithmic Learning Theory. PMLR, 2021: 931-962.
- [101] Chen K, Huang Y, Wang Y. Machine unlearning via gan[J]. arXiv preprint arXiv:2111.11869, 2021.
- [102] Zhang H, Shen F, Liu W, et al. Discrete collaborative filtering[C]//Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, 2016: 325-334.
- [103] Dacrema M F, Cremonesi P, Jannach D. Are we really making much progress? a worrying analysis of recent neural recommendation approaches[J]. arXiv preprint arXiv:1907.06902, 2019.
- [104] Yi X, Yang J, Hong L, et al. Sampling-bias-corrected neural modeling for large corpus item recommendations[C]//Proceedings of the 13th ACM Conference on Recommender Systems. 2019: 269-277.
- [105] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization.[J]. Journal of machine learning research, 2011, 12(7).
- [106] Rendle S, Krichene W, Zhang L, et al. Neural collaborative filtering vs. matrix factorization revisited[C]//Fourteenth ACM Conference on Recommender Systems. 2020: 240-248.

- 
- [107] Krichene W, Rendle S. On sampled metrics for item recommendation[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 1748-1757.
- [108] Ren Z, Liang S, Li P, et al. Social collaborative viewpoint regression with explainable recommendations[C]//Proceedings of the tenth ACM international conference on web search and data mining. 2017: 485-494.
- [109] Yang B, Lei Y, Liu J, et al. Social collaborative filtering by trust[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(8): 1633-1647.
- [110] Liao L, He X, Zhang H, et al. Attributed social network embedding[J]. IEEE Transactions on Knowledge and Data Engineering, 2018.
- [111] Jiang M, Cui P, Chen X, et al. Social recommendation with cross-domain transferable knowledge [J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(11): 3084-3097.
- [112] Zhang X, Cheng J, Yuan T, et al. Toprec: domain-specific recommendation through community topic mining in social network[C]//Proceedings of the 22nd international conference on World Wide Web. 2013: 1501-1510.
- [113] Bruna J, Zaremba W, Szlam A, et al. Spectral networks and locally connected networks on graphs[J]. arXiv preprint arXiv:1312.6203, 2013.
- [114] Wang X, He X, Nie L, et al. Item silk road: Recommending items from information domains to social users[C]//Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. 2017: 185-194.
- [115] Sun P, Wu L, Wang M. Attentive recurrent social recommendation[C]//The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018: 185-194.
- [116] Pan W, Liu M, Ming Z. Transfer learning for heterogeneous one-class collaborative filtering[J]. IEEE Intelligent Systems, 2016, 31(4): 43-49.
- [117] Zhou X, Liu D, Lian J, et al. Collaborative metric learning with memory network for multi-relational recommender systems[C]//Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2019: 4454-4460.
- [118] Berg R v d, Kipf T N, Welling M. Graph convolutional matrix completion[J]. arXiv preprint arXiv:1706.02263, 2017.
- [119] Marcheggiani D, Titov I. Encoding sentences with graph convolutional networks for semantic role labeling[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017: 1506-1515.
- [120] Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling relational data with graph convolutional networks[C]//European semantic web conference. Springer, 2018: 593-607.
- [121] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.
- [122] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]//Icml. 2010.
- [123] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The journal of machine learning research, 2014, 15(1): 1929-1958.

- 
- [124] Evgeniou A, Pontil M. Multi-task feature learning[J]. *Advances in neural information processing systems*, 2007, 19: 41.
- [125] Hamilton W L, Ying R, Leskovec J. Inductive representation learning on large graphs[C]// *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017: 1025-1035.
- [126] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. *arXiv preprint arXiv:1710.10903*, 2017.
- [127] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[J]. *Advances in neural information processing systems*, 2013, 26.
- [128] Vashishth S, Sanyal S, Nitin V, et al. Composition-based multi-relational graph convolutional networks[J]. *arXiv preprint arXiv:1911.03082*, 2019.
- [129] Maas A L, Hannun A Y, Ng A Y, et al. Rectifier nonlinearities improve neural network acoustic models[C]//*Proc. icml: volume 30*. Citeseer, 2013: 3.
- [130] Ma W, Zhang M, Cao Y, et al. Jointly learning explainable rules for recommendation with knowledge graph[C]//*The world wide web conference*. 2019: 1210-1221.
- [131] Blondel M, Fujino A, Ueda N, et al. Higher-order factorization machines[J]. *Advances in Neural Information Processing Systems*, 2016, 29.
- [132] Chen Y, Ren P, Wang Y, et al. Bayesian personalized feature interaction selection for factorization machines[C]//*Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019: 665-674.
- [133] Xian Y, Fu Z, Muthukrishnan S, et al. Reinforcement knowledge graph reasoning for explainable recommendation[C]//*Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 2019: 285-294.
- [134] Cao Y, Wang X, He X, et al. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences[C]//*The world wide web conference*. 2019: 151-161.
- [135] Juan Y, Zhuang Y, Chin W S, et al. Field-aware factorization machines for ctr prediction[C]// *Proceedings of the 10th ACM conference on recommender systems*. 2016: 43-50.
- [136] Shi Y, Zhu Q, Guo F, et al. Easing embedding learning by comprehensive transcription of heterogeneous information networks[C]//*Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018: 2190-2199.
- [137] Zhang F, Yuan N J, Lian D, et al. Collaborative knowledge base embedding for recommender systems[C]//*Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016: 353-362.
- [138] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion[C]//*Twenty-ninth AAAI conference on artificial intelligence*. 2015.
- [139] Wang H, Zhao M, Xie X, et al. Knowledge graph convolutional networks for recommender systems[C]//*The world wide web conference*. 2019: 3307-3313.
- [140] Chen T, Zhang W, Lu Q, et al. Svdfeature: a toolkit for feature-based collaborative filtering[J]. *The Journal of Machine Learning Research*, 2012, 13(1): 3619-3622.

- 
- [141] Baltrunas L, Church K, Karatzoglou A, et al. Frappe: Understanding the usage and perception of mobile app recommendations in-the-wild[J]. arXiv preprint arXiv:1505.03014, 2015.
- [142] Bayer I, He X, Kanagal B, et al. A generic coordinate descent framework for learning from implicit feedback[C]//Proceedings of the 26th International Conference on World Wide Web. 2017: 1341-1350.
- [143] Ai Q, Azizi V, Chen X, et al. Learning heterogeneous knowledge base embeddings for explainable recommendation[J]. Algorithms, 2018, 11(9): 137.
- [144] Yang B, Yih W t, He X, et al. Embedding entities and relations for learning and inference in knowledge bases[J]. arXiv preprint arXiv:1412.6575, 2014.
- [145] Xin X, He X, Zhang Y, et al. Relational collaborative filtering: Modeling multiple item relations for recommendation[C]//Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. 2019: 125-134.
- [146] Volkovs M, Yu G, Poutanen T. Dropoutnet: Addressing cold start in recommender systems[J]. Advances in neural information processing systems, 2017, 30.
- [147] Chen M, Zhang Z, Wang T, et al. When machine unlearning jeopardizes privacy[C]//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. 2021: 896-911.
- [148] Carlini N, Liu C, Erlingsson U, et al. The secret sharer: Evaluating and testing unintended memorization in neural networks[C]//Proceedings of the 28th USENIX Conference on Security Symposium. USA: USENIX Association, 2019: 267-284.
- [149] Jagielski M, Oprea A, Biggio B, et al. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning[C]//2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018: 19-35.
- [150] Wang C, Ma W, Zhang M, et al. Toward dynamic user intention: Temporal evolutionary effects of item relations in sequential recommendation[J]. ACM Transactions on Information Systems (TOIS), 2020, 39(2): 1-33.
- [151] Sekhari A, Acharya J, Kamath G, et al. Remember what you want to forget: Algorithms for machine unlearning[J]. Advances in Neural Information Processing Systems, 2021, 34.
- [152] Thudi A, Deza G, Chandrasekaran V, et al. Unrolling sgd: Understanding factors influencing machine unlearning[J]. arXiv preprint arXiv:2109.13398, 2021.
- [153] Girvan M, Newman M E. Community structure in social and biological networks[J]. Proceedings of the national academy of sciences, 2002, 99(12): 7821-7826.
- [154] Choi M, Jeong Y, Lee J, et al. Local collaborative autoencoders[C]//Proceedings of the 14th ACM International Conference on Web Search and Data Mining. 2021: 734-742.
- [155] Kanungo T, Mount D M, Netanyahu N S, et al. An efficient k-means clustering algorithm: Analysis and implementation[J]. IEEE transactions on pattern analysis and machine intelligence, 2002, 24(7): 881-892.
- [156] Grill J B, Strub F, Altché F, et al. Bootstrap your own latent: A new approach to self-supervised learning[C]//Neural Information Processing Systems. 2020.

- [157] Lee D, Kang S, Ju H, et al. Bootstrapping user and item representations for one-class collaborative filtering[J]. arXiv preprint arXiv:2105.06323, 2021.
- [158] Zhou X, Sun A, Liu Y, et al. Selfcf: A simple framework for self-supervised collaborative filtering[J]. arXiv preprint arXiv:2107.03019, 2021.

## 致 谢

回顾过去五年的博士生涯，感慨万千。在此，我想向那些曾经鼓励过、帮助过、支持过我的人表示由衷的谢意。

衷心感谢我的导师张敏副教授对我学业、科研和生活上的悉心指导。张老师既是良师、也是益友，她对科研的热情和严谨、对学术前沿的敏锐直觉、对我从始至终的鼓励和肯定深深地感染着我，也激励着我。感谢课题组马少平教授为我提供一流的学习和科研条件，给予我的关怀和支持，带我进入马家军这一温暖的大家庭。感谢课题组刘奕群教授、金奕江工程师在研究工作中给予我的支持和帮助。我的进步和成长离不开各位老师的全力支持、耐心指导与殷切关心。能够成为清华大学信息检索课题组的一分子，我非常珍惜也倍感荣幸。

感谢美国罗格斯大学的张永峰助理教授在我交流学习期间的关照和悉心指导，在合作中张老师严谨创新的科研精神令我受益匪浅。同时感谢访学期间陈瀚雄师兄对我生活和科研工作的帮助。

感谢课题组马为之、林肖师兄在我进组科研的初期就对我尽心的指导和帮助；感谢毛佳昕、罗成、刘斌、苏宁、卢泓宇、谢晓晖等师兄在我科研和生活上的鼓励和帮助；感谢我的舍友张俊祺，同期施韶韵、郑玉昆、刘梦旻、吴之璟、李祥圣、邵英玮等同学的鼓励、支持和陪伴；感谢王晨阳、阎世宏、杨俊、王振、李佳玉、王亦凡、汪佳茵、李阳崑等同学在我科研生活中的重要帮助；感谢信息检索组所有成员对我的支持和帮助！

在此，我还要感谢很多人：感谢中国人民大学宋睿华副教授、上海交通大学张娅副教授、张伟楠副教授、俞勇教授、杨小康教授、华东师范大学张伟副研究员、香港中文大学（深圳）查宏远教授在职业发展和生涯规划方面给我的关怀和帮助；感谢阿里巴巴的孙飞老师、丁博麟老师、腾讯的杨月奎老师、华为的何秀强老师、王钊伟老师、田奇老师在科研合作和职业发展等方面对我的帮助和支持！

最后，我要衷心地感谢我的父母和家人始终给予我无条件的支持；感谢爱人黄又黄一路的陪伴、关怀和无数次的支持和鼓励。你们是我生活中的港湾和奋斗下去的动力。

声 明

---

声 明

本人郑重声明：所提交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名： 陈冲 日 期： 2022.3.2 | .

## 个人简历、在学期间完成的相关学术成果

### 个人简历

1997年2月22日出生于河南省商丘市。

2013年9月考入清华大学计算机科学与技术系计算机科学与技术专业，2017年7月本科毕业并获得工学学士学位。

2017年9月免试进入清华大学计算机科学与技术系攻读人工智能博士学位至今。

### 在学期间完成的相关学术成果

#### 第一作者学术论文：

- [1] **Chong Chen**, Min Zhang, Fei Sun and Bolin Ding. Recommendation Unlearning. The Web Conference 2022. (Full paper, TheWebConf 2022, CCF-A类, Th-CPL-A类)
- [2] **Chong Chen**, Min Zhang, Yiqun Liu and Shaoping Ma. Revisiting Negative Sampling VS. Non-Sampling in Implicit Recommendation. ACM Transactions on Information Systems. (Journal paper, TOIS, CCF-A类, Th-CPL-A类)
- [3] **Chong Chen**, Weizhi Ma, Min Zhang, Zhaowei Wang, Xiuqiang He, Chenyang Wang, Yiqun Liu and Shaoping Ma. Graph Heterogeneous Multi-Relational Recommendation. The 45th AAAI Conference on Artificial Intelligence. (Full paper, AAAI 2021, CCF-A类, Th-CPL-A类)
- [4] **Chong Chen**, Min Zhang, Weizhi Ma, Yiqun Liu and Shaoping Ma. Jointly Non-Sampling Learning for Knowledge Graph Enhanced Recommendation. The 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. (Full paper, SIGIR 2020, CCF-A类, Th-CPL-A类)
- [5] **Chong Chen**, Min Zhang, Weizhi Ma, Yiqun Liu and Shaoping Ma. Efficient Non-Sampling Factorization Machines for Optimal Context-Aware Recommendation. The Web Conference 2020 (Full paper, TheWebConf 2020, CCF-A类, Th-CPL-A类)
- [6] **Chong Chen**, Min Zhang, Yongfeng Zhang, Yiqun Liu and Shaoping Ma. Efficient Neural Matrix Factorization without Sampling for Recommendation. ACM Transactions on Information Systems. (Journal paper, TOIS Vol. 38, No. 2, CCF-A类, Th-CPL-A类)
- [7] **Chong Chen**, Min Zhang, Weizhi Ma, Yongfeng Zhang, Yiqun Liu and Shaoping

- Ma. Efficient Heterogeneous Collaborative Filtering without Negative Sampling for Recommendation. The 44th AAAI Conference on Artificial Intelligence. (Full paper, AAAI 2020, CCF-A 类, Th-CPL-A 类)
- [8] **Chong Chen**, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu and Shaoping Ma. An Efficient Adaptive Transfer Neural Network for Social-aware Recommendation. The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. (Full paper, SIGIR 2019, CCF-A 类, Th-CPL-A 类)
- [9] **Chong Chen**, Min Zhang, Yiqun Liu and Shaoping Ma. Social Attentional Memory Network: Modeling Aspect- and Friend-level Differences in Recommendation. The 12th ACM International Conference on Web Search and Data Mining. (Full paper, WSDM 2019, CCF-B 类, Th-CPL-A 类)
- [10] **Chong Chen**, Min Zhang, Yiqun Liu and Shaoping Ma. Neural Attentional Rating Regression with Review-level Explanations The Web Conference 2018. (Full paper, TheWebConf 2018, CCF-A 类, Th-CPL-A 类)
- [11] **Chong Chen**, Min Zhang, Yiqun Liu and Shaoping Ma. Missing Data Modeling with User Activity and Item Popularity in Recommendation. The fourteenth Asia Information Retrieval Societies Conference. (Full paper, AIRS 2018, EI-indexed, Best Paper Honorable Mention)

### 合著学术论文:

- [12] Chenyang Wang, Weizhi Ma, Min Zhang, **Chong Chen**, Yiqun Liu and Shaoping Ma. Towards Dynamic User Intention: Temporal Evolution of Item Relations in Sequential Recommendation. ACM Transactions on Information Systems. (Journal paper, TOIS Vol. 39, No. 2, CCF-A 类, Th-CPL-A 类)
- [13] Zelong Li, Jianchao Ji, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, **Chong Chen** and Yongfeng Zhang. Efficient Knowledge Graph Embedding without Negative Sampling. The Web Conference 2021. (Full paper, WWW 2021, CCF-A 类, Th-CPL-A 类)
- [14] Chenyang Wang, Weizhi Ma, **Chong Chen**, Min Zhang, Yiqun Liu and Shaoping Ma. Sequential Recommendation with Multiple Contrast Signals. ACM Transactions on Information Systems. (Journal paper, TOIS, CCF-A 类, Th-CPL-A 类)

### 专利:

- [15] **陈冲**, 张敏, 马为之, 王钊伟, 何秀强, 刘奕群, 马少平. 一种基于知识图谱与用户多种交互行为的神经网络推荐方法: 中国, 2021107427281 (专利申请号)

### 主要奖励与荣誉：

- [16] 蒋南翔奖学金，2021 年
- [17] 清华大学特等奖学金提名，2021 年
- [18] 惠研英才奖学金，2021 年
- [19] 百度奖学金（全球 10 名同学获奖），2020 年
- [20] 字节奖学计划（全国 10 名同学获奖），2020 年
- [21] 钟士模奖学金，2020 年
- [22] 清华大学计算机系学术新秀，2020 年
- [23] ‘84’ 创新未来奖学金，2020 年
- [24] 国家奖学金，2020、2019 年
- [25] AIRS 2018 最佳论文提名奖，2018 年

## 指导教师学术评语

推荐系统技术作为解决信息过载的有效手段，是当前人工智能最成功的一个应用场景。论文研究高效且精准的神经网络推荐系统的理论和方法，这也是研究与产业界共同关注的一个重要问题，选题具有重要的理论意义和实用价值。

作者独立完成了论文中的全部内容，开展了系统深入的研究，其主要工作和创新点如下：

1、针对海量隐式数据的高效学习问题，提出了神经网络训练的非采样学习框架，并设计了一系列非采样学习算法，将基于全样本数据的学习的时间复杂度降低了一个数量级。

2、针对不同的推荐应用场景特点，提出了结合异质交互行为与内容信息的精细化高效神经网络推荐模型，在公开数据集上的实验结果验证了所提出模型在推荐效果和训练效率上均取得了显著提升。

3、针对推荐系统的数据可遗忘需求，提出了一个根据数据遗忘请求来去除相应数据对于推荐结果的影响的通用框架，在真实场景中的实验结果表明，所提出方法能够在保证推荐精度的前提下实现高效的数据遗忘。

论文文献综述全面，观点明确，具有很高的创新性。所提出和使用的公式、模型描述正确，实验方法得当，实验结果可信，结论有说服力。该论文表现出作者已经掌握坚实的基础理论和系统的专门知识，并具有出色的独立科研能力。该博士生政治立场正确，科研态度端正，学术作风好，具有很好的英文阅读、写作与沟通能力。

综上所述，论文已经达到博士学位论文水平，同意答辩。

## 答辩委员会决议书

论文从理论分析和算法研究角度，在高效非采样隐式数据学习、高效多场景推荐建模及高效可遗忘推荐框架三个方面展开了系统研究，提出一系列创新算法，选题具有重要的理论意义和实用价值。

论文的主要工作和创新成果如下：

- 1、针对海量隐式数据的高效学习问题，提出了高效非采样学习算法，大幅降低了全样本数据学习的复杂度，提高了神经网络模型的实用性。
- 2、针对推荐系统不同应用场景的数据特点，提出多个结合异质交互行为与内容信息的神经网络推荐模型，在性能和效率上均取得了显著的效果提升。
- 3、针对推荐系统中的数据可遗忘需求，设计了一个通用的高效可遗忘推荐系统框架，所提方法能够在保证推荐精度的前提下实现高效的数据遗忘。

论文工作表明，该同学已经掌握了本学科坚实宽广的基础理论和系统深入的专门知识，独立从事科学研究工作能力强。论文写作逻辑清晰，行文流畅，结果详实，写作规范，达到了工学博士学位论文水平，是一篇优秀的博士学位论文。答辩过程中叙述清楚，回答问题正确。经答辩委员会无记名投票表决，一致同意通过论文答辩，并建议授予陈冲同学工学博士学位。