中国科学技术信息研究所

硕士学位论文

推荐系统中的隐语义模型算法研究

Research on Algorithm of Latent Factor Model in Recommender System

作者　江雪琴

导师　张志平

中国科学技术信息研究所

论文提交日期（2015年09月）

中图分类号 TP391/G35　　学校代码 80901
UDC 004　　密　　级 公开

中国科学技术信息研究所

硕士学位论文

推荐系统中的隐语义模型算法研究

Research on Algorithm of Latent Factor Model in Recommender System

作者姓名 江雪琴　　学　　号 809011311
导师姓名 张志平　　职　　称 研究员
学位类别 管理学　　学位级别 硕　士
学科专业 情报学　　研究方向 知识管理与技术

中国科学技术信息研究所

论文提交日期（2015 年 09 月）

独创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，论文中除已经加以标注和致谢的地方外，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，均已在文中明确说明并表示了谢意。

研究生签名：江雪琴　　时间：2015 年 11 月 2 日

关于论文使用授权的说明

本人完全了解中国科学技术信息研究所有关保留、使用学位论文的规定，即：所里有权保留送交论文的打印稿和电子稿，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。同意中国科学技术信息研究所用不同方式在不同媒体上发表、公布论文的全部或部分内容。保密的论文在解密后遵守此规定。

研究生签名：江雪琴　　时间：2015 年 11 月 2日

导师签名：　　时间：2015 年 11 月 2 日

致　谢

光阴似箭，岁月如歌，转瞬间两年半的研究生学习生涯即将步入尾声。回首2013年的那个夏天，怀揣着求学梦的我初次来到这偌大的北京城，无论是它的繁荣，还是喧嚣都给我留下了深刻的印象，我对接下来两年半的北京生活充满好奇和期待。

中国科学技术信息研究所就是我完成研究生学业的地方。在这里，有着治学严谨、科研实力雄厚的师资队伍；轻松活跃、安静舒适的学习环境；设备先进、配套齐全的教学设施……虽然，两年半的研究生学习即将结束，但在中信所的点点滴滴，都令我受益颇多。

毕业论文设计，不仅是对我们科研实力的考验，更是对我们学习生活的总结。历经大半年时间的努力和磨炼，硕士毕业论文终于完稿。回首这期间，从选题、资料收集、整理、思索、停滞、修改直至最终定稿，我得到了许多关怀和帮助，在此，我要向他们表达我最诚挚的谢意。

首先，我要衷心感谢我的导师张志平老师。张老师为人谦和、平易近人、治学严谨，是一位难得的好导师。我的论文是在张老师的悉心指导下完成的，从论文定题到写作定稿，倾注了张老师大量的心血。在我攻读研究生期间，张老师对我学习的关心、爱护和谆谆教诲，都令我深深受益。作为我的导师，他为我指点迷津，让人如沐春风；作为我的长辈，他对我关怀备至，让人感念至深。我非常庆幸能够成为张老师的学生，是他让我在研究生学习期间，学会了如何做好科研，如何学好专业知识。在此，我谨向张老师表达我最诚挚的谢意！

其次，我要衷心感谢李琳娜老师。在论文写作过程中，每当我有疑问，李老师都会放下手中繁忙的工作，耐心地指导我。当我的论文实验受阻时，李老师会帮助我找到问题的源头，分析实验设计可能存在的问题等。在我论文初稿完成之后，李老师又会在百忙之中对我的论文进行细心地批注和修改，提出许多中肯的建议，使我能在写作中把握住研究主题和核心内容，完成一篇真正合格的硕士毕业论文。李老师不仅在论文设计中提供了我很大帮助，而且她还关心我平时的学习、生活和工作。她在我的心中，就像一位姐姐，为我提供丰富的学习资料，帮助我增长专业知识；和我谈人生、谈工作，为我将来就业指点迷津，开拓思路，促进我成长；让我能够时刻审视自己，追求适合自己的未来。

再次，我要衷心感谢所有教导过我、关心过我的老师。特别是教研室的张运良老师，他带领我参加一些学术会议，聆听学术报告，让我能够接触新的科学知识，了解学术动态提高学习积极性，使我的学习生活更加丰富和充实。另外，还要感谢研究生部的全体老师，是你们为我们营造了一个轻松优越的学习环境，让我们能够在中信所踏实安心地学习和工作。在毕业设计的整个阶段，研究生部的罗勇老师时刻提醒我们安排好工作计划，在开题、中期直至最终答辩，都悉心地给予我们专业地指导，让我们能够按时、按质、按量地完成论文。生活中，张泽玉老师、王桂凤老师和刘敏老师等，时刻关心我们的生活状况，积极宣传宿舍安全意识，为我们能够拥有一个安全舒适的生活环境付出了心血，衷心地感谢你们。

最后，我要衷心感谢一直关心和支持我的同学、朋友，感谢你们给予我的帮助和鼓励，能够与你们相伴，共同学习，共同进步，你们的情谊我将终生难忘。我还要感谢我的家人，你们的支持是我坚强的后盾。当我面临困难而迷茫之际，是你们为我排忧解难，你们无私的爱是我不断前行的动力。

感谢我生活学习过的母校——中国科学技术信息研究所，您给了我一个宽阔的学习平台，让我不断吸取新知识，不断成长。很高兴在最美好的时光与您相遇，这段旅程虽有坎坷和挫折，但收获颇丰，这里的点点滴滴我将永记于心。

在论文的最后，我还要衷心感谢各位评审专家，能够在百忙之中抽出宝贵的时间审阅我的论文，感谢你们辛勤地付出，以及各位评委老师的赐教和指正。谢谢！

推荐系统中的隐语义模型算法研究

摘　要

推荐系统是一种考虑用户差异，为用户提供个性化服务的信息过滤技术，能够很好地缓解信息过载问题所带来的挑战。推荐系统的个性化服务方式，不仅促进了电子商务领域的快速发展，同时也对传统的数字图书文献领域产生了重大影响。

本文深入地研究了协同过滤推荐技术中隐语义模型的相关内容，主要包括：

(1)为明确本文的研究内容和研究思路，比较分析了推荐系统领域常见推荐技术的优缺点，全面介绍了隐语义模型的发展背景和研究现状，为后期研究工作的展开奠定基础。

(2)详细阐述了隐语义模型的工作原理、学习算法和评测指标，介绍了基本的隐语义模型Base-SVD、引入偏置项的隐语义模型Bias-SVD、考虑隐式反馈的隐语义模型SVD++和融合邻域的隐语义模型Asymmetric-SVD等算法，并在MovieLens数据集上比较分析了几种模型的推荐效果，以及受不同参数的影响情况，总结各个算法的特点。结果表明：模型Asymmetric-SVD受隐特征维度、学习速率等参数的影响最大，综合评分预测效果要优于其他模型。

(3)详细讨论了隐语义模型在科技文献推荐领域的应用，提出一种融合异构信息网络和隐语义模型的文献推荐算法。主要考虑用户对文献的隐式评分，以元路径的方式挖掘文献间的近邻关系，从语义关联的角度表达用户对文献的偏好程度，融合隐语义模型预测用户评分，实现推荐。该算法能综合多维度信息，有效缓解评分数据稀疏性问题。

(4)对于元路径权值分配问题，文中利用二分类和贝叶斯优化排序两种学习算法进行求解。并在CiteULike-a数据集上反复实验，分析不同算法的运行结果，验证文献推荐模型的可操作性和有效性，最后得出基于贝叶斯优化排序算法实现的推荐模型综合评分预测效果较好。

全文共图15幅，表7个，参考文献65篇，其中英文参考文献48篇。

关键词：推荐系统；隐语义模型；评分矩阵；异构信息网络；元路径

分类号：TP391；G35

Research on Algorithm of Latent Factor Model in Recommender System

Abstract

Recommendation system is a kind of information filtering technology. which considers the difference between users and provides users with personalized services, to ease the challenges of information overload. The personalized service mode of recommendation system, not only promotes the rapid development of electronic commerce, but also has a great influence on the traditional field of digital libraries.

In this article, we study the latent factor model of collaborative filtering and some related contents, the main work is as follows:

Firstly, in order to make clear the research contents and ideas in this article, the common recommender technologies' advantages and disadvantages are comparative analyzed. Besides, the development background and research status of LFM are also introduced to lay the foundation of the later work.

Secondly, the working principle, learning algorithms and evaluation metrics of LFM are analyzed in detail, and do a brief introduction for Base-SVD, Bias-SVD which includes bias item, SVD++ which considers implicit feedback of users and Asymmetric-SVD which is a integration model of combining LFM with neighborhood collaborative filtering model. Further more, the effect of different algorithms influenced by different model parameters, as well as the performance of different algorithms are compared through experiments on Movielens dataset. The results show that model parameters, like latent factors and learning rate, have biggest impact on Asymmetric-SVD, and this model can get better effect of rating prediction.

Thirdly, an optimized algorithm which integrates the heterogeneous information network and LFM is proposed, aiming at the discussion on application of LFM in paper recommendation system. This algorithm mainly considers the implicit feedback of users, then neighbor relation between papers is mined through meta-path to express users' preference

for papers, and users' rating matrix is achieved by LFM. The proposed algorithm can effectively alleviate the problem of data sparsity, which can synthesize multi dimension information.

Finally, two learning algorithms, Classification and Bayesian Ranking Optimization, are used to implement the paper recommendation model. Besides, the results of different algorithms are analyzed, the operability and validity of the model are verified through several experiments on CiteULike-a dataset, As a result, the paper recommendation algorithm based on Bayesian Ranking Optimization is better than others.

Key words: Recommendation system; Latent factor model; Rating matrix; Heterogeneous information network; Meta-path

目 录

图目录

表目录

引 言

随着互联网技术的迅速发展，爆炸式增长的信息资源同时呈现在我们面前。例如，Netflix 上有数万部电影，Amazon 上有数百万本的图书，Del.icio.us 上有超过数十亿级的网页收藏等，如此庞大的信息量，只通过简单地浏览或是查找获取自身需要的那部分信息是非常困难的。传统的信息检索方式，只能呈献给所有用户相同的排序结果集，无法针对用户的兴趣偏好提供相应的信息服务，对于每个用户来说，准确性并不高。而个性化推荐技术的提出，却能考虑用户的特征，提高用户获取信息的准确性和自身满意度。

自 20 世纪 90 年代个性化推荐研究概念被独立提出之后，受到各界广泛关注，并得到迅猛发展，尤其是在电子商务领域取得良好效益。面对海量的商品信息，用户往往难以发现最喜欢或最适合的商品，电子商务系统产生的海量交易数据，能够从中挖掘有用的信息使得交易更高效是非常有意义的。个性化推荐技术就是针对以上问题和需求产生的。个性化推荐技术不仅减少了用户查找信息所消耗的时间，同时也提高了获取信息的准确性。

此外，数字图书馆拥有丰富的图书文献资源，用户能够从中获取满足自身需求的文献。传统的文献获取方式，用户需通过关键词进行检索，这一过程需要花费大量的时间和精力。而个性化推荐技术为数字图书馆实现个性化的文献服务提供了可能。数字图书馆个性化文献推荐技术，面向的是个体用户，主要为其提供个人感兴趣的文献列表，缩短了数字图书馆内容和个人信息需求之间的差距，已经成为数字图书馆技术发展的重要组成部分和前沿课题。

1 绪论

1.1 选题背景

从信息检索的角度来看，在互联网刚刚起步阶段，以 Yahoo 为代表的门户网站，通过建立网页资源的导航功能，让用户能够利用导航浏览网络上的信息，用户几乎不需要描述自己的检索需求；随着检索技术的不断进步和信息资源的爆炸式增长，导航网站已经难以满足用户的信息需求。以 Google 为代表的搜索引擎行业的出现，为用户提供了便捷的信息检索方式，用户只需要输入关键词，搜索引擎就会根据关键词与文档的匹配返回相应的结果。但这种检索方式并不能完全区分用户自身的需求差异，对于相同的输入，返回的结果也可能是相同的。然而，个性化网页搜索、定制化服务等个性化检索方式是传统搜索引擎技术的进步，它将用户的行为记录作为检索结果生成的影响因素，以便满足用户的需求和提高用户对检索结果的满意度[1,2]。同时，电子商务领域的出现，对用户提供定制化服务的需求越来越大。而个性化推荐系统，它以用户的历史行为记录作为用户特征发现的重要资源，实现用户的个性化信息推荐，促进了电子商务领域的快速发展。

推荐系统是一种根据用户的偏好特征和购买行为，对繁杂的资源进行有效过滤，向用户提供其感兴趣的物品或信息的个性化服务方式，它很好地改善了信息过载问题所带来的挑战。对于个性化推荐系统来说，向用户提供多少数量的物品或信息并不是最重要的，重要的是推荐的物品或信息是否为用户所满意的[3]。推荐系统的个性化服务方式，不仅成功地为电子商务领域创造了巨大效益，同时也对传统的数字图书文献检索产生了重大影响。例如文献推荐系统，它相比传统的文献检索来说，用户和检索系统的主动权发生了转变，文献检索是用户主动向检索系统提出请求，系统予以反馈；而文献推荐则是检索系统根据用户的搜索、点击、浏览等历史行为主动为用户提供服务。

Netflix 是美国一家提供在线电影租赁服务的公司，用户只需按月缴纳一定的租赁费用就可以向 Netflix 订阅大量的电影。在收到客户的订阅需求后，Netflix 会通过电子邮件（电子版）或投递（DVD 碟片）的方式为客户提供其所需的电影，用户还可以为 Netflix 上的电影打分（包括 $1, 2, \cdots, 5$），分数越高表示用户对相应电影的评价越高。2006 年 10 月，Netflix 对外发布了一个电影评分数据集，并建立了 Netflix

Prize 竞赛。Netflix Prize 竞赛的主要目标是提高推荐系统的推荐效果，其推荐效果需在其原有推荐系统（Cinematch）的基础上改进 10%（使用均方根预测误差 RMSE 来衡量）[4-6]。第一个达到要求推荐精度的参赛团队将被授予 Grand Prize，奖金为 100 万美元。2009 年 9 月 10 日，Netflix Prize 官方宣布了竞赛的最终胜利者，整个竞赛正式结束。

Netflix Prize 竞赛过程中涌现出很多优秀的推荐算法，其中隐语义模型（latent factor model，简称 LFM）最受关注，逐渐成为推荐系统领域最热门的名词，它无论在单模型推荐还是组合模型推荐中都取得了较好的推荐效果，引起了国内外学者的研究兴趣，比如部分研究人员致力于对隐语义模型算法的改进，希望能够提高模型的推荐效果等。但是，就隐语义模型的发展现状来看，国内缺少对该模型的系统理论和应用等研究。因此，本文将结合中国科学技术信息研究所科研项目预研基金“基于海量科技文献的异构学术网络挖掘研究”，详细介绍隐语义模型的工作原理和典型算法，同时利用科技文献的特有属性，提出适用于图书文献领域的隐语义模型推荐算法。

1.2 研究意义

个性化推荐技术能够有效解决信息过载问题，是一个集信息检索、人机交互、数据挖掘和用户建模等多学科交叉发展的领域，多年来已经在研究上取得了丰富成果，特别是在电子商务领域取得了良好效果。此外，数字图书馆个性化文献推荐技术能够面向个体用户，提供符合其个人偏好的文献，缩短图书馆内容和个人信息需求之间的差距，已经成为数字图书馆技术发展的重要组成部分和前沿课题。针对本文选择的课题，主要有三点意义：

(1)个性化推荐系统的最大优点在于考虑用户的偏好差异，主动为用户推送服务，这不仅减少了用户查找信息的时间，同时也提升了信息生产者或提供者的效益。对文献推荐来说，个性化的文献推荐服务能帮助科研人员快速从海量文献中发现其感兴趣的文献，这在很大程度上节省了科研人员的文献检索时间，同时也能及时获取新发表的高质量的科研文献。

(2)自 2006 年 10 月 Netflix Prize 竞赛举办以来，隐语义模型成为推荐系统领域最热门的研究话题，并且根据参加 Netflix Prize 竞赛团队的提交经验，隐语义模型的单独应用或是与其他算法的组合，都可以有效地改进推荐模型的预测结果精度。由此，选择隐语义模型作为研究主题，无论是在理解该算法的思想、原理，还是在算法的改进和

应用推广方面，对掌握推荐系统关键技术和了解领域发展动态都是非常有帮助的。

(3)目前，隐语义模型一般以用户的显式行为信息为主要研究对象，缺少对隐式反馈行为和针对文献特征建立推荐模型的相关研究。所以，综合考虑文献特有属性和用户的隐式反馈行为，对隐语义模型在文献推荐中的应用问题进行具体地分析和探讨，这对文献推荐领域技术发展是非常有意义的。

1.3 研究内容

根据文章选择的研究主题，以及对现有研究成果的调研，现将主要工作内容分为以下三个方面：

(1)研究隐语义模型在推荐系统中的实现机制，分别就隐语义模型中几种代表性的改进算法，详细描述其核心思想、关键技术和实现过程，并在真实电影评分数据集 MovieLens 上经多次反复实验，比较分析不同参数对模型预测效果的影响和不同算法的优缺点。

(2)针对用户-文献显式评分难获取和数据稀疏性问题，提出依据用户的隐式反馈行为，从构建异构信息网络的角度，采用元路径的方式挖掘文献间潜在的语义关联，重新描述用户-文献间的交互行为。然后，融合隐语义模型，得到用户、文献在不同元路径下的潜在因子特征分布，以线性组合方式实现多维信息下用户对文献的评分预测模型。

(3)针对异构信息网络下文献推荐模型的元路径权值分配问题，提出采用二分类模型和贝叶斯优化排序两种参数学习算法进行求解。为了验证算法的可实施性和有效性，分别将不同模型在 CiteULike-a 数据集上进行实验，进一步比较分析实验结果。

1.4 结构安排

本文以推荐系统中的隐语义模型算法研究为核心，从原理、模型建立和实验等方面做详细论述，基本框架如图 1.1 所示。

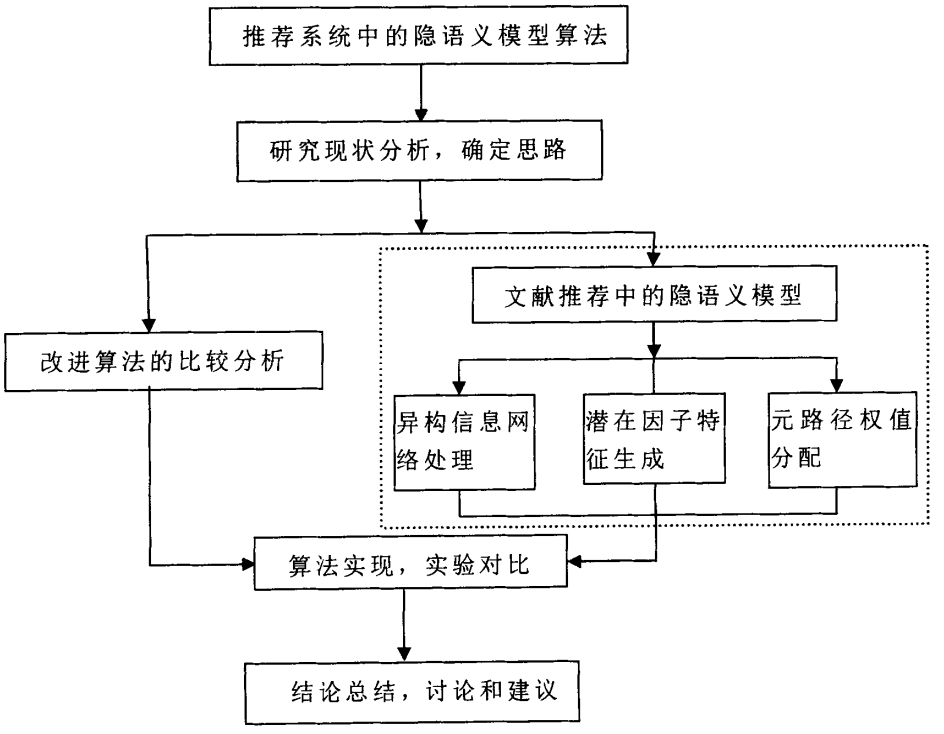


图 1.1 本文主要内容框架

论文总共分为五章，具体安排如下：

第一章 绪论。分别介绍个性化推荐系统的发展、隐语义模型的研究背景，阐述推荐系统中隐语义模型算法研究的重要意义。

第二章 相关领域研究现状。简单介绍推荐系统的概念、常见个性化推荐技术的优缺点，一般文献推荐领域算法的发展概况等。针对隐语义模型的国内外研究内容，分析和讨论其存在的缺陷，以及国内相关研究的不足。

第三章 隐语义模型典型改进算法与实验比较。从理论层面，论述隐语义模型中应用最广泛的奇异值分解、推荐模型结果评价常用指标和模型学习算法；从方法层面，重点介绍隐语义模型的几种典型改进算法，并以真实的公开评分数据集作为实验对象，分析和讨论模型中涉及的参数特征对模型的影响，以及比较几种模型的推荐效果。

第四章 基于隐语义模型的文献推荐算法。该部分主要针对文献推

荐的特点，从异构信息处理和考虑文献语义关系的角度，建立适用于文献推荐的隐语义模型，并从分类和排序的角度对模型参数进行求解，以真实文献数据集的实验结果为基础，讨论和分析隐语义模型在文献推荐中的应用特点。

第五章 结论与展望。主要是对论文内容进行总结、说明创新点、讨论存在的不足之处，以及对后期研究工作提出设想和建议。

2 相关领域研究现状

2.1 推荐系统领域相关研究

2.1.1 推荐系统介绍

自 20 世纪 90 年代，推荐系统被独立提出之后，逐渐成为国内外重要的研究方向。推荐系统是建立在海量数据挖掘基础上的一种信息服务平台，它根据用户的兴趣特征和购买行为，向用户推荐其可能感兴趣的物品或信息（英文中称为 item），帮助用户处理因信息过载所带来的问题，为其提供个性化的决策支持和信息服务。

一个完整的推荐系统主要包括用户信息的行为记录模块、分析用户喜好的模型分析模块和推荐算法模块。行为记录模块负责用户的偏好行为，如评分、下载、购买、浏览等；分析模块主要是对用户的行为信息进行分析，建立合适的模型描述用户的兴趣特征；推荐算法模块是推荐系统的核心，通过后台的算法实现，为用户筛选出其感兴趣的物品或信息，完成个性化推荐[7]。

目前，实现个性化推荐的算法主要分为基于内容的推荐（Content-Based Filtering，CBF）、协同过滤推荐（Collaborative Filtering，CF）和混合推荐（Hybrid-Recommendation，HR），其中，协同过滤推荐方法应用最为广泛。

(1)基于内容的推荐方法

该方法不需要依赖用户对物品的评价，而是利用用户或物品的属性特征计算他们之间的相关程度，进而将匹配度高的数个物品推荐给相应的用户。这个过程主要有两个缺陷：一是选择能够代表用户或物品的特征，这在实际问题中往往非常困难，需要获取和记录用户或物品大量的信息资源，才能发现用户或物品的特征；二是基于内容的推荐一般是依据用户已经选择的物品内容信息，计算物品间的相似性从而实现推荐，推荐的物品与用户已经消费的物品可能很相似。因此，这并不利于发现用户潜在感兴趣的不同类别的物品，缺乏新颖性[7]。

(2)协同过滤的方法

该方法不需要事先获取用户和物品的属性特征，而是依赖于所有用户的历史行为（通常以评分的形式），进而构建评分矩阵、计算行为相似度、预测和推荐，最大限度地找到目标用户所感兴趣的物品[3]。协同过滤推荐可以进一步被分为基于邻居的协同过滤和基于模型的协

同过滤，典型的算法有邻居模型（Neighborhood Approach）和隐语义模型（Latent Factor Model）[8]。协同过滤推荐克服了基于内容推荐方法的缺陷，但也面临着新的挑战，如冷启动问题，对于一个新用户，由于缺少他（她）对物品的评分，系统无法为其提供可靠的推荐结果。对于一种新物品，类似的问题也同样存在，系统很难判断将其推荐给哪些用户。

(3)混合推荐方法

该方法采用多种策略将基于内容的推荐和协同过滤推荐融合在一起，这样既能克服两种方法各自的缺点，又能发挥两者特有的优势。通常采用的组合策略有加权组合、混合组合（后处理）和序贯组合三种。其中，加权组合应用最为广泛，主要表现为线性加权模型，它对不同推荐系统算法赋予一定的权值，经线性组合不同因素，同时训练得到推荐结果。而混合组合则是对独立推荐算法结果的后处理，选择好的预测结果形成最终的推荐列表。序贯组合相比上述两种方法较为复杂，需根据优先级先后次序对独立算法进行处理，并且组合的算法要满足特定的标准。

2.1.2 文献推荐领域研究现状

文献推荐系统是推荐系统的一种，它通过分析用户与文献之间的二元关系，利用已有的选择过程或相似性挖掘用户感兴趣的文献，进而为用户提供文献服务。文献推荐系统有别于传统的信息检索，它通过分析用户的历史行为，包括查询、浏览、点击、下载、打分等，建立属于用户自己的特征模型，主动为用户提供文献服务，并不需要用户将大量时间花费在检索词的选择、检索式的构建和检索策略等方面，大大提高了文献获取效率和文献利用率。

一般的，电子商务领域的推荐算法同样适用于文献推荐领域，但在具体的技术实现上又存在差异，因为文献有标题、关键词、引文等特有属性可以利用。对于学术文献的推荐技术，可以大致分为以下几类：

(1)基于网络结构的推荐技术。如刘韵毅等[9]提出模糊联想记忆神经网络，建立偏好评价模型，根据用户的偏好对搜索引擎的检索结果进行评级，再为用户推荐评级较高的文献。Massa 等[10]受社会关系网络的启发，提出一种基于信任感知的协同过滤推荐技术，推荐过程主要受用户声誉的影响，声誉由传播信任关系的图结构计算得到。Sun 等[11]针对异构信息网络问题，提出一种基于元路径的相似性测度 PathSim，寻找网络中的对等节点，通过特征建立用户和文献间的关系，为推荐模型提供有用的行为信息。姚远[12]根据作者和论文信息组合形

成异构的论文引用网络，通过提取作者、研究领域和文本相似度等信息作为推荐依据，利用余弦相似度计算文献间的相似度值，进而采用重启动随机游走算法进行推荐，并取得较好的推荐效果。一般的，网络结构是发现用户和物品间潜在关系的重要方法，能够对用户与物品间的关系进行很好的描述，所以在推荐算法设计中比较常用。

(2)基于数据挖掘的推荐技术。如 Chen 等[13]考虑到文献推荐领域用户空间相对庞大，而文献空间相对较小的特点，借鉴数据挖掘中的聚类思想，首先使用蚁群算法对用户进行聚类，然后在每一个簇内使用 Apriori 算法发现频繁模式，进一步发现关联规则，完成相关推荐。陈祖琴等[14]利用数据挖掘中的关联规则挖掘分析论文数据库，提出适用于中文引文分析的混合加权关联规则挖掘算法，采用 PageRank 算法确定权重，获得一些有意义的结果。由目前推荐技术的发展来看，数据挖掘技术扮演着非常重要的角色，也为文献推荐系统的实现提供了非常重要的技术支持。

(3)基于本体的推荐技术。基于本体的文献推荐，从概念结构层面对用户的行为特征进行描述，能够有效揭示用户对某类或某种主题文献的兴趣偏好。应用比较成功的研究成果也有一些，如 Middleton 等[15]提出利用本体方法构建用户偏好的推荐方法，设计了两个能够进行实时文献推荐的实验系统 Quickstep 和 foxtrot，根据检测到的用户浏览行为和用户的反馈信息构建用户偏好。Liao 等[16-18]设计的文献推荐系统 PORE，将中图分类法作为图书馆本体，首先根据用户的借书记录计算用户对每个类别的喜欢程度，将大于预先设定的阈值的类别保留，重新组织成用户个性化本体，然后计算用户对每个类别的关键词的兴趣度值，最后根据计算的关键词兴趣度值计算用户对每本书的偏好值，从而完成个性化推荐过程。

(4)基于向量空间的推荐技术。如徐勇等[19]针对科技文献特征词在语义上的层次特性，指出基于概念泛化的内容过滤推荐算法，用矢量空间模型作为用户兴趣偏好和科技文献特征的描述模型。黄泽明[20]采用基于主题模型的方法，在文档中单词分布已知的情况下，计算每篇文档主题分布的后验概率，挖掘潜在主题与结构。它不再单独考虑文档在词典空间上的维度，而是引入主题空间，实现文档在主题空间上的表示，不仅可以捕捉文档内的语义信息，而且有利于发现文档间潜在的联系，给予用户和论文推荐很好的解释性。向量空间技术主要来源于文献检索领域，通过关键词与文档的向量模型揭示文献特征，而在推荐领域，则是表达用户行为特征的有效方式。

(5)基于文献计量的推荐技术。文献计量学中的理论，为文献推荐

算法的设计提供了思路。比如文献特有的引用关系，它在一定程度上反映了用户对被引文献的兴趣，这就是一个很好的用户行为信息。如Goodrum[21]介绍了 citeseer 系统利用文献间的引用关系发现相关文献的方法。McNee 等[22]通过离线实验和在线实验分析了四个基于引文网络和协同过滤推荐技术向一篇目标文献推荐相关参考文献的算法。陈祖琴等[14,23]主要研究论文引文数据在文献推荐中的作用，提出混合加权关联规则挖掘算法，实验演示了引文的关联规则挖掘在相关文献推荐中的应用，并取得一定效果。

(6)其他推荐技术。除了上述描述的几种推荐技术外，还有许多不同的研究见解。如 Gipp 等[24]实现了文献推荐系统 Scienstein，该系统是一个集成引文分析、作者分析、源分析、隐式打分、显式打分等多方面信息的混合推荐系统。Basu 等[25]针对如何将需要评审的很多会议论文提交给相应评审人的问题，提出了将评审人的多种信息及论文的多种信息融合的推荐技术。尉萌[26]提出一种基于用户搜索行为演化模式的文献推荐方法（CALL），从文献库与检索日志中提取文献、读者与检索日志特征；将文献分为几个阅读阶段，用最长公共子序列算法从三个特征中寻找到文献阅读序列，并将超过一定长度与频率的文献序列作为推荐结果等。

从上述的文献推荐技术来看，与一般电子商务领域的个性化推荐技术相比，既有共同点，又存在差异。比如两者的推荐目标相同，都希望为用户提供其感兴趣和满意的物品或信息，只不过一种以文本形式为主，一种以图像、音频、视频等形式为主；电子商务领域，用户对商品的打分信息可以是显式的(如 1-5 分评价)，也可以是隐式的(如浏览时间的长短)，在文献推荐领域也同样如此，用户可以明确给出对相应文献的评分，也可以根据其浏览行为（如是否下载了文献全文等）判断用户对文献的评分。

一般的个性化推荐技术，可能更多关注的是像评分、下载、点击、评论、标签等这样的用户行为信息。而在文献推荐领域，这样的行为信息比较难获取，用户主动赋予评论、打分等情况就更少。但每篇文献有标题、摘要、关键词、正文等内容信息可以利用[27]。所以，对于不同推荐系统，采用的推荐技术应综合考虑对象特征，合理选择和准确应用。

2.2 隐语义模型的国内外研究现状

隐语义模型（Latent Factor Model，简称 LFM）是一种有效的隐含

语义分析技术，其核心思想是通过潜在特征联系用户和物品。其过程分为三个部分：将物品映射到潜在分类、确定用户对潜在分类的兴趣度、在用户兴趣度高的分类中选择物品推荐给用户[28]。

一般地，在推荐系统中，用户对物品的行为信息往往是以评分矩阵的形式表示，无论在电子商务领域还是在文献推荐领域，由于物品数量规模巨大，所表示的评分矩阵通常具有很高的维度，从而造成计算上的困难。隐语义模型则将这一高维评分矩阵降到低维度的子空间，使用户和物品的潜在语义关系在子空间中自然显现，从而能很好地描述用户的兴趣偏好，在推荐结果预测方面体现出很高的准确性和稳定性[29]。相类似的模型有 PLSA[30]、LSI[31]、LDA[32]等。

隐语义模型有别于传统的协同过滤推荐技术，它不再拘泥于单方面分析用户之间、物品之间的关系，而是通过挖掘用户和物品之间的潜在语义关系以实现推荐，无论是在单模型还是组合模型的推荐中，都具有较好的预测效果。

2.2.1 国外研究现状

自 2006 年 10 月 Netflix Prize 竞赛举办以来，对隐语义模型的研究一直是推荐系统领域的热门话题。目前，研究界对隐语义模型的研究主要集中在以下四个方面：

(1)奇异值分解

隐语义模型实现的基础是矩阵分解（Matrix Factorization，简称 MF），它的输入是 m 个用户对 n 个物品的评分数据所构成的 m×n 的评分矩阵。矩阵分解可以对输入的评分矩阵做降维处理，将用户和物品映射到 f 维的因子空间，建立用户和物品间的联系，以实现隐语义模型在推荐系统中的应用[33]。

矩阵分解是一种重要的机器学习技术，其实现方法有很多。其中，奇异值分解（Singular Value Decomposition，简称 SVD）最为常用，它最早被应用于信息检索领域的潜在语义模型（Latent Semantic Model）[30]，之后又被协同过滤推荐方法所采用[34-36]。2006 年 Netflix 举办推荐算法竞赛，Simon Fun 提出了 Funk-SVD 算法，后来 Koren[4-6, 8, 37]将其称为 Latent Factor Model，即隐语义模型，其实质就是奇异值分解。竞赛之后，又出现了 RSVD、SVD_KNN、SVD_KRR、NSVD、SVD++、timeSVD 等改进的隐语义模型。

隐语义模型在进行奇异值分解时，需要将稀疏的评分矩阵转为稠密矩阵，但是由于受高维数据的影响，奇异值分解的时间、空间复杂度都很高，所以在大数据集上无法直接使用。因此，对于隐语义模型的实现，经常采用优化目标函数（如最小化均方根误差、平均绝对误

差等）的方式学习模型中的参数，其学习算法主要包括两种：梯度下降法（Gradient Descent,SGD）和交替最小二乘法（Alternating Least Squares，ALS）。

梯度下降法是优化理论中最基础的优化算法，它首先通过求参数的偏导数找到最快下降方向，然后通过迭代法找到优化参数。Simon Funk 和 Koren[8,37]采用梯度下降法对训练数据集进行计算，在算法的运行时间性能方面取得了很大进步。

交替最小二乘法的优化思想与梯度下降法相类似，它通过交替固定因子变量（用户和物品）的方式，每次只优化一个变量，反复进行，直到找到最小误差值。

虽然，梯度下降法计算效率比较高，但至少在两种情况下，交替最小二乘法 ALS 显得更为有效。一是更容易并行化处理问题，因为 ALS 对于每一个物品参数和每一用户参数都是独立计算的，并不影响其他参数的计算，这就为大量的并行处理提供了可能[38]；二是方便处理隐式数据（implicit data），因为数据集不能总被认为是稀疏的显式行为，许多情况下往往以隐式行为存在，如果按照梯度下降法循环处理单个训练样例，这是不可行的，而 ALS 则可以很好地处理这种情况[39]。

(2)评分数据处理

评分数据处理主要是从数据本身存在的问题出发，通过一些方法减少或避免因数据本身存在的偏差对预测结果产生影响。

Koren 等从用户的评分倾向、物品类型、评分时间间隔等方面考虑评分数据的预处理，提出基准预测和全局作用两种方法，将其与隐语义模型相结合，构建引入偏置项的隐语义模型[6,40]。同时，Koren 等还充分考虑了时间效应在评分数据上的特点，一方面是对物品的处理，主要是对时间采用切片的方式获取物品在不同时期的评分差异；另一方面是对用户的处理，Koren 采用时间偏离度这一指标，计算用户评分时间偏离平均评分日期的值来衡量用户的评分时间效应[5,41]。时间效应仅仅考虑的是用户在时间维度上表现出的评分差异，至于时间效应的其他描述方式以及像地理信息等多元上下文情境信息的应用，都有可能成为隐语义模型及其他推荐模型今后的研究方向。

此外，协同过滤推荐依靠最多的数据是用户对物品的显式评分，隐语义模型在显式反馈数据上解决评分预测问题达到了很好的精度。但是，显式反馈数据并不总是能够被获取的，而且仅依靠显式评分描述用户偏好特征会存在一定的偏差。因此，考虑隐式反馈数据（即负样本数据）是隐语义模型评分处理的另一个研究内容。

在隐式反馈数据集上应用隐语义模型解决 Top-N 推荐的第一个关键问题是如何给用户生成负样本。对负样本的采样，应该保证每个用户正负样本数的平衡。Koren[8, 33]发现 Netflix 评分数据集不仅给出了具体的评分数值，而且还告诉我们哪些电影被赋予了评分，哪些没有。忽略用户是怎样赋予评分的这一因素，将评分矩阵转成二值矩阵，“1”代表被评分的情况，“0”代表未被评分的情况。当然，这样二值矩阵并不能完全代表隐式反馈信息，但通过实验发现融合该二值矩阵的算法模型提高了评分预测结果的准确性。

Rendle[42]等提出贝叶斯排序模型（Bayesian Personalized Ranking，简称 BPR），实现对隐式反馈二值矩阵表达方式的改进。该模型将观察到的用户行为矩阵转为低阶稠密矩阵，基于贝叶斯后验估计量为优化标准、采用梯度下降法训练模型；Mnih[43]运用此种模型，加入物品的分类信息（Taxonomy Information），对物品的低阶关系矩阵做补充，实现隐语义模型在隐式反馈数据中的应用。Lei 等[44]充分考虑用户或物品间的社会情境关系，生成隐式反馈数据，作为用户潜在因子变量的补充；Yao 等[45]提出将用户的隐式行为通过时间、地点等上下文情境信息表示成图模型，然后运用随机游走等算法建立用户和物品间的兴趣模型等。这些都是以用户的隐式行为为基础所做的研究成果。

综上所述，隐式反馈行为的表示方法还不是很多，主要以布尔模型、BPR 模型的形式体现，许多研究成果以此为基础，通过加入一些附属信息，提高隐式反馈矩阵的准确性。由此，如何准确描述和表示用户的隐式反馈行为，也许是隐语义模型等协同过滤推荐算法将来可以突破的地方。

(3)附属信息的应用

协同过滤推荐算法最大的优点是不需要知道物品本身的特征，就能为用户提供推荐列表。但是冷启动问题一直是协同过滤推荐算法的核心问题，新的物品可能因为没有得到用户的反馈，而在很长时间内都得不到推荐；新的用户可能因为缺少历史行为记录，导致系统不知道应该推荐什么样的物品给用户。

处理冷启动问题的常见方法主要是融合物品或用户的内容信息，实现基于内容和协同过滤的混合推荐。例如，Melville[46]等提出一种混合推荐的方法，首先通过基于内容的推荐预测评分，对用户评分矩阵进行填补，然后运用协同过滤算法对填补后的用户评分矩阵进行处理，最终形成推荐结果；Forbes[47]等将用户的特征表示成向量形式，具有的特征对应的值即为“1”，不具有的特征对应的值即为“0”，然后将该向量作为因子式和隐语义模型融合，并通过实验证明了其算法对

内容特征描述的准确性和可操作性。

物品在推荐系统中拥有多种属性特征，例如商标、价格、商家、文本说明等，这些属性包含了物品丰富的特征信息。用户也是一样，性别、年龄、爱好、地址等人口统计信息，足以表征用户的特点。另外，用户和物品的其他附属信息，如分类、标签、评论等，也能在一定程度上描述物品的特征和用户的兴趣。

因此，用户或物品的附属信息足以对推荐系统预测结果产生影响。例如，在隐语义模型研究中，就其输入数据类型的单一性，Ahmed[48]等提出LFUM的新方法，集成了基于物品属性特征模型和LFM的优点，对物品属性特征运用个性化的贝叶斯层级模型学习用户的兴趣偏好，数据的稀疏性处理则采用了基于物品分类信息（taxonomy）的向前过滤和向后平滑的方法，该方法能够同时处理固有属性和变化属性问题，最后采用 BPR 排序算法实现用户偏好矩阵和 LFM 的混合推荐。Zhan Chenyi[49] 等为了提高学术文献推荐结果的准确性，提出内容+属性的隐语义模型，通过主题模型 LDA 提取文献的主题分布，融合属性特征，最终实现隐语义模型在异构学术网络推荐中的应用。

综上所述，在隐语义模型中加入附属信息，对准确挖掘用户的兴趣偏好是很有意义的，而且有利于提高推荐结果的精度。目前，融合用户的人口统计信息、物品的属性描述、标签、分类、评论等的推荐方法已有一些。通过比较可以发现，他们大多只是融合了某一种信息，并未综合考虑多种因素。因此，多维度信息的综合处理对推荐模型性能的影响可能会是促进推荐技术发展的重要研究内容。

(4)模型的组合策略

Netflix Prize 竞赛中，Bell[5]等提出的隐语义模型与 KNN 邻居模型的组合推荐取得了较好的预测结果。邻居模型的核心是计算用户与用户间或物品与物品间的相关度，不需要考虑用户-物品间的关系，以选取相关度高的物品或用户为邻居，进行近邻推荐。邻居模型只考虑评分数据中的局部邻居作用，很难挖掘隐藏在用户评分矩阵中的潜在信息。而隐语义模型则考虑数据中的全局作用，但对小规模数据集的处理存在缺陷，所以两者的组合模型，可以避免各自的缺点，提高预测的准确性[8]。

一些研究人员建议把少量几种（通常为两种或三种）预测因素组合到单个模型，以便在模型预测时考虑多方面因素[5-6,8]。这些单个模型虽然可以组合几种预测因素，但实际预测因素可能还有更多，如何组合这些因素是组合推荐中需要解决的关键问题。

在隐语义模型与 KNN 邻居模型的组合方式中，Paterek[37]采用后

处理（Post Processing）的方式对 SVD 预测结果进行调整，他提出 SVD_KNN（Post Processing SVD with KNN）和 SVD_KRR（Post Processing SVD with Kernel Ridge Regression）两种组合算法。这种基于预测结果做模型组合的方法还包括线性组合[4]、神经网络组合[50]等。另外，Koren [8]等从得到预测结果前的模型组合角度出发，提出对称看待 KNN 邻居模型和隐语义模型，将两种模型进行融合后再进行评分结果的预测。

综上所述，可以将组合策略分为两种：预处理和后处理。前者是对称看待每个模型，通过简单地叠加进行模型地融合，以达到同时考虑多种预测因素的效果；后者则是对单个模型预测结果的组合，选择合适的数学模型进行不断地优化和调整，以达到最佳的预测结果。能与隐语义模型进行组合推荐的除了邻居模型（KNN）之外，还有如受限玻尔兹曼机（Restricted Boltzmann Machines，RBM）[51]、基于图的推荐模型[52]等。当然，并不是任何的组合方式都会得到好的推荐效果，这还与模型本身、实际应用场景等因素相关，需要在实际操作中反复实验得出结论。

2.2.2 国内研究现状

相比国外，国内对隐语义模型做专门研究的文献比较少。最相关的文献基本上是借鉴国外的研究成果，在此基础上实现或改进，缺少比较系统地介绍和研究。如鲁权等[53, 54]研究了基于协同过滤和隐语义模型的推荐系统理论与实现方法，对融合隐语义模型和邻域模型的推荐算法进行简单的优化和实现；冯晓龙[55]在 PPTV 视频数据集和 Movielens 数据集上对隐语义模型进行实验，结果表明邻居模型在显式评分数据集 Movielens 上准确度更高，而隐语义模型在隐式评分映射的 PPTV 数据集上准确度更高，最后还利用线性加权的方式实现了两种模型的混合推荐。

虽然隐语义模型在国内文献中出现的频次还比较少，但与其相关的技术方法如矩阵分解（MF）、奇异值分解（SVD）等在许多研究中都得到了应用。如张川[56]详细介绍现有矩阵分解的几种基本模型之后，为了提高矩阵分解算法的预测精度，提出基于偏差向量、用户相似度、项目相似度和融合用户相似度、项目相似度的矩阵分解办法；段华杰[57]主要是借鉴 Koren[39]的研究成果，考虑时间效应的物品偏差和用户偏差对评分结果的影响，并做具体实验。顾晔等[58]针对基础的奇异值分解算法不适宜大规模数据操作的缺点，提出基于一系列评分值对用户-物品矩阵进行分解的改进增量的奇异值分解算法；罗铁坚等[59]就时间因素对学术资源推荐系统的影响问题，提出融合奇异值分解模型和

动态转移链的学术资源推荐算法（SVD&DTC）。

总之，目前国内单独对隐语义模型的基本思想、工作原理和发展情况做专门分析和系统介绍的成果并不是很多，而且，缺少对隐语义模型在文献推荐领域的应用特点做分析和探讨的研究成果。因此，能够比较全面地介绍隐语义模型，充分了解国外对隐语义模型研究的发展状况、现有研究成果中存在的不足等，不仅能够为我们探讨文献推荐领域隐语义模型的应用问题和模型的改进方法提供指导，而且能为国内推荐系统领域技术方面的研究提供帮助。

2.3 本章小结

第二章，重点介绍与隐语义模型算法研究相关的领域知识，包括推荐系统的定义、一般个性化推荐技术的分类、文献推荐领域的常见算法，以及隐语义模型在国内外的主要研究内容等。通过系统地调研和分析，认识隐语义模型在推荐系统中的发展状态和研究情况，以及存在的问题，为后期工作内容安排和设计提供基础。

3 隐语义模型典型改进算法与实验比较

本章将从理论方法的角度，详细介绍隐语义模型的工作原理、模型表示、学习方法和评价指标等，并通过具体实验分别讨论几种典型隐语义模型算法的推荐效果，以全面认识和理解该模型在推荐系统中的应用特点。

3.1 隐语义模型的形式化定义

隐语义模型（Latent Factor Model，LFM）来源于 2006 年 Netflix Prize 大赛中受到广泛关注的奇异值分解（SVD，Singular Value Decomposition），它与 PLSA、LDA、隐含类别模型等同属于隐含语义分析技术。隐语义模型的实现基础是矩阵分解，它将用户及物品表示成向量形式，向量的取值由评分模式来确定（一般以显式评分为主），通过矩阵分解将用户和物品映射到维度为 F 的潜在因子空间，用户-物品的关系则可以表示成该空间上的内积，形成语义上的关联。

假设，给定一个用户-物品的评分数据集 R，其中，用户个数为 U，物品个数为 M，R 是一个 U×M 的矩阵。经隐语义模型建模后，可以得到如图 3.1 所示的模型：

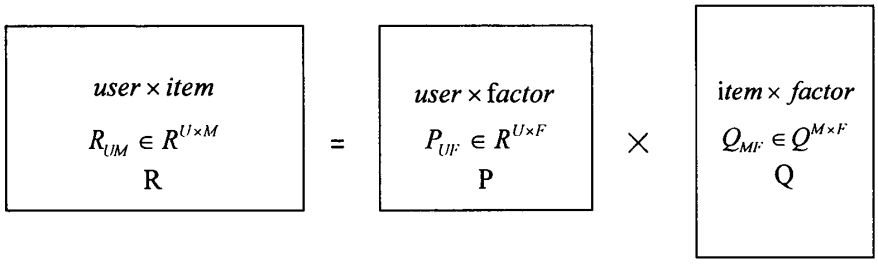


图 3.1 隐语义模型矩阵分解示意图

矩阵 P 表示用户 user 对因子类别 factor 的兴趣度，矩阵 Q 表示物品 item 在因子类别 factor 上的权值，代表物品在因子空间的特征分布。矩阵 P 和 Q 的内积表示用户对物品的总体兴趣度值，假设用 $\hat{R}_{UM}$ 表示，具体模型定义如下：

$$\hat{R}_{UM} = P_U Q_M = \sum_{f=1}^{F} p_{Uf} Q_{Mf} \quad \text{(Base-SVD)} \qquad \text{（公式 3.1）}$$

隐语义模型的工作原理可以理解为分类问题，它将用户 user 对物品 item 的兴趣偏好划分到 F 个类别，但又不明确到某一类，而是计算其属于每一类的概率，是典型的软分类问题。而且，隐语义模型并不关心分类的角度和粒度，最终的分类结果根据用户的行为数据自动形成，通过设置最终分类数目来控制粒度，分类数目越大，粒度越细，反之，粒度越粗。隐语义模型从语义分析的角度尽可能地挖掘用户和物品的潜在关联，建立属于用户自己的兴趣偏好特征模型，为最终推荐列表的形成提供了重要基础。

3.2 模型的评价指标

评价指标是衡量推荐算法好坏的重要标准。目前，常见的指标有预测准确度（以评分预测为主）、分类准确度、排序准确度等准确性指标。

然而，对于不同的推荐算法，评价指标的选择也会有所差异。主要体现在：

(1)不同类型的推荐算法在不同数据集上的表现存在差异。例如有些算法在用户数量多的数据集上推荐效果很好，但在物品数量相对较多的数据集上推荐效果并不理想。所以，数据集中用户和物品的规模、评分尺度、打分稀疏性等因素都会影响推荐结果评价指标的选择。

(2)评价目的不同。有些推荐系统看重推荐给用户的物品列表的准确度，而有些则关注推荐结果的错误率、多样性等。

(3)指标的综合评价。一般的，研究人员会将自己新的或改进的推荐算法结果与前人的实验结果进行比较，而这种比较性评价往往会受到自然变量（如时间，不同时间段可能得到不同的结果）的影响。因此，如何选择合适的指标进行综合评价是比较困难的[60]。

这里，考虑到隐语义模型的实现基础是评分数据，并且推荐算法的核心目的是为了预测用户评分，故选择预测准确度作为衡量标准，以平均绝对误差 MAE 和均方根误差 RMSE 作为推荐算法具体的评价指标。

3.2.1 平均绝对误差（Mean Absolute Error，MAE）

平均绝对误差 MAE 是推荐系统准确度评价中比较常用的一种，它计算用户实际评分与预测评分之间的平均绝对误差，具体计算如公式

3.2 所示：

$$MAE=\frac{1}{|T|}\sum_{(u,m)\in T}|\hat{r}_{um}-r_{um}| \qquad \text{（公式 3.2）}$$

其中，T 代表测试集，|T|即所有 user-item 评分对的总数，r_{um}表示用户对物品的实际打分。$\hat{r}_{um}$表示预测评分，推荐结果的误差值是所有用户预测评分与真实打分误差的平均。

平均绝对误差通俗易懂，便于操作，并且每个推荐系统的绝对误差唯一，利于比较两个系统绝对误差的差异。但该种评价指标也存在缺陷，因为对 MAE 值影响比较大的往往是那些难预测准确的低分评价，假设推荐算法 A 得到了比算法 B 低的 MAE 值，这很可能只是由于 A 在预测低分评价的效果比较好，也可能是算法 A 比算法 B 能更好地区分用户非常讨厌和一般讨厌的物品，显然这种情况下的评价并不合理。

3.2.2 均方根误差（Root Mean Squared Error，RMSE）

均方根误差与平均绝对误差相类似，都是准确度评价指标，自 Netflix Prize 竞赛之后受到广泛应用。均方根误差 RMSE 对实际评分和预测评分之间的误差做平方处理，加重了因预测不准确而产生绝对误差的惩罚，对算法的要求更加严格。均方根误差的具体定义如公式 3.3：

$$RMSE=\sqrt{\frac{\sum_{(u,m)\in P}(r_{um}-\hat{r}_{um})^2}{|T|}} \qquad \text{（公式 3.3）}$$

虽然，准确度评价指标在推荐系统领域发挥着重要作用，但也存在一些不足之处。比如，将最准确的物品推荐给用户，并不总是最好的。有时用户可能对以往喜欢的物品失去了新鲜感，更需要一些与他偏好不同的、更新颖的物品。在这种情况下，就有必要尝试考虑其他评价指标。因此，推荐系统评价指标的选择，应由推荐系统的具体目标需求决定。

3.3 学习算法

早期的隐语义模型虽然能够很好地挖掘文本潜在的语义关系，但却并未得到广泛应用。这是因为该方法具有两点重要缺陷：

一方面，隐语义模型需要对稀疏评分矩阵进行补全。一般的个性

化推荐系统中用户和物品的行为评分是非常稀疏的，一旦补全，评分矩阵就会变得稠密，需要消耗大量的存储空间，同时也会引入大量误差，这在实际操作中是非常困难且难以接受的。

另一方面，隐语义模型通过 SVD 降维，计算复杂度非常高，特别是在大规模的稠密矩阵中，需要耗费大量的时间和空间。而实际的推荐系统往往包括上千万用户和上百万物品数据，所以，该方法难以在实际中得到运用。2006 年 Netflix Prize 竞赛中，机器学习在隐语义模型求解问题中的应用受到了广泛关注，尤其是随机梯度下降法在隐语义模型求解中的成功运用，使隐语义模型的发展在推荐系统领域迈出了重要一步。

3.3.1 随机梯度下降法

梯度下降法[33]又称最速下降法，是最小化风险函数或损失函数的一种常用方法。随机梯度下降法(Stochastic Gradient Descent, SGD)是梯度下降法的一种，它通过迭代拟合原始数据，求解模型未知参数，可以有效降低计算复杂度并获得良好的训练结果。

这里，本文以线性模型为例，详细介绍梯度下降法的工作原理。假设，h(x)是需要拟合的函数，J(θ)为损失函数，θ是参数，即需要通过迭代求解的值，m 代表训练样本的条数，n 为每条样本向量空间的维度，具体定义如公式 3.4：

$$h_\theta(x)=\sum_{j=0}^{n}\theta_j x_j$$

$$J(\theta)=\frac{1}{2m}(y-h_\theta(x))^2=\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}(y^i-h_\theta(x^i))^2 \qquad \text{（公式 3.4）}$$

$$cost(\theta,(x^i,y^i))=\frac{1}{2}(y^i-h_\theta(x^i))^2\text{(单个样本的损失函数)}$$

注：$h_\theta(x^i),y^i$ 分别表示第 i 个样本的预测值和真实值；J(θ)以最小化每个样本的预测误差（真实值与预测值的差）平方和进行计算。

这里，学习目标是最小化损失函数 J(θ)。首先，给参数赋予初始值，然后通过迭代不断修改参数值，使损失函数值变小，直到函数收敛到期望的最优解，整个学习过程结束。梯度下降法的核心思想是函数在某点的梯度是一个向量，它的方向就是函数增长或下降最快的方向。但梯度下降法每次需要训练所有样本来更新参数，无论是对计算复杂度和空间复杂度的要求都非常高。所以，这里采用随机梯度下降法，通过最小化每条样本的损失函数，找到一个合适的训练路径（学

习顺序），及时更新参数θ，使损失函数整体沿着负梯度方向递减，最大可能地找到最优解[33]。具体推导过程如下：

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\cos t(\theta,(x^i,y^i)) &= \frac{\partial}{\partial\theta_j}\frac{1}{2}(y^i-h_\theta(x^i))^2 \\ &= (y^i-h_\theta(x^i))\bullet\frac{\partial}{\partial\theta_j}(y^i-\sum_{i=0}^{n}\theta_i x_i) \\ &= -(y^i-h_\theta(x^i))x_j^{\ i}\end{aligned} \qquad \text{（公式 3.5）}$$

由此，公式 3.5 经迭代可转换为：

$$\theta_j=\theta_j-\alpha\frac{\partial}{\partial\theta_j}\cos t(\theta,(x^i,y^i))=\theta_j+\alpha[(y^i-h_\theta(x^i))x_j^{\ i}] \qquad \text{（公式 3.6）}$$

其中，参数α又称为学习速率（learning rate），用来控制每一次迭代的步长，如果选择太小，算法收敛慢，如果选择太大，容易导致发散。随机梯度下降法以最小化每条样本的损失函数，找到梯度下降的方向，虽然不是每次迭代得到的梯度都是全局最优的方向，但也在最优解附近。因此，最终得到的结果并不是全局最优解，往往是靠近最优解的值。

3.3.2 基于随机梯度下降法的隐语义模型

隐语义模型是基于评分预测进行推荐的算法，其核心问题是矩阵填充，经填充后的评分矩阵可以得到用户对物品的预测评分，推荐模型就可以根据预测评分的排序结果实现推荐。传统的奇异值分解对矩阵填充的代价非常大，所以。通常采用随机梯度下降法，基于最小化RMSE构建损失函数，直接通过训练集中的观察值学习P、Q矩阵，损失函数定义为：

$$C(p,q)=\frac{1}{2}\sum_{(u,m)\in T}(r_{um}-\sum_f p_{uf}q_{mf})^2 \qquad \text{（公式 3.7）}$$

对上述损失函数直接优化容易导致学习的过拟合，因此在损失函数中加入正则项以避免该问题的出现，其中λ是正则项系数：

$$C(p,q)=\frac{1}{2}\sum_{(u,m)\in T}[(r_{um}-\sum_f p_{uf}q_{mf})^2+\lambda(\|p_u\|^2+\|q_m\|^2)] \qquad \text{（公式 3.8）}$$

采用随机梯度下降法对公式 3.8 进行优化，通过求参数的偏导找到模型的负梯度，然后迭代不断优化参数。这里，主要有两组参数（p_{uf}和q_{mf}），首先对它们求偏导数，可得：

$$\frac{\partial}{\partial p_{uf}} = -(r_{um} - \sum_{f} p_{uf} q_{mf}) q_{mk} + \lambda p_{uk} = -e_{um} * q_{mk} + \lambda p_{uk}$$

同理：

$$\frac{\partial}{\partial q_{mf}} = -(r_{um} - \sum_{f} p_{uf} q_{mf}) p_{uk} + \lambda q_{mk} = -e_{um} * p_{uk} + \lambda q_{mk} \qquad \text{（公式 3.9）}$$

$$p_{uf} = p_{uf} + \alpha(e_{um} * q_{mk} - \lambda p_{uk})$$

$$q_{mf} = q_{mf} + \alpha(e_{um} * p_{uk} - \lambda q_{mk})$$

其中，e_{um} 表示观察值与预测值的误差，α 是学习速率，它的取值可通过反复实验获得。而正则项系数 λ 是影响预测结果准确度的重要因素，对于超参数 λ 的选择需要在训练集上反复验证。模型中参数的初始值通常采用随机数生成器进行采样，如采用阈值为(0,1)的浮点数对参数做初始化等。

3.4 隐语义模型的改进算法

隐语义模型是协同过滤推荐技术中重要的一类，是基于矩阵分解技术实现推荐的算法。它在 Netflix Prize 竞赛被提出之后，许多著名的模型都是通过对隐语义模型的改进和补充获得的，接下来将逐一介绍几种典型的改进模型。

3.4.1 Bias-SVD

推荐系统中观察到的用户-物品评分，不仅和用户、物品间的交互相关，而且很大程度上也与用户或物品本身相关，这被称为偏差或偏置（Bias）。例如，一个评分系统有些固有属性与用户或物品无关，而用户也有些属性与物品无关，物品有些属性与用户无关，部分用户总体上给的评分都较高或较低，而部分物品得到的评分总体上也都较高或较低等，这些都可以认为是偏差或偏置。因此，只考虑用户和物品的交互关系并不太合理。相反，尝试通过偏差解释用户或物品本身的属性特征，是对原有隐语义模型的很好补充[8]。偏置项定义如下：

$$b_{um} = \mu + b_u + b_m \qquad \text{（公式 3.10）}$$

b_{um}：评分的偏置项，表示评分系统、用户和物品本身的属性特征；

μ：全局平均数，所有评分对的平均数，代表评分系统本身对用户

评分的影响；

b_u：用户偏置（user bias），表示因用户的评分习惯与物品无关部分的因素，如用户喜欢打高分等；

b_m：物品偏置（item bias），表示物品接受的评分中与用户无关部分的因素，如物品本身的质量等。

结合偏置项的定义，预测评分 $\hat{r}_{um}$ 可重新定义为：

$$\hat{r}_{um} = \mu + b_u + b_m + p_u q_m^T \qquad \text{（公式 3.11）}$$

这里，模型主要包括四个部分，全局平均数、用户偏置、物品偏置以及用户物品间的交互关系，除全局平均数可直接得到之外，其他参数需要通过学习算法求解得到，λ_1、λ_2分别为偏置和用户、物品因子的正则项系数。Bias-SVD 学习的目标函数可定义为：

$$C = \min \sum_{(u,m)\in T} [(r_{um} - \mu - b_u - b_m - p_u q_m^T)^2 + \lambda_1 (b_u^2 + b_m^2) + \lambda_2 (\| p_u \|^2 + \| q_m \|^2)]$$

（公式 3.12）

3.4.2 SVD++

协同过滤推荐技术实现的基础是推荐系统中用户的历史行为信息，主要有两种：一是用户通过系统明确展示给我们的反馈信息（如评分等），称之为显式反馈（explicit feedback），二是那些用户在系统中操作的，但没有明确反馈意图的行为（如浏览等），称之为隐式反馈（implicit feedback）。

一般的，推荐系统对用户的显式反馈比较敏感，希望尽可能多地得到用户的显式行为。但在实际问题中，部分用户并不喜欢给予物品明确的评价，没有显式反馈意图，所以用户的显式信息比较难获取，而且在很大程度上存在缺失。虽然，用户没有留下大量的显式反馈数据，但系统对用户的操作行为存有记录，而这些被称为隐式反馈的行为记录蕴含了用户的观点或偏好，这对推荐系统分析用户特征也是非常有帮助的。

因此，在 Bias-SVD 的基础上引入用户对物品的隐式反馈信息，对评分预测模型做进一步修正，可得 SVD++模型[8]：

$$\hat{r}_{um} = b_{um} + q_m^T (p_u + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} y_j) \quad \text{（公式 3.13）}$$

SVD++中将用户定义为 $p_u + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} y_j$，其中，$p_u$是用户因子向量，可以通过显式评分数据学习得到，$\frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} y_j$是用户的隐式反馈部分，N(u)表示用户 u 的隐式反馈集合，y_j表示用户隐式评分下的物品因子向量。q_m表示显式评分下的物品因子向量。通过最小化 RMSE 可建立目标函数：

$$C = \min \sum_{(u,m) \in T} [(r_{um} - b_{um} - q_m^T (p_u + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} y_j))^2 + \lambda_1 (b_{um}^2) + \lambda_2 (\|p_u\|^2 + \|q_m\|^2) + \lambda_3 \sum_{j \in N(u)} \|y_j\|^2]$$

（公式 3.14）

SVD++补充考虑了隐式反馈行为对评分预测结果的影响，在原有隐语义模型的基础上，将用户因子分为显式评分和隐式评分两部分，综合考虑了影响评分预测结果的因素，使得模型信息更加完整。但 SVD++增加了未知参数的数量，对模型的学习提出了更高要求。

3.4.3 Asymmetric-SVD

如果在 SVD++模型的基础上，将每个物品 m 与三个维度相同的物品因子向量关联，即 p_m、y_m、$x_m \in R^f$。那么，用户因子向量没有提供显式参数，而是考虑用户评分过的物品，利用用户的历史评分偏好建立用户特征向量，则得到考虑邻域和隐式反馈的 Asymmetric-SVD[8]模型：

$$\hat{r}_{um} = b_{um} + q_m^T (\frac{1}{\sqrt{|R(u)|}} \sum_{j \in R(u)} (r_{uj} - b_{uj}) x_j + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} y_j) \quad \text{（公式 3.15）}$$

R(u)表示用户 u 的显式评分集合，x_j表示在给定的显式评分集中物品 j 的插值权重，其目标函数定义如公式 3.16：

$$C = \min \sum_{(u,m) \in T} (r_{um} - \hat{r}_{um})^2 + \lambda_1 (b_u^2 + b_m^2) + \lambda_2 (\|p_u\|^2 + \|q_m\|^2) + \lambda_3 \sum_{j \in R(u)} \|x_j\|^2 + \lambda_4 \sum_{j \in N(u)} \|y_j\|^2$$

（公式 3.16）

该模型相比上述几种模型具有以下优点：

(1)参数少。一般个性化推荐系统中用户的数量远大于物品，用物品参数表示用户参数，减少了未知参数的个数，降低了模型复杂度。

(2)新用户处理。Asymmetric-SVD 模型在处理推荐系统的冷启动问题上表现出特有的优势。因为该模型不需要生成用户参数，对于新用户而言，只要他们对系统提供了反馈，系统就可以及时产生推荐，而不需要重新训练模型或估计参数。并且，系统还可以根据用户的新行为，及时更新用户的偏好特征。但对于新物品而言，参数的学习还是需要通过训练得到的。Asymmetric-SVD 模型体现了用户和物品的非对称性，这恰恰也符合推荐系统的实际应用。一方面，推荐系统对新用户推荐的及时性要求比较高，另一方面，系统中新物品的引入具有时段性，这给新物品参数的估计提供了重新训练的时间。

(3)解释性。隐语义模型，例如 Base-SVD 对评分预测结果的解释性是比较差的，因为它通过用户因子向量这个中间层表达用户偏好，将可解释用户偏好的行为和预测结果分开了。而 Asymmetric-SVD 并没有对用户层面做抽象处理，直接通过用户的历史反馈行为预测结果，这可以帮助系统找出对预测结果影响最大的那些历史行为。

(4)整合隐式反馈。Asymmetric-SVD 模型综合考虑了用户的显式和隐式反馈信息，当用户的显式反馈数据较多，则 R(u)比较大，显式反馈作用明显；当用户隐式反馈数据较多，则 N(u)比较大，隐式反馈作用明显。当然，单个显式反馈比单个隐式反馈价值更大，至于两者的比例，需要通过设置合适的参数 x_j、y_j和不断的学习进行设定。

3.5 实验与讨论

3.5.1 Movielens 数据集

MovieLens 电影推荐系统，可以允许用户对自己看过的电影打分，并且根据用户的历史打分行为，预测用户对其他未观看过电影的打分，将预测分值高的推荐给用户，认为这些电影是用户所感兴趣的。

本节对隐语义模型的几种典型算法的评估实验主要基于 MovieLens 数据集，它包括用户 ID 信息、物品 ID 信息，用户对物品的评分信息等。实验过程将评分数据集分为训练集（80%）和测试集（20%），共随机划分为 5 组，最终实验结果以 5 次交叉验证得到的 MAE 平均值和 RMSE 平均值为最终衡量标准，数据集的详细信息可见表 3.1。

表 3.1 MovieLens 数据集

数据集	用户数	电影数	评分数	训练集比例 x
MovieLens	943	1682	100000	80%

3.5.2 实验参数

本节实验选取的算法有 Base-SVD、Bias-SVD、SVD++和 Asymmetric-SVD，分别通过实验讨论它们在显式评分预测问题中的性能特点。算法中涉及的重要参数主要包括：

(1)隐特征维度 F：隐语义模型的核心思想就是将评分矩阵进行降维，分解得到用户特征矩阵 P 和物品特征矩阵 Q，特征的维度 F 代表了保留评分矩阵信息的多少，F 越大，保留信息越多，但降维的优点就难以体现；F 越小，保留信息越少，但预测结果可能会出现较大偏差。因此，F 值的确定需要根据模型的特点、实际应用场景和实验情况等因素综合考虑来确定。对于受隐特征维度影响大的模型，需要慎重选择最佳的 F 值；而对于受隐特征维度影响小的模型，操作过程中可以减少对其选择问题的考虑。

(2)学习速率α：学习速率的大小直接关系到模型迭代过程参数的改变量，如果α过大，模型收敛速度比较快，而且在梯度下降过程中很有可能会跳过最优解；如果α过小，模型收敛所需的迭代次数就非常多，收敛速度比较缓慢。当然，对于不同模型，学习速率的取值是有差异的。为了避免出现跳过最优解的现象，可以在实验过程中采用变化的学习速率方法，比如学习速率的初始值可以精度大一些，每次迭代就缩小一点，如$\alpha=\alpha*0.9$等。

(3)正则项系数λ：λ的选择也是需要根据算法本身、数据集等的特点，通过多次实验来确定，如果选择过大，欠拟合问题就会出现；如果选择太小，正则化的效果就会不明显。

此外，在实验开始前，需要对未知参数即用户特征因子 P 和物品特征因子 Q 在每个隐特征维度上的因子变量赋予初始值，然后才能通过不断迭代沿着梯度方向往下寻找模型最优解。这里，本文在实验中采用的方法是随机产生 0～1 的浮点数，并考虑隐特征维度 F，计算 0.1*rand(0,1)/sqrt(F)作为 P 和 Q 因子的初始值，同时以多次实验结果的平均值作为最终预测结果。

3.5.3 结果分析

(1)隐特征维度 F 对算法性能的影响

隐语义模型在实现过程中，隐特征维度的选择是一个难点，它关

系到保留用户和物品评分信息多少的问题。为了探讨隐特征维度 F 在隐语义模型中的作用，以及对最终预测结果的影响，实验设置相同的学习速率 α 和正则项系数 λ，即 α=0.005*0.9、λ_1=0.05、λ_2=0.1、λ_3=0.0175、λ_4=0.00075 作为实验参数，在训练集上迭代 100 次，改变 F 的值，经多次反复实验得到的预测结果作为分析对象，讨论几种算法受隐特征维度 F 的影响情况。

实验主要基于 MovieLens 数据集，经五次交叉验证得到不同隐语义模型的预测误差 RMSE 和 MAE 分布情况，部分数值见表 3.2，分布曲线如图 3.2、图 3.3 所示。

表 3.2 不同隐语义模型在隐特征维度 F 变化下的预测误差分布（部分）

模型	Base-SVD		Bias-SVD		SVD++		Asymmetric-SVD	
F	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
20	0.995845	0.786098	0.947078	0.750418	0.941204	0.745268	0.923481	0.730193
40	0.995846	0.786102	0.94708	0.750421	0.94117	0.74524	0.9225	0.728955
60	0.995844	0.786096	0.947078	0.750418	0.941192	0.745255	0.922501	0.728906
80	0.995843	0.7861	0.947079	0.750419	0.94119	0.745257	0.922074	0.728672
100	0.995841	0.786098	0.947079	0.750419	0.941195	0.74526	0.922016	0.728671
150	0.99584	0.786096	0.947079	0.750419	0.941185	0.745251	0.921809	0.728443
200	0.995837	0.786096	0.947079	0.750419	0.941187	0.745252	0.921856	0.72851
300	0.995833	0.786095	0.947079	0.750419	0.941187	0.745239	0.921705	0.728393
500	0.995839	0.786097	0.947075	0.750413	0.941189	0.745241	0.92168	0.728359
700	0.99609	0.786283	0.947076	0.750413	0.941183	0.745234	0.921743	0.728329
1000	0.995839	0.786097	0.947079	0.750419	0.941213	0.745264	0.921738	0.728311

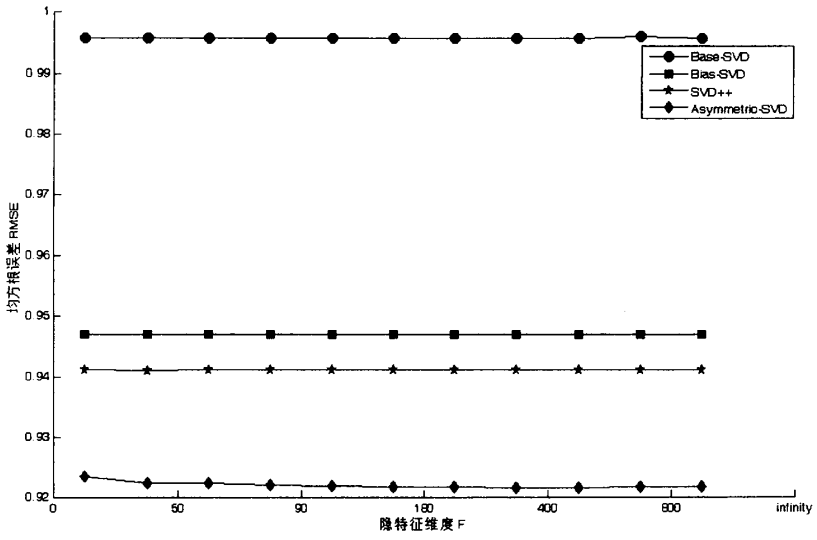


图 3.2 不同隐语义模型随隐特征维度 F 变化的 RMSE 值分布

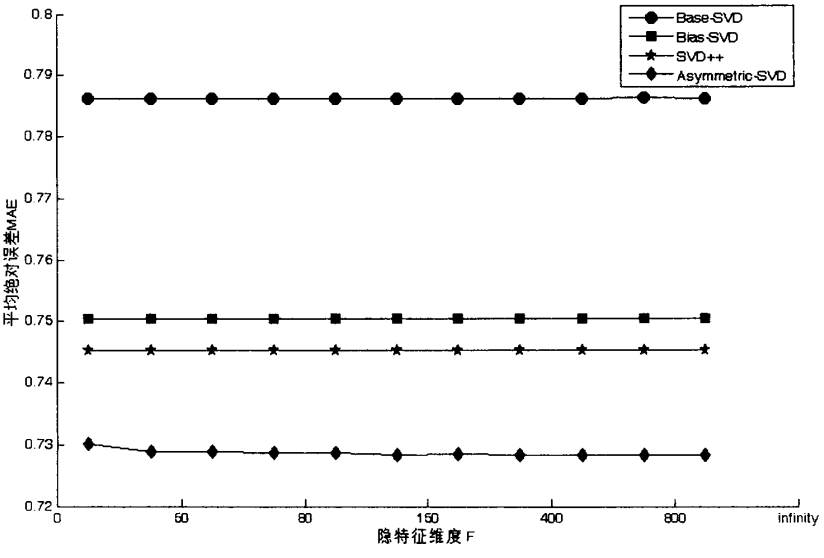


图 3.3 不同隐语义模型随隐特征维度 F 变化的 MAE 值分布

由表 3.2 和图 3.2、图 3.3 可知，模型 Base-SVD、Bias-SVD 和 SVD++随隐特征维度 F 的不断变化，均方根误差 RMSE 和平均绝对误差 MAE 值变化非常小，尤其是模型 Bias-SVD，它在不同 F 值取得的预测误差，变化的数量级约为 10^{-5} 左右。同时，从分布曲线的波动幅度来看，这三种模型几乎接近一条水平直线，保持相对平稳的状态，这表明隐特征维度 F 对以上三种模型的推荐效果或预测精度影响并不大。因此，在对这三种模型进行实际操作过程中，可以减少对隐特征维度选择问题的考虑。

然而，模型 Asymmetric-SVD，在隐特征维度 F 较小时，预测误差下降的速度比较快，随着 F 的增加，下降速度逐渐变得缓和。表明该模型在隐特征维度 F 偏小时，取得的预测误差较大；随着隐特征维度的增加，预测误差先逐渐减小，然后当 F 达到一定维度之后，预测误差就不再减小，趋于稳定。因此，可以认为模型 Asymmetric-SVD 受隐特征维度 F 的影响相比其他模型要大，推荐精度随 F 的变化要明显。由此可知，在实际推荐系统中，对于不同的隐语义模型，隐特征维度的选择应有所区别，对于受影响大的模型，应经过多次反复实验，确定最终维度；对于受影响小的模型，可以减少对隐特征维度选择的考虑。

此外，观察图 3.2 和图 3.3，模型 Base-SVD 预测效果最差，RMSE 和 MAE 值最大，变化曲线处于最高位置；其次是引入偏置项的隐语义模型 Bias-SVD；然后是考虑隐式反馈的模型 SVD++；而融合邻域的模型 Asymmetric-SVD 预测效果最佳，分布曲线位于最低位置。这也验证了在本次实验中，保持相同参数设置的条件下，改进的模型 Asymmetric-SVD 能够提高隐语义模型的预测精度。同时也从侧面反映了随着模型的不断改进，考虑的因素不断增加，预测误差逐渐减小，推荐效果不断提高。当然，也会存在一些缺陷，如模型复杂度的提高，需要消耗更大的时间和空间等，这在实际推荐系统设计中是需要考虑的重要因素。

(2)学习速率 α 对算法性能的影响

学习速率控制的是算法在迭代过程中参数的改变量，它的大小直接关系到模型最后的预测结果。本次实验采用固定隐特征维度 F=40，正则项系数不变，改变学习速率 α 和相应的迭代次数，对不同模型的预测误差变化情况进行分析，四种算法的实验结果分别如图 3.4-3.7 所示。

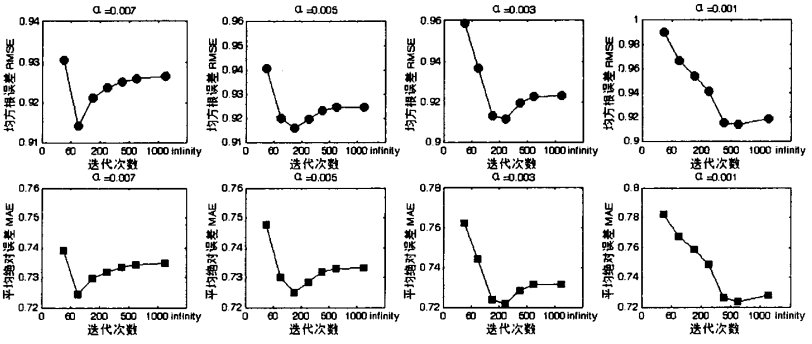


图 3.4 模型 Base-SVD 学习速率和迭代次数变化曲线

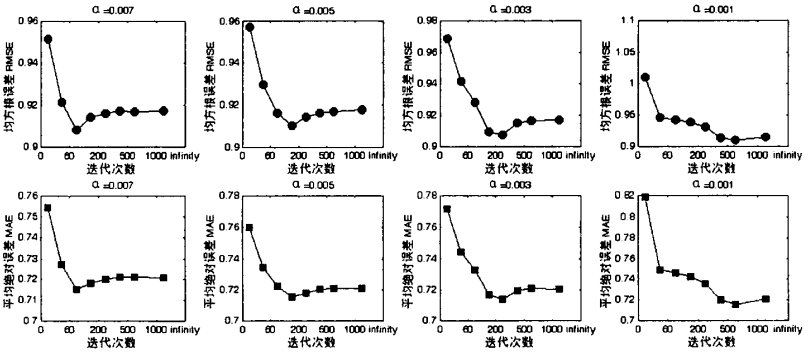


图 3.5 模型 Bias-SVD 学习速率和迭代次数变化曲线

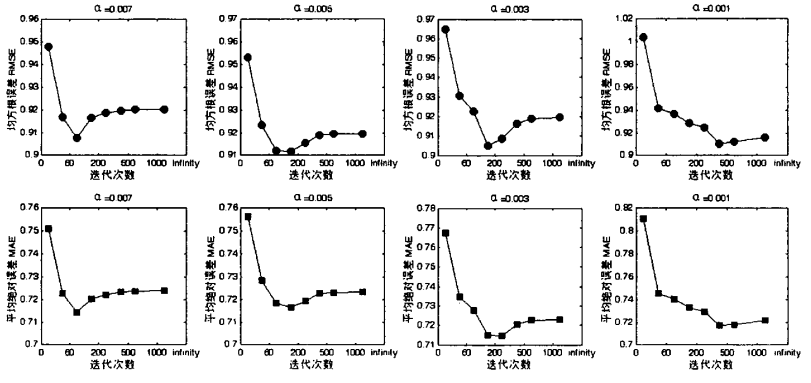


图 3.6 模型 SVD++学习速率和迭代次数变化曲线

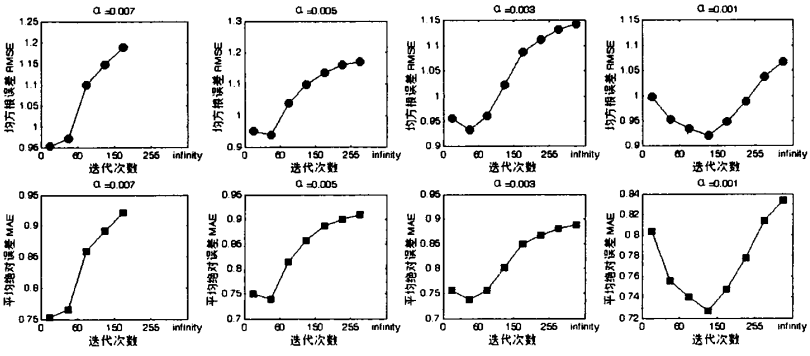


图 3.7 模型 Asymmetric-SVD 学习速率和迭代次数变化曲线

从学习速率和迭代次数的变化关系来看，以模型 Asymmetric-SVD 为例，观察图 3.7 发现，学习速率由 0.007 降到 0.001，模型的预测误差 RMSE 和 MAE 的曲线变化有所不同。例如，当α=0.007，迭代次数越小，模型预测误差越低，推荐效果越好，迭代次数变大，预测误差反而增大，甚至得到趋于无穷大的数值；当α=0.005 时，预测误差随迭代次数的增加先有所下降，取得较低的预测误差，但随后迭代次数再变大，预测误差又随之升高；当α继续下降，模型取得较低预测误差要求的迭代次数越多，而得到无穷大数值的情况也越不容易出现。

由于论文中采用的是有限次数的实验，其他模型学习速率和迭代次数的变化曲线特征可能没有模型 Asymmetric-SVD 明显，但都有类似的变化趋势，只是不同模型对学习速率的具体数值要求有所不同而已。如图 3.4，模型 Base-SVD，取得较低误差的迭代次数相对比较大，而且在学习速率α为 0.007 至 0.001 之间，有限次数的实验过程中，预测误差也并未出现无穷值。但随迭代次数的不断增加，预测误差呈现上升的趋势。假设，当迭代次数无限大时，预测误差可能也会取得无穷值，跳过最优解。

从学习速率对模型性能影响的角度来看，学习速率由 0.007 下降到 0.001，不同模型取得的局部最低预测误差值，以及误差的变化情况都有所不同。从表 3.3 的学习速率和迭代次数变化情况来看，学习速率逐渐降低，预测误差也逐渐减小，但相应迭代次数却不断增加，这也进一步验证了更小的学习速率能够取得较好的预测误差，但对算法效率的要求比较大。此外，四种模型在学习速率不断改变的情况下，取得的预测误差值具体变化程度有所不同。

如模型 Bias-SVD 变化最小，它的 RMSE 值从 0.90827 降到

0.90688，减少了 0.00139，MAE 值从 0.71526 降到 0.71329，减少了 0.00197；模型 Asymmetric-SVD 变化最大，它的 RMSE 值从 0.95203 降到 0.92134，减少了 0.03069，MAE 值从 0.75338 降到 0.72725，减少了 0.02613。由此可知，模型 Asymmetric-SVD 受学习速率影响的程度最大，模型 Bias-SVD 最小。而且，在实验过程中发现，相同学习速率下，模型 Asymmetric-SVD 相比其他模型更容易跳过最优解。因此，对该模型在实际操作中应选择较小的学习速率。

实验结果表明：当模型的学习速率取值偏大，取得较好推荐效果所需的迭代次数比较少，模型的收敛速度比较快，若迭代次数过大，预测误差容易趋于无穷大，很有可能在沿着梯度方向寻找局部最优解的时候会跨过最优解；当学习速率取值偏小，取得较好推荐效果所需的迭代次数比较多，模型的收敛速度比较缓慢，需要很长时间才能找到局部最优解。虽然，更小的学习速率可以产生更低的预测误差，但同时算法的收敛速度也变得更慢。因此，在实际算法应用中，除了考虑模型本身的特性，还应根据推荐系统的内存空间、时间性能要求等因素，综合考虑以选择较好的学习速率和迭代次数等。

表 3.3 不同学习速率和迭代次数下模型局部最低预测误差分布

Base-SVD				Bias-SVD			
学习速率	迭代次数	RMSE	MAE	学习速率	迭代次数	RMSE	MAE
0.007	95	0.91409	0.72437	0.007	95	0.90827	0.71526
0.005	195	0.91572	0.72485	0.005	195	0.91038	0.71564
0.003	255	0.91157	0.72165	0.003	255	0.90792	0.7143
0.001	700	0.91022	0.72056	0.001	700	0.90688	0.71329
SVD++				Asymmetric-SVD			
学习速率	迭代次数	RMSE	MAE	学习速率	迭代次数	RMSE	MAE
0.007	95	0.91022	0.72056	0.007	5	0.95203	0.75338
0.005	195	0.91022	0.72056	0.005	55	0.93763	0.7391
0.003	195	0.91022	0.72056	0.003	55	0.93313	0.7383
0.001	500	0.91022	0.72056	0.001	195	0.92134	0.72725

注：实验过程存在一定的局限性，比如有限次数的实验、参数的设定等问题，所以实验得到的预测误差值并不能完全反映模型的最优解，只能作为局部问题讨论的参考数据，深入的分析需要更多实验。

3.6 本章小结

本章重点介绍了隐语义模型的基本原理、学习算法和评价指标等，分别对 Base-SVD、Bias-SVD、SVD++和 Asymmetric-SVD 等四种典型算法的核心思想、模型表示和学习方法做了详细描述，并通过实验对不同算法的性能进行比较分析。结果表明：隐特征维度对模型 Base-SVD、Bias-SVD 和 SVD++影响较小，对模型 Asymmetric-SVD 影响较大，且该模型的推荐结果预测精度较其他模型要好。由此证明，融合邻域和考虑隐式反馈的隐语义模型能够取得较好的推荐效果。此外，针对控制算法参数改变量的学习速率，从迭代次数和对模型性能影响的角度，分析和讨论了它在算法实现中存在的问题和作用，得出该参数的选择对模型的预测结果影响较大，最终参数值的确定需综合考虑算法本身和推荐系统的需求。

4 基于隐语义模型的文献推荐算法

文献推荐与一般个性化推荐相比，有其领域的特殊性。如文献有作者、关键词、引文、学科类别等属性特征，但用户往往很少给文献给予显式反馈，如打分。针对文献推荐的特殊性，用户隐式反馈数据在推荐模型实现过程中必不可少。此外，一般个性化推荐技术往往仅考虑用户与物品间的关系，很少考虑用户间或物品间存在的近邻关系。因此，本章节将讨论文献推荐领域，基于用户隐式反馈行为，采用异构信息网络发现文献间的近邻关系，融合隐语义模型，建立适合文献特征的推荐模型。

4.1 用户-文献行为表示

文献推荐相比一般个性化推荐来说，用户的显式评分行为更不容易被获取。因为文献的选择与用户研究兴趣、阅读爱好、关注内容、研究领域等密切相关，更侧重文献内容是否满足用户的认知需求，用户很少愿意对文献赋予显式评分。对于缺少评分数值的文献数据集，可采用二值矩阵的方式描述用户-文献间的交互行为，认为有行为的即为“1”，其他即为“0”（具体定义如下），至于用户-文献的交互频次，文中暂不做考虑。

$$R_{i,j}=\begin{cases}1 & \text{用户和文献间有交互}\\ 0 & \text{其他}\end{cases}$$

4.2 异构信息网络

4.2.1 异构信息网络的定义

异构信息网络（Heterogeneous information network，HIN）是由多类型的对象（objects）和对象间的关系链接（links）所组成的逻辑网络，这种信息网络无处不在，是现代化信息建设的重要组成部分，如社交网络、科研协作网络、商品推荐网络等。它不仅能够帮助我们很好地观察网络的内部结构，也能了解每个对象在信息空间中所起的不同作用，对信息网络的获取和挖掘、推荐以及网络模式的演化预测具有重要的研究价值[61]。

从信息网络的角度出发，推荐领域用户、物品、属性及行为就可以构成一个完整的信息网络结构，对象（如用户、产品、卖家、销售商等）形成网络的节点，关系（如购买、浏览、评分、打标签等）生成网络的链接，这些节点和链接具有语义上的关联。考虑多类型信息构建异构推荐网络模型，通过链接的关联特性预测用户对项目的兴趣程度，并动态更新推荐网络，实现最终预测，生成推荐结果，可以有效缓解推荐领域数据稀疏性和可扩展性问题[62]。

4.2.2 异构信息网络的处理

根据异构信息网络的通用模型，将文献领域的异构信息网络表示为 G=(V, E)，V 表示网络中的对象类型（节点）（如文献集合 I，用户集合 U），E 表示关系链接（边）（如 U 和 I 的交互）。V 和 E 可以通过多种语义关系进行关联，文中采用元路径[63,64]的方式对这种关系进行描述。以用户 U 或物品 I 同类型对象间的链接表示用户或物品的近邻关系，用户 U 和物品 I 的链接表示两者的交互行为。

对于科技文献领域，异构信息网络的节点和边的生成，可作如下说明：

文献元数据：即文献的内容属性，如作者、题名、关键词、分类号、被引次数、发表刊物等；

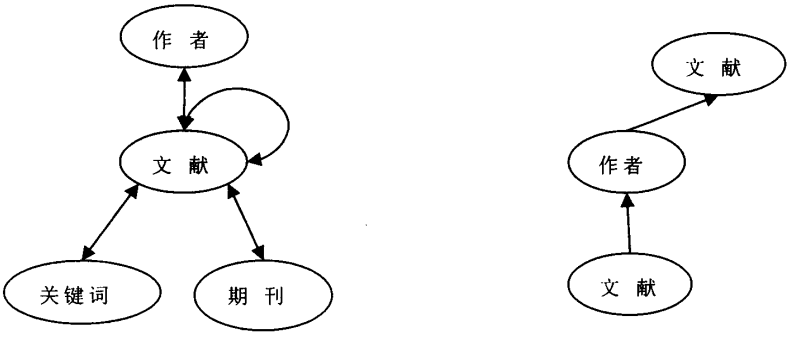
用户元数据：即用户的属性，如姓名、研究方向、住址、所在单位等；

关系信息：如文献与文献的引用关系、文献与标签（或关键词概念）的标注关系、用户与文献的行为关系等。

协同信息：指的是通过用户的历史行为（如对论文的引用、著作、浏览等）计算出的相似用户或相似论文信息。

定义 1：元路径（meta-path），即通过一组关系连接多类型对象的路径，用来描述异构信息网络中不同类型对象之间潜在的语义关系。

由元路径的定义，科技文献推荐模型对以上元数据、关系信息进行抽取，就会得到文献—文献、文献—作者—文献等各种类型的点和边，形成异构信息网络结构。例如，图 4.1 所示是一个简单的文献异构信息网络图和元路径实例。



文献异构信息网络图（部分）　　元路径：PAP

图 4.1 文献异构信息网络结构图例

对于推荐系统来说，采用元路径的方式，从语义层面上认识用户或物品的关联是有意义的，主要体现在：

(1)结合元路径方式能够充分考虑用户的行为信息，尤其是隐式反馈数据，便于从不同的角度挖掘用户或物品的潜在关联特征，获得新的知识；

(2)通过元路径的语义关联，以信息扩散的方式预测相似邻居，能在很大程度上缓解数据稀疏性问题。

基于元路径连接的对象间的近邻关系（如文献间的关联），文中主要采用 PathSim[64]相似度度量方法进行计算，定义如下：

定义 2：PathSim，即一种基于路径计算相似度的方法，对于给定的元路径 P（对称性），对象 x 和 y 的相似度为：

$$s(x,y)=\frac{2\times|\{p_{x\sim y}:p_{x\sim y}\in P\}|}{|\{p_{x\sim x}:p_{x\sim x}\in P\}|+|\{p_{y\sim y}:p_{y\sim y}\in P\}|} \qquad \text{（公式 4.1）}$$

其中，$p_{x\sim y}$表示对象 x 和 y 之间的语义关联，$p_{x\sim x}$和 $p_{y\sim y}$分别表示 x、y 自身的关联。至于关联的量化，可以用满足对象间元路径关系的路径数量进行赋值。

文中对给定的文献 e_i和 e_j，元路径 L，相似度函数能够返回范围为[0,1]的文献 e_i和 e_j的相似度值，表示在元路径 L 下，文献 e_i和 e_j的语义关联程度。假设有 L 种元路径，那么对象间就有 L 种语义关联，其对应的相似度分别为 $s^{(1)}, s^{(2)}, \cdots\cdots, s^{(L)}$。

4.3 文献推荐模型

4.3.1 目标任务

文献是科研活动的重要元素，文献推荐的目的是在一个文献库中向用户推荐满足其兴趣需求的文献集合，以减少用户检索、浏览和筛选文献的时间。假设给定了用户-文献的二值矩阵 R，以及相关的异构信息网络 G，对于某一用户 u_i，推荐模型的目标就是为用户 u_i 推荐其感兴趣的文献列表 $I_{ui} \in I$，并且认为这些文献就是用户 u_i 所需要的。

4.3.2 模型描述

本小节主要针对文献评分数据的稀疏性问题，以及文献推荐模型的实现机制，利用用户-文献的隐式反馈行为和元路径下文献间的近邻关系，重新构建用户-文献的评分矩阵，然后融合隐语义模型生成用户、文献的潜在特征因子，实现推荐模型的评分预测。

步骤 1：构建元路径下的偏好矩阵

根据 4.1、4.2 节介绍的用户-文献二值矩阵的表示和相似度度量方法，采用文献间的相似度和用户-文献的交互行为，可以建立用户在不同元路径下的兴趣偏好模型。给定二值矩阵 $R \in R^{m \times n}$（隐式反馈）、文献间相似度矩阵 $S^{(l)} \in R^{n \times n}$，则用户 u_i 对文献 e_j 的偏好可定义为：

$$\tilde{R}_{ij}^{(l)} = \sum_{t=1}^{n} R_{it} S_{jt}^{(l)} = RS^{(l)} \qquad \text{（公式 4.2）}$$

$\tilde{R}_{ij}^{(l)}$ 表示第 l 条元路径下用户对文献的偏好（评分）矩阵。这里，采用异构信息网络处理方式，以用户有过行为的文献和文献的近邻关系对用户-文献的交互行为进行描述，能够更好地挖掘用户潜在感兴趣的文献。

步骤 2：融合隐语义模型生成潜在因子特征

步骤 1 中构建完成的元路径下的偏好矩阵 $\tilde{R}_{ij}^{(l)}$，作为隐语义模型的数据基础，通过矩阵分解生成用户和文献的潜在因子特征矩阵 $P^{(l)}$ 和 $Q^{(l)}$，得到用户、文献在不同语义关系下的特征分布。那么，该步骤的目标函数可定义为：

$$(P(l), Q(l)) = \arg\min_{P,Q} \| \tilde{R}^{(l)} - P^{(l)} Q^{(l)^T} \|_F^2 \qquad \text{（公式 4.3）}$$

通过对目标函数的求解，可得到 L 条元路径下用户、文献的因子特征对（$P^{(1)}$，$Q^{(1)}$；$P^{(2)}$，$Q^{(2)}$；……；$P^{(l)}$，$Q^{(l)}$），每一对因子代表了用户、文献在特定语义关系下的特征分布。如果 L=1，表明只考虑一种语义关系，那么，模型可以直接运用潜在因子特征的内积，预测用户对文献的评分；如果 L>1，表明不止考虑一种语义关系。而且，不同元路径对用户或文献间的语义关联影响程度有所不同。所以，对 L 条元路径下的因子特征可以赋予一定的权值θ（L>1），以区分元路径的影响。然后，对 L 条元路径进行线性组合，生成最终的预测模型。

$$\hat{r}_{u_i,e_j} = \sum_{l=1}^{L} \theta_l P_i^{(l)} Q_j^{(l)} (\theta_l \geq 0) \qquad \text{（公式 4.4）}$$

步骤 3：目标函数和参数求解

步骤 2 中元路径的组合模型，参数θ的权值分配是首先需要解决的关键问题。一般地，考虑用户隐式反馈的推荐模型，其评分往往定义为两种，“1”表示正反馈，“0”表示负反馈。那么，对于这类模型参数的学习方法主要有两种：分类和排序。

方法一：分类(Classification)

由于用户的隐式评分只有“1”和“0”，所以，可以将文献推荐看作是二分类问题。根据元路径下用户和物品的特征因子，尝试对给定的用户-文献对（u_i, e_j），给出每条元路径下的一个评分预测$\hat{r}_{u_i,e_j}^{(l)}$，将其作为变量 x，用户对物品的实际评分为 y（y=0 或 1），参数即为元路径权值θ，运用回归分类模型 LR（Logistic Regression）对参数进行求解，具体目标函数表示如下（推导过程参见附件 A）：

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{l}\theta_j^{\,2} \qquad \text{（公式 4.5）}$$

其中，m 表示训练样本的数量，i 表示第 i 个样本数据，y 表示真实评分 y=0 或 1，h 表示预测评分(h∈(0,1))。

方法二：贝叶斯优化排序（Bayesian Ranking Optimization）

假设，用户对给予评分“1”的文献比评分“0”的文献更感兴趣。那么，基于该假设定义的目标函数，针对每个用户对文献的交互行为，应该确保每一个评分对的排列顺序是正确的，即评分为“1”的文献始终要排在评分为“0”的文献前面[42]。具体定义如下：

(1)数据假设：

a：每个用户之间的偏好行为相互独立；

b：同一用户对不同物品的偏好行为相互独立。

那么，用户 u_i 对文献 e_a 和 e_b 更偏向 e_a 的概率为 $p(e_a > e_b; u_i \mid \theta)$，则模型的似然函数可表示为后验概率 $p(R \mid \Theta)$：

$$p(R \mid \Theta) = \prod_{u_i \in U} \prod_{(e_a > e_b) \in R_i} p(e_a > e_b; u_i \mid \theta) \qquad \text{(公式 4.6)}$$

其中，$(e_a > e_b) \in R_i$ 表示对于用户 u_i，文献反馈结果排序正确，即用户偏好的文献 a 排在文献 b 前面；R 表示用户-文献的行为评分矩阵。

(2)概率 $p(e_a > e_b; u_i \mid \theta)$ 由逻辑斯蒂函数(logistic sigmoid function)定义，即：

$$\begin{aligned} p(e_a > e_b; u_i \mid \theta) &= \sigma(\hat{r}(u_i, e_a) - \hat{r}(u_i, e_b)) \\ \sigma(x) &= \frac{1}{1+e^{-x}} \end{aligned} \qquad \text{(公式 4.7)}$$

(3)根据定义（1）和（2），确定目标函数为最小化对数似然函数[64]，即：

$$\min_{\Theta} -\sum_{u_i \in U} \sum_{(e_a > e_b) \in R_i} \ln(\sigma(\hat{r}(u_i, e_a) - \hat{r}(u_i, e_b))) + \frac{\lambda}{2} \| \Theta \|^2 \quad (\frac{\lambda}{2} \| \Theta \|^2 \text{ 为正则项})$$

(公式 4.8)

这里，目标函数中的未知参数可通过随机梯度下降法 SGD 进行求解，即：

$$\frac{\partial}{\partial \Theta} = -\sum_{u_i \in U} \sum_{(e_a > e_b) \in R_i} \frac{e^{-((\hat{r}(u_i, e_a) - \hat{r}(u_i, e_b))}}{1 + e^{-((\hat{r}(u_i, e_a) - \hat{r}(u_i, e_b))}} \frac{\partial}{\partial \Theta} (\hat{r}(u_i, e_a) - \hat{r}(u_i, e_b)) + \lambda \Theta$$

(公式 4.9)

步骤 4：评价指标

对于基于显式评分的推荐模型，平均绝对误差 MAE 和均方根误差 RMSE 是最常用的评价指标，但却并不完全适合基于隐式评分的推荐模型评估。因此，从预测结果的准确性考虑，文中选用召回率（Recall）、准确率（Precision）、F1（F-measure）值和正确率（Accuracy）四种指标对模型的预测结果进行评估。假设 P 表示预测结果集（Prediction Sets），T 表示真实结果集（Test Sets，即测试集）。那么，推荐结果在两个结果集上的列联表可表示为：

表 4.1 推荐结果在结果集（P）和（T）上的分布列联表

结果集	T(y=1)	T(y=0)	所有结果集
P(y=1)	$N(T_{(u,i)} \cap P_{(u,i)} \mid y=1)$	$N(T_{(u,i)} \mid y=0 \cap P_{(u,i)} \mid y=1)$	$N(P_{(u,i)} \mid y=1)$
P(y=0)	$N(T_{(u,i)} \mid y=1 \cap P_{(u,i)} \mid y=0)$	$N(T_{(u,i)} \cap P_{(u,i)} \mid y=0)$	$N(P_{(u,i)} \mid y=0)$
所有结果集	$N(T_{(u,i)} \mid y=1)$	$N(T_{(u,i)} \mid y=0)$	N(T(u,i))or N(P(u,i))

由表 4.1，可以得到预测结果集 P 的召回率（Recall）、准确率（Precision）、F1 值和预测正确率（Accuracy）。这里，除正确率综合评估模型有行为预测（y=1）和无行为预测（y=0）能力之外，其他指标只考虑模型的行为预测（y=1）能力，具体公式如下：

$$\mathrm{Re}call(y=1)=\frac{N(P_{(u,i)} \cap T_{(u,i)} \mid y=1)}{N(T_{(u,i)} \mid y=1)} \quad \text{（公式 4.10）}$$

$$\mathrm{Pr}ecision(y=1)=\frac{N(P_{(u,i)} \cap T_{(u,i)} \mid y=1)}{N(P_{(u,i)} \mid y=1)} \quad \text{（公式 4.11）}$$

$$F1(y=1)=\frac{2\times \mathrm{Pr}ecision(y=1)\times \mathrm{Re}call(y=1)}{\mathrm{Pr}ecision(y=1)+\mathrm{Re}call(y=1)} \quad \text{（公式 4.12）}$$

$$Accuracy=\frac{N(T_{(u,i)} \cap P_{(u,i)} \mid y=1)+N(T_{(u,i)} \cap P_{(u,i)} \mid y=0)}{N(P_{(u,i)})} \quad \text{（公式 4.13）}$$

4.4 实验与讨论

4.4.1 数据集介绍和预处理

(1)数据集介绍

CiteULike 是一个著名的论文书签网站，研究人员可以提交和收藏

自身感兴趣的论文，还可以给论文打标签，以帮助其他科研人员发现他们所感兴趣的论文。CiteULike 提供了公开的用户-论文行为数据集，为科学研究提供了重要的数据支持。本节实验将选取数据集 CiteULike-a[65]，探讨文献推荐领域隐语义模型算法的特点，以及文献推荐算法的推荐效果。数据集包含了用户-论文的交互行为、论文属性（含 ID、题名、摘要等信息）、标签数据（来源于 CiteULike 网站）和引用数据（来源于 Google Scholar），具体信息如表 4.2 所示，反馈频次和对应的用户数量分布情况如图 4.2，从中可以观察到绝大多数用户给予的反馈频次都在 1～30 之间，给予反馈频次较多的用户数量非常少。

表 4.2　CiteULike-a 数据集

数据集	用户数	论文数	评分数	引文数	引文关系（links）	标签数	标注关系（links）	训练集比例 x
CiteULike-a	5551	16980	210537	44709	106398	46391	256233	90%

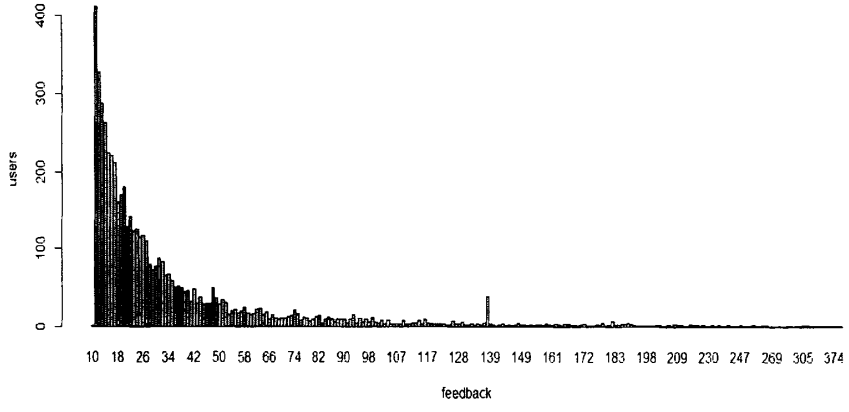


图 4.2 用户对论文反馈行为分布情况

根据 CiteULike-a 数据集所包含的用户和论文信息，抽取两条元路径关系，即论文（P）—标签（T）—论文（P）（标注关系 PTP）、论文（P）—引文（C）—论文（P）（引用关系 PCP），作为推荐模型实现的基础，挖掘两条元路径下论文间的相似度，从而建立用户与论文的语义关联。然后组合两条元路径下用户对论文的偏好特征，实现评分预

测模型，为用户提供论文推荐列表。

(2)数据预处理

数据表示：CiteULike-a 数据集给出了论文间的引用关系和论文标签（忽略频次）数据，故以二值矩阵（“1”或“0”）的方式对这些数据进行描述，反映论文-引用论文和论文-标签的相关关系。

近邻关系矩阵：经二值表示的引用关系和标注关系矩阵，利用 PathSim 相似度计算方法（公式 4.1），分别获得 PTP、PCP 元路径下论文间的相关矩阵，即近邻关系。

负样本生成：本章节主要是基于隐式反馈讨论文献推荐问题，这种数据集的特点是有明确的正样本（用户喜欢什么物品，即有行为），没有负样本（用户对什么物品不感兴趣）。一般地，认为用户没有行为的物品都是用户所不喜欢的，都将其作为负样本。但是，这会出现正负样本比例的严重失调，从而导致因负样本过多而影响推荐效果的问题。因此，在隐式反馈数据集上建立推荐模型，首先需要解决负样本生成问题。

假设 1：用户对论文有行为，表示用户对论文相关的内容、主题等感兴趣；

假设 2：论文间相似度越高，论文的内容、主题等越相近；

假设 3：与用户有过行为的论文相似度较高的论文，但用户却没有行为，更能表明该用户对这篇论文非常不感兴趣。

基于上述假设，通过预先设定正负样本比例的方式，自动选择论文相似度阈值 ξ，生成负样本，即用户 u 有行为的论文 i 和没有行为的论文 j，满足条件 1，则作为用户 u 的负样本。

条件 1： $s(i,j)\geq\xi,\text{且}i\in R_{(u)},\ j\notin R_{(u)}$

本次实验，依据 PCP 和 PTP 元路径下论文的近邻关系，共选取负样本 41,2401 条，正样本 21,0537 条，总样本 62,2938 条，正负样本比例约为 1：2。

偏好矩阵：偏好矩阵反映了用户对论文的兴趣度，而给出的用户-论文行为信息只是简单的二值反馈，数据稀疏性问题严重。因此，根据数据集中给出的用户-论文交互行为，以及选择的负样本，基于论文的近邻关系矩阵，以公式（4.2）重新构建用户-论文在 PCP 和 PTP 元路径下的偏好矩阵。

训练集和测试集：采用十字交叉验证的方法，将总样本随机平均分为 10 份，每次实验抽取其中 1 份作为测试集，剩余 9 份作为训练集，最终将十次交叉实验的结果取平均值作为最终的实验结果。

4.4.2 实验结果分析

实验中涉及的主要模型分为两种：一是基于单元路径关系的隐语义模型，即引用关系隐语义模型（PCP_SVD）和标注关系隐语义模型（PTP_SVD）；二是元路径线性组合的隐语义模型，即基于分类学习（PCP+PTP_LR）和贝叶斯优化排序（PCP+PTP_Bayesian）的元路径组合模型。具体实验和结果如下：

(1)隐特征维度 F 对预测精度的影响

潜在特征因子，反映了用户行为信息映射到低维度空间上用户和文献的类别概率分布情况，是挖掘和表示用户或文献潜在特征的重要因素。实验过程采用控制变量的方式，保证其他超参数一致的条件下，选取不同的隐特征维度 F，在 CiteULike-a 数据集上分别对引用关系（PCP_SVD）、标注关系（PTP_SVD）的隐语义模型，以及两种元路径组合的隐语义模型（PCP+PTP_LR、PCP+PTP_Bayesian）进行有限次数的交叉实验，以评价指标的平均值作为最后的模型性能评估依据。实验结果如表 4.3。

表 4.3 不同隐特征维度 F 的实验结果

模型	PCP				PTP			
F\指标	Recall	Precision	F1	Accuracy	Recall	Precision	F1	Accuracy
20	0.481065	0.552009	0.514098	0.692863	0.533648	0.579709	0.551916	0.711585
40	0.481284	0.55213	0.514275	0.692941	0.531259	0.579807	0.554468	0.711645
60	0.481156	0.552079	0.514179	0.692906	0.531149	0.579618	0.554322	0.711534
80	0.481208	0.552094	0.514216	0.692917	0.531202	0.579723	0.554399	0.711595
100	0.48116	0.552046	0.514168	0.692888	0.53123	0.579749	0.554426	0.711611
300	0.48116	0.5521	0.514191	0.692917	0.531206	0.579743	0.554411	0.711606
500	0.481293	0.552168	0.514297	0.692962	0.531206	0.579743	0.554411	0.711606
模型	PTP+PCP_LR				PCP+PTP_Bayesian			
F\指标	Recall	Precision	F1	Accuracy	Recall	Precision	F1	Accuracy
20	0.225363	0.588046	0.325844	0.685041	0.697585	0.56986	0.627284	0.720016
40	0.225359	0.588092	0.325846	0.68505	0.698731	0.569922	0.627785	0.720156
60	0.225349	0.587928	0.325811	0.685013	0.697704	0.56987	0.627339	0.720033
80	0.225382	0.588036	0.325863	0.685041	0.698184	0.569902	0.627552	0.720096
100	0.225349	0.587927	0.325811	0.685013	0.69778	0.569868	0.627368	0.720038
300	0.225349	0.587986	0.32582	0.685026	0.697809	0.569903	0.627401	0.720066
500	0.22533	0.587966	0.325797	0.68502	0.69798	0.569996	0.627526	0.720147

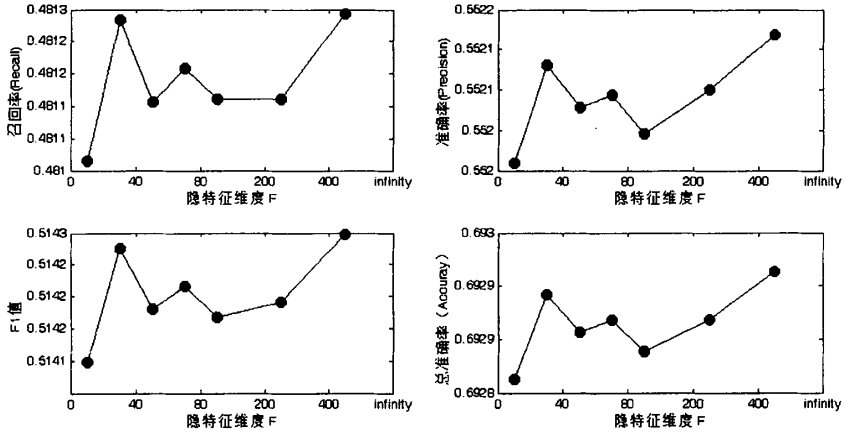


图 4.3 引用元路径下的隐语义模型评估指标随 F 的变化分布

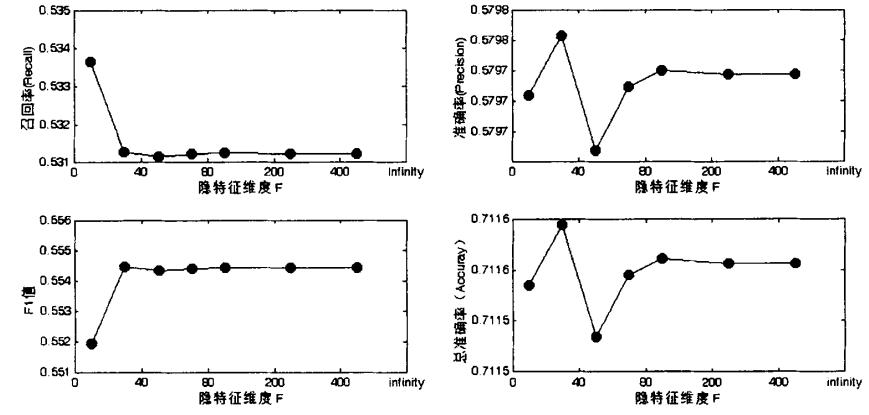


图 4.4 标注元路径下的隐语义模型评估指标随 F 的变化分布

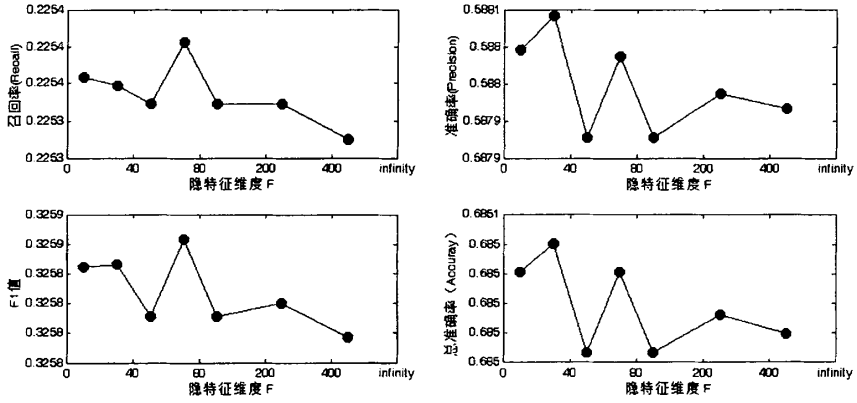


图 4.5 基于分类学习模型评估指标随 F 的变化分布

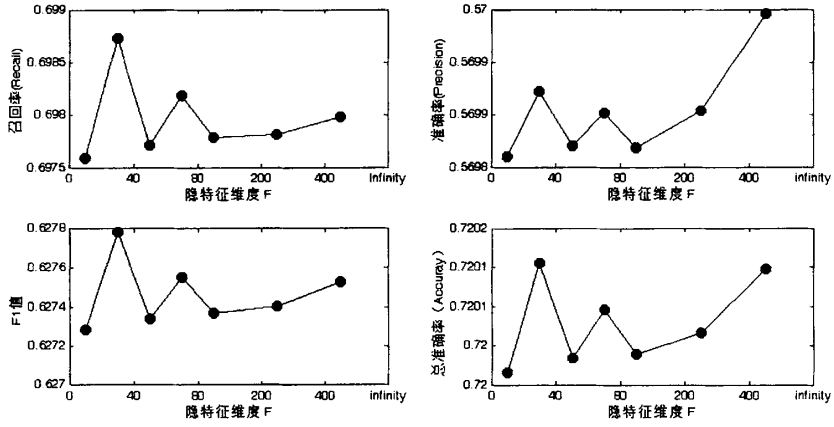


图 4.6 基于贝叶斯优化排序学习模型评估指标随 F 的变化分布

观察表 4.3，四种模型随着隐特征维度 F 的变化，召回率、准确率、F1 值和正确率四种评估指标也随之发生变化，波动的范围大约在 10^{-5}～10^{-3} 数量级之内，浮动比较小。表明，对于隐式反馈推荐模型，以元路径近邻关系发现处理后的用户-文献交互行为，作为推荐模型实现的数据基础，隐特征维度的大小对模型预测效果产生的影响并不大。因此，对于类似推荐模型来说，可以认为在矩阵分解过程中，隐特征维度将不再是影响最终预测结果的重要因素，可以减少对其选择问题的考虑。

图 4.3 至图 4.6，分别描述了四种模型的评估指标随 F 变化的具体变动情况，波动范围虽然很小，但也存在细微差异。如图 4.3、4.5 和 4.6，模型 PCP_SVD、PTP_SVD 和 PCP+PTP_Bayesian 在隐特征维度 F 不断变化的过程中，召回率、准确率、F1 值和正确率指标也随之波动，但总体变化趋势相一致。但图 4.4，模型 PTP_SVD 在随 F 值变化的情况下，召回率和准确率呈现反比关系，召回率降低，准确率有所提高，这也与推荐系统中召回率和准确率不能两全的事实相吻合，但在其他模型中表现却并不明显。模型 PTP 随着 F 达到某一维度之后，所有评估指标的变化曲线都接近一条水平直线，保持相对平稳。与其他模型曲线的波动幅度相比，可以认为该模型受隐特征维度 F 的影响最小。

此外，比较图 4.5 和 4.6 发现，基于分类学习的元路径组合模型和基于贝叶斯优化排序的元路径组合模型，两者在隐特征维度 F 较小时，评估指标变化几乎一致。但在 F 较大时，前者略有下降，而后者则持续上升。同时，在实验过程中发现，隐特征维度 F 对元路径参数权值 θ 的分配影响非常小，最终影响推荐模型预测效果的因素仍与选取的元路径语义关系密切相关。

(2)不同模型推荐效果的比较

经十次交叉实验，分别得到四种模型的平均评估指标，如表 4.4，图 4.7 分别展示了四种模型评估指标的比较情况。

表 4.4 不同模型的预测结果分布

模型 / 指标	PCP_SVD	PTP_SVD	PCP+PTP_LR	PCP+PTP_Bayesian
Recall(y=1)	0.48119	0.531557	0.225355	0.697968
Precision(y=1)	0.552089	0.579727	0.587997	0.569903
F1(y=1)	0.514203	0.554051	0.325828	0.685029
Accuracy	0.692914	0.711597	0.627465	0.720079

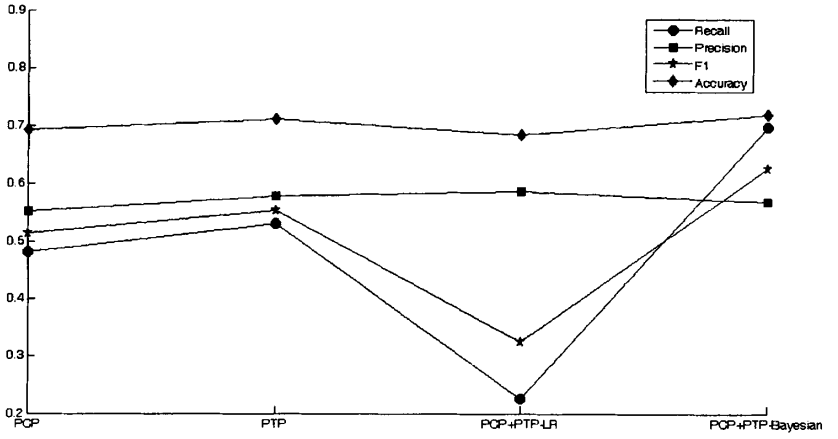


图 4.7 四种模型不同指标分布

从召回率角度，它描述的是有多少比例的用户-论文行为（评分）记录包含在最终的推荐列表中，是评估模型推荐结果查全率的指标。从结果来看，基于分类学习的组合模型 PCP+PTP_LR 召回率最低，约为 22.54%。而基于贝叶斯优化排序的组合模型 PCP+PTP_Bayesian 却得到较优的召回率，约为 69.80%。表明，利用排序方式得到的元路径组合模型能够较全面地发现用户潜在感兴趣的文献，为用户提供其可能感兴趣的文献列表。

从准确率角度，它描述的是最终推荐列表中有多少比例是发生过的用户-文献行为（评分）记录。比较图 4.7 中变化曲线，发现结果恰恰与召回率相反，模型 PCP+PTP_LR 的准确率最高，约为 58.80%，要优于其他模型，而标签元路径下的模型 PTP_SVD 次之，约为 57.97%，高于引用元路径下的模型 PCP_SVD。一方面说明模型 PCP+PTP_LR 提供的推荐列表中，绝大多数都是用户真正感兴趣的文献。反之，其他模型的推荐列表相比可能包含了较多的错误推荐，并不是用户所感兴趣的。另一方面，单元路径下的隐语义模型，考虑标签关系的推荐模型更能得到较好的预测准确度。

从模型的总体性能角度，F1 值是召回率和准确率的调和平均数，是评估推荐模型提供的推荐列表查全性和准确性的重要指标。而正确率是评估推荐模型有行为预测（y=1）和无行为预测（y=0）准确性的综合性指标。由图 4.7 可知，基于分类学习的组合模型 PCP+PTP_LR，F1 值和召回率明显低于其他模型，但正确率却相差较少，这从侧面反

映了该模型对用户无行为预测（y=0）的能力比较好，提高了模型的正确率指标。

对于两种组合模型，其中基于贝叶斯优化排序的组合模型PCP+PTP_Bayesian，无论在F1值，还是正确率，都要优于基于分类学习的组合模型 PCP+PTP_LR。由此可以得出，采用贝叶斯优化排序学习算法得到的推荐模型更适合元路径组合模型参数的学习，能够取得较好的推荐效果。此外，模型 PCP+PTP_Bayesian 的 F1 值和正确率优于任一单元路径下的隐语义模型，那么，可以认为综合考虑多维度的语义关系，能够很好地挖掘用户-文献间潜在的偏好特征，有利于生成较好的推荐结果，提高模型的预测精度。

4.5 本章小结

本章讨论了文献推荐领域，基于异构信息网络结构，以文献间近邻关系和用户隐式反馈行为建立的推荐模型，在文献推荐中的作用以及取得的推荐效果。

首先结合文献特征，给出了异构信息网络结构和元路径的定义，以及根据元路径获得文献间相似度的计算方法；然后，对文献推荐模型的具体工作原理和操作步骤进行了详细描述，包括元路径下用户偏好矩阵的构建、潜在因子特征的生成、组合模型参数的学习算法和评价指标等；最后，以数据集 CiteULike-a 为实验对象，分别探讨了单元路径下的隐语义模型和元路径组合模型受隐特征维度的影响情况，以及不同算法模型的性能，证明了考虑异构信息网络结构的隐语义模型在文献推荐中的可行性。并且，综合多维度的语义关系能够在基于隐式反馈数据的推荐模型中取得较好预测效果。

5 总结与展望

5.1 本文工作总结

面对互联网资源的爆炸式增长、信息技术的迅猛发展和电子商务的快速普及，个性化推荐系统成为重要的信息过滤手段之一，它的目标是为用户提供个性化的信息推荐服务，而实现这一目标的核心就是推荐算法。目前，推荐系统领域常见的算法主要分为三类，即基于内容的推荐、协同过滤推荐和混合推荐。其中，协同过滤推荐技术应用最广泛，但仍面临一些关键问题。

本文属于推荐算法研究，选取的研究对象是隐语义模型，主要工作内容总结如下：

首先，本文简明介绍了推荐系统领域的经典算法，重点讨论了文献推荐领域常见算法的研究现状和研究意义。并针对隐语义模型在推荐系统中的应用和发展情况做了系统调研，充分分析和讨论了隐语义模型的主要研究内容以及存在的问题，为后期研究工作的展开提供了基础。

其次，为了进一步理解和掌握隐语义模型，文章从算法的工作原理、模型实现、参数学习和评价指标等角度，全面描述隐语义模型在推荐系统中的应用机制。此外，分别介绍了隐语义模型的几种典型算法的核心思想和改进办法，包括 Base-SVD、Bias-SVD、SVD++和 Asymmetric-SVD，并通过真实数据集 MovieLens 验证算法的预测效果，比较分析几种模型的性能。

再次，考虑到文献推荐领域用户-文献评分难获取、数据稀疏性等问题，提出将隐语义模型与异构信息网络结构处理相融合的算法思路，不仅考虑用户与文献间的关联，还考虑用户间或文献间丰富的语义关系，以此发现用户、文献间潜在的关联特征，作为推荐模型实现的基础。文中主要采用元路径描述多维语义关系，重新构建用户-文献间的交互行为（即评分矩阵）；通过隐语义模型，得到用户、文献在不同元路径下的潜在特征因子分布。以线性组合方式综合多维度语义关系，实现适合文献推荐领域的评分预测模型。

最后，在文献推荐算法设计的基础上，针对元路径权值分配问题，提出基于二分类和贝叶斯优化排序两种参数学习方案，通过在 CiteULike-a 文献行为数据集上的反复实验，验证算法的可实施性和有效性。结果表明，以异构信息网络结构考虑用户或文献间的语义关联，

能够很好地解决隐式评分数据的稀疏性问题，并且能够有效地发现用户或文献的潜在特征分布以及两者间的潜在语义关联，有利于提高推荐模型的预测效果。

综合上述对本文工作内容的总结，在隐语义模型算法研究部分，主要体现出以下几个创新点：

(1)提出基于异构信息网络的文献推荐模型。在隐式反馈行为数据的基础上，综合用户、文献、评分、近邻关系等多维度信息，以元路径发现用户、文献间的语义关联，重新建立用户-文献交互关系，缓解数据稀疏性问题。并且，融合隐语义模型挖掘用户、文献的潜在特征因子分布，最终实现文献评分预测模型。

(2)在文献推荐模型算法实现过程中，针对多条元路径权值分配问题，提出运用二分类和贝叶斯优化排序两种参数学习算法，求解得到不同元路径的权值。并通过真实数据集的反复实验，比较分析单元路径和元路径组合模型的推荐效果，讨论隐特征维度对不同推荐模型预测效果的影响。

5.2 工作展望

文中基于隐语义模型算法的研究和实验，提出的考虑异构信息网络结构的文献推荐模型，虽然取得了一定的研究成果，但这只是我们在文献推荐领域应用的初步探索。对于参数调优、算法时间效率等问题，文中讨论的还比较少，仍存在许多不足和需要补充的地方。在后续工作中，我们拟从以下几个方面做进一步深入的探讨和研究：

(1)参数的选择：隐语义模型在矩阵分解过程中涉及隐特征维度、学习速率、正则项系数、迭代次数，以及考虑异构信息网络的文献推荐模型中，元路径权值学习算法的超参数等，这些参数都会对推荐模型的执行结果产生不同程度的影响。由此，可以认为参数的选择是模型实现过程中的关键问题。当前，本文仅通过有限次数的实验来选择参数，最终选取的结果可能并不是最优的。因此，后期工作中，拟从参数调优方法设计的角度，研究参数的自动选择问题，进而提高模型的预测效果。

(2)元路径选择问题：在建立异构信息网络结构的文献推荐模型过程中，存在多种元路径关系，而不同的元路径反映了文献间不同的语义关联，会给模型带来不同的预测结果。因此，如何选择有效的元路径语义关系或者选择多少条元路径语义关系作为最终推荐模型的实现基础，需要我们后期尝试多组实验，根据实验结果做进一步深入分析。

(3)冷启动问题：推荐系统领域，冷启动问题一直未能得到很好地解决。本文采用异构信息网络结构，发现文献间的近邻关系，虽然在一定程度上缓解了评分数据的稀疏性问题，但并没有解决冷启动问题。如何利用异构信息网络或其他资源为新用户进行推荐，以及将新文献如何有效地推荐给用户，这将是后期工作中值得关注和探讨的地方。

(4)计算效率：目前，文中对隐语义模型算法实现方法的设计中，大规模的数据样本需要通过梯度下降法进行反复迭代，计算量大，时间长，故而效率比较低。对于算法计算效率的问题，后期我们将考虑选用一些方法予以解决，例如考虑并行计算，在梯度下降法的过程中实现并行优化等，进而能够提高推荐模型的计算效率。

当然，文中对推荐系统中隐语义模型算法的研究部分，除了上述提到的几点内容之外，仍存在许多值得探讨和研究的地方，需要逐渐补充和完善。希望我们能够通过后续工作的推进和深入探讨，取得更好的研究成果。

参考文献

[1]SUN J T,WANG X H,SHEN D,et al. Mining Clickthrough Data for Collaborative Web Search[C]. In: WWW 2006:Proceedings of the 15th International Conference on World Wide Web,ACM,New York,USA,2006.
[2]SUN J T,ZENG H J,LIU H,et al. CubeSVD: a Novel Approach to Personalized Web Search[C]. Proceedings of the 14th International Conference on World Wide Web,ACM,New York,USA,2005:382-390.
[3]GEDIMINAS A,ALEXANDER T. Toward the Next Generation of Recommender Systems:A Survey of the State-of-the-Art and Possible Extensions[J]. IEEE Transactions on Knowledge and Data Engineering,2005,17(6):634-749.
[4]ROBERT M. BELL,Y K,CHRIS V. The BellKor solution to the Netflix Prize. Tech.Rep(2007):1-15.
[5]ROBERT M. BELL,Y K,CHRIS V. The BellKor 2008 Solution to the Netflix Prize[C]. Tech. Rep(2008):1-21.
[6]YEHUDA K. The BellKor Solution to the Netflix Grand Prize[C]. Tech. Rep(2009):1-52.
[7]刘建国,周涛,汪秉宏. 个性化推荐系统研究进展[J]. 自然科学进展,2009,19(1):1-15.
[8]YEHUDA K. Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model[C]. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,ACM,2008:426-434.
[9]刘韵毅,梁標. 基于用户偏好的文献推荐系统[J]. 情报理论与实践,2007,30(1):61-63,25.
[10]PAOLO M,PAOLO A Trust-aware Recommender Systems[C]. In Proceedings of the 1st ACM Conference on Recommender Systems (RecSys,07),2007:17-24.
[11]XIAO Y,XIANG R,Y. SUN,et al. Recommendation in Heterogeneous Information Networks with Implicit User Feedback[C]. Proceedings of ACM Conference on Recommender Systems,2013:347-350.
[12]姚远. 基于论文引用网络的文献推荐算法研究[D]. 北京:北京交通大学,2014:34-57.
[13]CHEN C C,CHEN A P. Using Data Mining Technology to Provide a Recommendation Service in the Digital Library[J]. Electronic Library:

Library and Information Studies. 2007,25(6):711-724.
[14]陈祖琴,张慧玲,葛继科等. 基于加权关联规则挖掘的相关文献推荐[J]. 现代图书情报技术,2007(10):57-61.
[15]MIDDLETON S E,SHADBOLT N R,DE R OURE D C. Ontological User Profiling in Recommender Systems[J]. ACM Transactions on Information Systems,2004,22 (1):54-88.
[16]LIAO I E,LIAO S C,KAO K F,et al. A Personal Ontology Model for Library Recommendation System[J]. Digital Libraries:Achievements, Challenges and Opportunities,2006, 4312:173-182.
[17]LIAO S C,KAO K F,LIAO I E,et al. Pore: a Personal Ontology Recommender System for Digital Libraries[J]. Electronic Library: Library and Information Studies. 2009,27(3):496-508.
[18]LIAO I-EN,HSU W C,CHENG M S,et al. A Library Recommender System based on a Personal Ontology Model and Collaborative Filtering Technique for English Collections[J]. Electronic Library: Library and Information Studies,2010, 28(3):386-400.
[19]徐勇,司凤山,吴延辉等. 基于概念泛化的科技文献推荐算法[J]. 图书情报工作,2012,56(21):101-108.
[20]黄泽明. 基于主题模型的学术论文推荐系统研究[D]. 大连:大连海事大学,2013:31-58.
[21] GOODRUM A. Scholarly Publishing in the Internet Age: a Citation Analysis of Computer Science Literature[J]. Information Processing & Management,2001,37(5): 661-675.
[22]MCNEE S M,ALBERT I,Cosley D,et al. On the Recommending of Citations for Research Papers[C]. Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work,NY,USA,2002:116-125.
[23]陈祖琴. 基于数据挖掘的引文分析——利用模拟日志分析进行相关文献推荐[D]. 重庆:西南大学,2008:15-50.
[24]GIPP B,BEEL J,HENTSCHEL C. Scienstein: a Research Paper Recommender System[C]. Proceedings of the International Conference on Emerging Trends in Computing. Virudhunagar,India,IEEE,2009:309-315.
[25]BASU C,HIRSH H,COHEN W W,et al. Technical Paper Recommendation: a Study in Combining Multiple Information Sources[J]. Artificial Intelligence Research,2001,14(1):231-252.
[26]尉萌. 利用演化模式做文献推荐术[J]. 现代图书情报技

术, 2014(4):20-26.
[27]李琳娜, 张志平, 乔晓东等. 基于文献共被引关系的协同过滤文献推荐系统[J]. 数字图书馆论坛, 2012(3):33-37.
[28]YEHUDA K. Factor in the Neighbors: Scalable and Accurate Collaborative Filtering[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2010,4(1):1-17.
[29]XIN L,YUANXIN O Y,ZHANG X. Improving Latent Factor Model based Collaborative Filtering via Integrated Folksonomy Factors[J]. International Journal of Uncertainty,Fuzziness and Knowledge-Based Systems,2011,19(2):307-327.
[30]THOMAS H. Latent Semantic Models for Collaborative Filtering[J]. ACM Transactions on Information Systems,2004,22(1):89-115.
[31]WANG Q,XU J,LI H,et al. Regularized Latent Semantic Indexing[C]. Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval ACM,2011,79(3):685-694.
[32]DAVID M B,ANDREW Y N,MICHAEL I J. Latent Dirichlet Allocation[J]. Journal of Machine Learning Research,2003(3):993-1022.
[33]YEHUDA K,ROBERT B,CHRIS V. Matrix Factorization Techniques for Recommender System[J]. IEEE Computer Society,2009,42(8):42-49.
[34]SARWAR B,KARYPIS G,KONSTAN J,et al. Application of Dimensionality Reduction in Recommender System-A Case Study[C]. Proceeding KDD Workshop on Web Mining for e-Commerce: Challenges and Opportunities (WebKDD),ACM Press,2000:1-14.
[35]SOUMEN C. Mining the Web : Discovering Knowledge from Hypertext Data[M]. Science and Technology Books,2002:65.
[36]MUKNND D F,GEORGE K. Item-based Top-N Recommendation Algorithms[J]. ACM Transactions on Information Systems (TOIS),2004, 22(1):143-177.
[37]ARKADIUSZ P. Improving Regularized Singular Value Decomposition for Collaborative Filtering[C]. Proceeding KDD Cup and Workshop,ACM Press,2007:39-42.
[38]YUNHONG Z,DENNIS W,ROBERT S,et al. Large-Scale Parallel Collaborative Filtering for the Netflix Prize[C]. Proceeding 4th Intel Conf. Algorithmic Aspects in Information and Management,LNCS 5034,Springer,2008:337-348.
[39]HU Y F,KOREN Y,VOLINSKY C. Collaborative Filtering for Implicit

Feedback Datasets[C]. Proceeding IEEE International Conference on Data Mining,IEEE CS Press,2008:263-272.
[40]ROBERT M. BELL,Y K. Scalable Collaborative with Jointly Derived Neighborhood Interpolation Weights[C]. Seventh IEEE International Conference on Data Mining,2007:43-52.
[41]YEHUDA K. Collaborative Filtering with Temporal Dynamics[J]. Communications of the ACM,2010,53(4):89-97.
[42]STEEN R,CHRISTOPH F,ZENO G,et al. BPR:Bayesian Personalized Ranking from Implicit Feedback[J]. In UAI2009,2009:452-461.
[43]ANDRIY M. Taxonomy-Informed Latent Factor Models for Implicit Feedback[C]. Proceedings of KDD-Cup 2011 competition,2012:169-181.
[44]GUO L,Ma J,CHEN Z,ZHONG H. Learning to Recommend with Social Contextual Information from Implicit Feedback[J]. Soft Computing,2015,19(5):1351-1362.
[45]YAO W,HE J,HUANG G,CAO J. A Graph-based Model for Context-aware Recommendation Using Implicit Feedback Data[J]. World Wide Web,2015,18(5):1351-1371.
[46]MELVILLE P,MOONEY R J,NAGARAJAN R. Content-boosted Collaborative Filtering for Improved Recommendation[C]. In Proceedings of the 18th National Conference on Artificial Intelligence,2002:187-192.
[47]PETER F,MU Z. Content-boosted Matrix Factorization for Recommender Systems: Experiments with Recipe Recommendation[C]. In Proceedings of the 5th ACM conference on Recommender systems,Chicago,Illinois,USA,2011:261-264.
[48]AHMED A,BHARGAV K,Sandeep Pandey. Latent Factor Models with Additive and Hierarchically-smoothed User Preferences[C]. Proceedings of 6th ACM International Conference on Web Search and Data Mining (WSDM),2013.
[49]ZHANG C,ZHAO X,WANG K,SUN J. Content+Attributes: A Latent Factor Model for Recommending Scientific Papers in Heterogeneous Academic Networks[C]. 36th European Conference on IR Research, Amsterdam,The Netherlands,2014:39-50.
[50]ANDREAS T,MICHAEL J. The BigChaos Solution to the Netflix Prize 2008[C].Tech.rep,Neuer Weg 23,A-8580 Koflach,Austria,2008:1-17.
[51]RUSLAN S,ANDRIY M,GEOFFREY H. Restricted Boltzmann Machines for Collaborative Filtering[C]. Proceedings 24th Annual

International Conference on Machine Learning,2007:791-79.
[52]BENJAMIN M. Collaborative Filtering: a Machine Learning Perspective[D]. Toronto:University of Toronto,2004:31-126.
[53]鲁权. 基于协同过滤模型与隐语义模型的推荐系统研究与实现[D]. 长沙:湖南大学,2013:20-45.
[54]鲁权,王如龙,张锦等. 融合领域模型与隐语义模型的推荐算法[J]. 计算机工程与应用,2013,49(19):100-103,134.
[55]冯晓龙. 基于用户行为分析的 P2P 流媒体推荐系统研究[D]. 北京:北京交通大学,2013:15-51.
[56]张川. 基于矩阵分解的协同过滤推荐算法研究[D]. 长春:吉林大学,2013:13-56.
[57]段华杰. 考虑时间效应的矩阵分解技术在推荐系统中的应用[J]. 微型电脑应用,2013,29(3):53-64.
[58]顾晔,吕红兵. 改进增量奇异值分解协同过滤算法[J]. 计算机工程与应用,2011,47(11):152-154.
[59]罗铁坚,程福兴,周佳. 融合奇异值分解和动态转移链的学术资源推荐模型[J]. 中国科学院大学学报,2014,31(2):257-266.
[60]JONATHAN L H,JOSEPH A K,LOREN G T,et al. Evaluating Collaborative Filtering Recommender Systems[J]. ACM Transactions on Information Systems,2004,22(1):5-53.
[61]SUN Y,HAN J,ZHAO P,et al. RankClus: Integrating Clustering with Ranking for Heterogeneous Information Network Analysis[C]. Proceedings of the 12th International Conference on Extending Data Base Technology,2009(3):565-576.
[62]冒九妹. 基于异构信息网络的协同过滤推荐技术研究[D]. 苏州:苏州大学,2014:16-38.
[63]SUN Y,HAN J,YAN X,et al. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks[J]. Proceedings of the VLDB Endowment,2011,4(11):992-1003.
[64]SUN Y,HAN J. Meta-Path-Based Search and Mining in Heterogeneous Information Networks[J]. TsingHua Science and Technology,2013,18(4):329-338.
[65]HAO W,BINYI C,LI W J. Collaborative Topic Regression with Social Regularization for Tag Recommendation[C]. Proceedings of the Twenty-third International Joint Conference on Artificial Intelligence (IJCAI),2013:2719-2725.

附录 A Logistic（LR）回归分类函数推导

Logistic 回归实际上是一种分类方法，主要用于两分类问题，利用 Logistic 函数（或称为 Sigmoid 函数），自变量取值范围为(-INF, INF)，自变量的取值范围为(0, 1)，函数形式为：

$$\sigma(z)=\frac{1}{1+e^{-z}}(sig\bmod 函数)$$

$$h_\theta(x)=\sigma(\theta^T x)=\frac{1}{1+e^{-\theta^T x}}$$

假设观察的评分数据 y=1 和 y=0 的概率分别为：

$$P(y=1\mid x;\theta)=h_\theta(x)$$
$$P(y=0\mid x;\theta)=1-h_\theta(x)$$

由假设统计样本的均匀分布特点可知，若 $h_\theta(x)>=0.5$ 则样本属于 1 类别，反之，样本属于 0 类别。若有 m 个观测样本的评分数据，且每个样本分布相对独立，那么，可得 1 和 0 类别的联合分布概率：

$$L(\theta)=\prod_{i=1}^{m}P(y^{(i)}\mid x^{(i)};\theta)=\prod_{i=1}^{m}(h_\theta(x^{(i)}))^{y^{(i)}}(1-h_\theta(x^{(i)}))^{1-y^{(i)}}$$

上式为 m 个观测样本的似然函数，我们的目标就是能够求出使得似然函数最大的参数估计，因此对上述函数求对数可得：

$$L(\theta)=\log L(\theta)=\sum_{i=1}^{m}[y^{(i)}\log h_\theta(x^{(i)}))+(1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

为了便于采用梯度下降法学习目标函数，我们将其转为最小化问题，即目标损失函数为：

$$J(\theta)=-\frac{1}{m}L(\theta)=-\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)}))+(1-y^{(i)})\log(1-h_\theta(x^{(i)}))]+\frac{\lambda}{2\mathrm{m}}\sum_{j=1}^{l}\theta_j^{\,2}$$

（$\frac{\lambda}{2\mathrm{m}}\sum_{j=1}^{l}\theta_j^{\,2}$ 为正则项）

附录 B 隐语义模型典型算法 Java 程序（部分）

```
//模型Base-SVD
输入：
    用户评分矩阵 r;
    用户、物品列表 user、items;
    隐特征个数 F;
    算法学习速率 gamma;
    迭代次数 iternum;
    因子正则项系数lambda_pu、lambda_qi;
初始化赋值：
    void initialize() {
        for(String u : users) {
            Double[] vec = new Double[F];
            for(int f=0; f<getF(); f++)
                vec[f] = (Math.random()*0.1)/Math.sqrt(getF());
            pu.put(u, vec);
        }//对用户因子矩阵初始化赋值
        System.out.println("Length of pu: "+pu.size());
        for(String i : items) {
            Double[] vec = new Double[F];
            for(int f=0; f<getF(); f++)
                vec[f] = (Math.random()*0.1)/Math.sqrt(getF());
            qi.put(i, vec);
        }/对物品因子矩阵初始化赋值
        System.out.println("Length of qi: "+qi.size());
        mu = Util.computeMu(r); // 全局平均数}
    //预测评分
    public Double r_hat(String u, String i) {
        Double bu_u = 0.0, bi_i = 0.0, pu_qi=0.0;
        if ( bu.containsKey(u) ) bu_u = bu.get(u);
        if ( bi.containsKey(i) ) bi_i = bi.get(i);
        if ( pu.containsKey(u) && qi.containsKey(i) ) //如果包含pu和qi
参数值
```

```
            pu_qi = pu*qi^T; //计算用户因子和物品因子的内积，可调用
Util.vecvecprod()方法
        else
            pu_qi = mu;//否则返回全局平均数
        return pu_qi;
    }
    //计算实际评分与预测评分的差值
    Double e(String u, String i) {
        return r.get(u).get(i) - r_hat(u,i);
    }

输出：模型参数学习
    void computeBuBiPuQi() {
        for(int iter=0; iter< iternum; iter++) {//repeat：每一次迭代
            for(String u : r.keySet()) {//repeat：每一位用户
                for(String i : r.get(u).keySet()) {//repeat：用户有过行为的
每一物品
                    Double e_ui = e(u,i);//调用方法e，得到预测残差

                    Double[] p_u = pu.get(u);
                    Double[] q_i = qi.get(i);
                    p_u=p_u+ gamma*(e_ui*q_i-lambda_pu*p_u);
                        q_i= q_i +gamma*(e_ui*p_u-lambda_qi*q_i);//
可调用方法Util.scalarvecprod()和Util.vecvecsum()
                }
            }
            gamma= gamma*0.9;//变化学习速率
        }
}

//模型Bias-SVD
输入：
    用户评分矩阵 r;
    用户、物品列表 user、items;
    隐特征个数 F;
    算法学习速率 gamma;
```

```
    迭代次数 iternum;
    正则项系数 lambda_pu、lambda_qi、lambda_bu、lambda_bi;
初始化赋值:
    void initialize() {
        for(String u : users) bu.put(u, 0.0);//初始化用户偏置项bu
        for(String i : items) bi.put(i, 0.0);//初始化物品偏置项bi
        for(String u : users) {
            Double[] vec = new Double[F];
            for(int f=0; f<getF(); f++)
                vec[f] = (Math.random()*0.1)/Math.sqrt(getF());
            pu.put(u, vec);
        }//对用户因子矩阵初始化赋值

        for(String i : items) {
            Double[] vec = new Double[F];
            for(int f=0; f<getF(); f++)
                vec[f] =(Math.random()*0.1)/Math.sqrt(getF());
            qi.put(i, vec);
        }//对物品因子矩阵初始化赋值
    }

//预测评分
    public Double r_hat(String u, String i) {
        Double bu_u = 0.0, bi_i = 0.0, pu_qi=0.0;
        if ( bu.containsKey(u) ) bu_u = bu.get(u);
        if ( bi.containsKey(i) ) bi_i = bi.get(i);
        if ( pu.containsKey(u) && qi.containsKey(i) )
            pu_qi = pu*qi^T;
        else
            pu_qi = mu;
        return mu+bu_u+bi_i+pu_qi;//引入偏置项的预测评分计算公式
    }

输出: 模型参数学习
    void computeBuBiPuQi() {
        for(int iter=0; iter<this.iternum; iter++) {
```

```
            for(String u : r.keySet()) {
                for(String i : r.get(u).keySet()) {
                    Double e_ui = e(u,i);
                    Double b_u = bu.get(u);
                    Double b_i = bi.get(i);
                    Double[] p_u = pu.get(u);
                    Double[] q_i = qi.get(i);
                    b_u = b_u + gamma*(e_ui-lambda_bu*b_u);
                    b_i= b_i + gamma*(e_ui-lambda_bi*b_i);
                    p_u= p_u +gamma*(e_ui*q_i-lambda_pu*p_u);
                    q_i= q_i +gamma*(e_ui*p_u-lambda_qi*q_i);
                }
            }
          gamma= gamma*0.9;
        }
    }

//模型SVD++
输入：
    用户评分矩阵 r;
    用户、物品列表 user、items;
    隐特征个数 F;
    算法学习速率 gamma;
    迭代次数 iternum;
    正则项系数 lambda_pu、lambda_qi、lambda_bu、lambda_bi、
lambda_yj;

初始化赋值：
    void initialize() {
        for(String u : users) bu.put(u, 0.0);
        for(String i : items) bi.put(i, 0.0);
        for(String u : users) {
            Double[] vec = new Double[F];
            for(int f=0; f<getF(); f++)
                vec[f] = (Math.random()*0.1)/Math.sqrt(getF());
            pu.put(u, vec);
```

```
        }
        for(String i : items) {
            Double[] vec = new Double[F];
            for(int f=0; f<getF(); f++)
                vec[f] = (Math.random()*0.1)/Math.sqrt(getF());
            qi.put(i, vec);
        }
        //初始化yj
        for(String i: items)
        {
            Double[] impfed=new Double[F];
            for(int f=0;f<getF();f++)
            {
                impfed[f]=(Math.random()*0.1)/Math.sqrt(getF());
            }
            yj.put(i, impfed);
        }
//预测评分
    public Double r_hat(String u, String i) {
        Double bu_u = 0.0, bi_i = 0.0, pu_qi=0.0;
        //Baseline
        if ( bu.containsKey(u) ) bu_u = bu.get(u);
        if ( bi.containsKey(i) ) bi_i = bi.get(i);
        //implicit feedback factors
        if ( pu.containsKey(u) && qi.containsKey(i) )
            pu_qi = q_i^T (p_u + |N(u)|^{-1/2} \sum_{j \in N(u)} y_j );
        return mu+bu_u+bi_i+pu_qi;
    }

输出：模型参数学习
    void computeBuBiPuQi() {
        for(int iter=0; iter<this.iternum; iter++) {
            for(String u : r.keySet()) {
                for(String i : r.get(u).keySet()) {
```

```
                    Double e_ui = e(u,i);
                    Double b_u = bu.get(u);
                    Double b_i = bi.get(i);
                    Double[] p_u = pu.get(u);
                    Double[] q_i = qi.get(i);
                    Double[] y_j=yj.get(i);
                    b_u = b_u + gamma*(e_ui-lambda_bu *b_u);
                    b_i = b_i + gamma*(e_ui-lambda_bi *b_i);
                    p_u = p_u+gamma*(e_ui*q_i-lambda_pu*p_u);


q_i=q_i+gamma*(e_ui* p_u+|N(u)|^(-1/2) ∑_{j∈N(u)} y_j)-lambda_qi*q_i);

                    y_j= y_j +gamma*(e_ui*|N(u)|^(-1/2) *q_i-lambda_yj*y_j);

            }
            gamma= gamma*0.9;
        }
    }
```

//模型Asymmetric-SVD

输入：

用户评分矩阵 r；

用户、物品列表 user、items；

隐特征个数 F；

算法学习速率 gamma；

迭代次数 iternum；

正则项系数 lambda_pu、lambda_qi、lambda_bu、lambda_bi、lambda_xj、lambda_yj；

初始化赋值：

```
    void initialize() {
        for(String u : users) bu.put(u, 0.0);
        for(String i : items) bi.put(i, 0.0);
        for(String i : items) {
```

```
            Double[] vec = new Double[getF()];
            for(int f=0; f<getF(); f++)
                vec[f] = (Math.random()*0.1)/Math.sqrt(getF());
            qi.put(i, vec);
        }
        //初始化显性反馈权值xj
        for(String i: items)
        {
            Double[] impfed=new Double[getF()];
            for(int f=0;f<getF();f++)
            {
                impfed[f]=(Math.random()*0.1)/Math.sqrt(getF());
            }
            xj.put(i, impfed);
        }
        //初始化隐式反馈权值yj
        for(String i: items)
        {
            Double[] impfed=new Double[getF()];
            for(int f=0;f<getF();f++)
            {
                impfed[f]=(Math.random()*0.1)/Math.sqrt(getF());
            }
            yj.put(i, impfed);
        }
    }

//预测评分
    public Double r_hat(String u, String i) {
        Double bu_u = 0.0, bi_i = 0.0, pu_qi=0.0;
        //Baseline
        if ( bu.containsKey(u) ) bu_u = bu.get(u);
        if ( bi.containsKey(i) ) bi_i = bi.get(i);
        //implicit feedback factors
        if (qi.containsKey(i) )
```

$$pu\_qi = q_i^T(|R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_{uj})x_j + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)$$

```
        return mu+bu_u+bi_i+pu_qi;
    }
输出：模型参数学习
    void computeBuBiPuQi() {
        for(int iter=0; iter<this.iternum; iter++) {
                for(String u : r.keySet()) {
                for(String i : r.get(u).keySet()) {
                    Double e_ui = e(u,i);
                    Double b_u = bu.get(u);
                    Double b_i = bi.get(i);
                    Double[] q_i = qi.get(i);
                    Double[] y_j = yj.get(i);
                    Double[] x_j = xj.get(i);
                    b_u = b_u + gamma*(e_ui-lambda_bu *b_u);
                    b_i = b_i + gamma*(e_ui-lambda_bi *b_i);
```

$$q\_i = q\_i + gamma*(e\_ui*(|R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_{uj})x\_j + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y\_j) - lambda\_qi*q\_i);$$

```
            y_j= y_j +gamma*(e_ui*|N(u)|^(-1/2)*q_i-lambda_yj*y_j);
            x_j=x_j+gamma*(e_ui*|R(u)|^(-1/2)*(r_uj-b_uj)*q_i-lambda_xj*x_j);
            }
            gamma= gamma*0.9;
        }
    }
```

附录C 算法调用的主要外部程序代码

```
//外部程序Util
package LFMjava;
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.*;

public class Util {
    //输出向量值
    public static void PrintVector(Double[] v) {
        for(int j=0; j<v.length; j++)
            System.out.print(v[j] + " ");
        System.out.print("\n");
    }

    //实现两个数组在f维上的乘积，对应f相乘
    public static Double vecvecprod(Double[] p, Double[] q) {
        Double pq = 0.0;
        for(int f=0; f<p.length; f++)
            pq += p[f]*q[f];
        return pq;
    }

    //实现数组元素乘以常数a，对应f个元素
    public static Double[] scalarvecprod(Double a, Double[] p) {
        Double[] q = new Double[p.length];
        for(int f=0; f<p.length; f++)
        {
            q[f] = a*p[f];
        }
        return q;
    }

    //实现两个数组在f维上的求和，对应f相加
```

```
    public static Double[] vecvecsum(Double[] p, Double[] q) {
        Double[] p_plus_q = new Double[p.length];
        for(int f=0; f<p.length; f++)
            p_plus_q[f] = q[f] + p[f];
        return p_plus_q;
    }

    //实现两个数组在f维上的求差，对应f相减
    public static Double[] vecvecminus(Double[] p, Double[] q) {
        Double[] p_plus_q = new Double[p.length];
        for(int f=0; f<p.length; f++)
            p_plus_q[f] = q[f] - p[f];
        return p_plus_q;
    }

    //计算全局评分平均数
    public static Double computeMu(Map<String,Map<String,Double>>
r) {
        Double sum = 0.0;
        int count = 0;
        for(String u : r.keySet()) {
            for(String i : r.get(u).keySet()) {
                sum += r.get(u).get(i);
                count++;
            }
        }
        return sum/count;
    }
//实现用户评分矩阵的转置
    Public static Map<String,Map<String,Double>>
Transpose(Map<String,Map<String,Double>> r) {
        Map<String,Map<String,Double>> r_i_u = new
HashMap<String,Map<String,Double>>();
        for(String u : r.keySet()) {
            for(String i : r.get(u).keySet()) {
                Double rating = r.get(u).get(i);
```

```
                if( !r_i_u.containsKey(i) ) {
                    Map<String, Double> map = new HashMap<String,
Double>();
                    r_i_u.put(i, map);
                }
                r_i_u.get(i).put(u, rating);
            }
        }
        return r_i_u;
    }

    //item集合
    public static Set<String>
get_items(Map<String,Map<String,Double>> r) {
        Set<String> items = new HashSet<String>();
        for(String u : r.keySet())
            for(String i : r.get(u).keySet())
                items.add(i);
        return items;
    }

    //user集合
    public static Set<String>
get_users(Map<String,Map<String,Double>> r) {
        return r.keySet();
    }

    //均方根误差的求解
    public static Double RMSE(Rec rec,
Map<String,Map<String,Double>> r, Map<String,Map<String,Double>>
test) {
        rec.buildRecommender(r);//初始化所需的参数、变量，迭代计算
评分预测的变量
        System.out.println("Computing RMSE and MAE...");
        Double RMSEsum = 0.0;
        int count = 0;
```

```
        for(String u : test.keySet()) {
            for(String i : test.get(u).keySet()) {
                double r_ui = test.get(u).get(i);
                double r_pred = rec.r_hat(u, i);
                //System.out.print(r_ui+" ; "+r_pred+" ; ");
                RMSEsum += Math.pow( r_ui - r_pred, 2.0 );
                //System.out.print(RMSEsum+" ; ");
                count++;
            }
        }
        System.out.println("Done with this test set");
        return Math.sqrt(RMSEsum/count);//RMSE的一种计算方法，预测
残差的平方和除以测试样本集合数的平方根
    }

    //均方根误差、平均绝对误差、召回率、准确率等指标的求解
    public static String RMSEplus(Rec rec,
Map<String,Map<String,Double>> r, Map<String,Map<String,Double>>
test) {
        rec.buildRecommender(r);
        System.out.println("Computing RMSE and MAE...");
        Double RMSEsum = 0.0;
        Double MAEsum = 0.0;
        Double TP = 0.0, FP = 0.0, TN = 0.0, FN = 0.0;
        Double threshold = 0.5;//阈值根据实际评分大小进行调整
        int count = 0;
        for(String u : test.keySet()) {
            for(String i : test.get(u).keySet()) {
                double r_ui = test.get(u).get(i);
                double r_pred = rec.r_hat(u, i);

    //System.out.println("====="+u+'\t'+i+'\t'+r_ui+'\t'+r_pred);

                RMSEsum += Math.pow( r_ui - r_pred, 2.0 );
                MAEsum += Math.abs(r_ui - r_pred);
                if (r_ui >= 0.5)
```

```
                    if (r_pred >= threshold)
                        TP += 1;
                    else
                        FN += 1;
                else
                    if (r_pred >= threshold)
                        FP += 1;
                    else
                        TN += 1;
                count++;
            }
        }

        Double precision = 1.0 * TP / (TP + FP);
        Double recall = 1.0 * TP / (TP + FN);
        Double fmeasure = 1.0 * 2 * (precision * recall) / (precision +
recall);
    Double accuracy = 1.0 * (TP+TN)/(TP+TN+FP+FN);

        String results = "RMSE="+Math.sqrt(RMSEsum/count)+";"
                        +"MAE="+MAEsum/count+";"
                        +"precision="+precision+";"
                        +"recall="+recall+";"
                        +"fmeasure="+fmeasure+";"
                        +"accuracy="+accuracy+";";
        return results;
    }

    //读取用户评分数据文件
    public static Map<String,Map<String,Double>> readData(String
filename) throws Exception {
        Map<String,Map<String,Double>> r = new
HashMap<String,Map<String,Double>>();
        System.out.println("Reading file " + filename + " ...");
        BufferedReader br = new BufferedReader( new
FileReader(filename) );
```

```
        String line;
        while ( (line = br.readLine()) != null )   {
            //System.out.println("Reading line: " + line);
            String[] array = line.split("\t");
            if (array.length == 1){
                array = line.split(" ");
                if (array.length == 1)
                    continue;
            }
            String user = array[0];
            String item = array[1];
            Double rating = Double.parseDouble(array[2]);

            if( !r.containsKey(user) )
                r.put(user, new HashMap<String,Double>());
            r.get(user).put(item,rating);
        }
        System.out.println("End of reading file " + filename);
        return r;
    }

    //根据物品读取评分数据文件
    public static Map<String,Map<String,Double>> readDataItem(String
filename) throws Exception {
        Map<String,Map<String,Double>> r = new
HashMap<String,Map<String,Double>>();
        System.out.println("Reading file " + filename + " ...");
        BufferedReader br = new BufferedReader( new
FileReader(filename) );
        String line;
        while ( (line = br.readLine()) != null )   {
            //System.out.println("Reading line: " + line);
            String[] array = line.split("\t");
            if (array.length == 1) continue;
            String user = array[0];
            String item = array[1];
```

```
            Double rating = Double.parseDouble(array[2]);
            if( !r.containsKey(user) )
                    r.put(user, new HashMap<String,Double>());
            r.get(user).put(item,rating);
        }
        System.out.println("End of reading file " + filename);
        return r;
    }
}
```

附录D 基于分类学习的元路径参数求解（Matlab）

```
输入：
    用户数量m；
    文献数量n；
    元路径下用户-文献评分列表 data(包括引用元路径、标注元路径)；
    用户对文献的实际行文（y=1 或 y=0)；
    迭代次数MAX_ITR；
    学习速率alpha

输出：theta
    % 初始化元路径权值theta
    theta = zeros(n+1, 1);
    % 定义：sigmoid function
    g = inline('1.0 ./ (1.0 + exp(-z))');

    J = zeros(MAX_ITR, 1);%初始化损失函数
    for i = 1:MAX_ITR
      % Calculate the hypothesis function
        z = x * theta;
        h = g(z);

    %计算梯度
        grad = (1/m).*x' * (h-y);
        %H = (1/m).*x' * diag(h) * diag(1-h) * x;
        H = (1/m).*x' * diag(h'* (1-h)) * x;
        J(i) =(1/m)*sum(-y.*log(h) - (1-y).*log(1-h));%损失函数
        theta = theta -alpha.*(H\grad);%元路径权值计算
end
```

附录E 基于贝叶斯优化排序的元路径参数求解（Java）

```
输入：
    用户users；文献items；
    元路径下用户-文献评分列表 data(包括引用元路径、标注元路径)；
    用户对文献的实际行文（y=1 或 y=0)；
    迭代次数iteration；
    学习速率alpha；
    正则项系数lamad
输出：theta
    //参数学习
    public static void learning(Map<String,Map<String,Double>>
t,Map<String,Double[]> pu1,
    Map<String,Double[]>qi1,Map<String,Double[]>pu2,Map<String,Dou
ble[]>qi2) //用户、文献对应的因子向量
        {
            while(iter<= iteration) //repeat:每一次迭代
            {
                System.out.println("The iteration "+iter+".....");
                Vector sum;//定义一个二维向量
                double c=0.0;
                for u,e in users,items // repeat:每一评分排序正确的用户-
文献对
                {
                    r_ab=r^(u_i,e_a)-r^(u_i,e_b);
            sum+=(Math.exp(-r_ab)/(1.0+Math.exp(-r_ab)))* ∂/∂theta (r_ab);
                    c+=Math.log(r_ab);
                }
                theta= theta- alpha*(-sum+lamad*theta);
                alpha=alpha*0.9;
                iter++;
            }  }
```

作者简介

姓名：江雪琴
性别：女
民族：汉
出生年月：1990-01-16
籍贯：安徽省黄山市

教育经历：
2009-09—2013-06 南京农业大学　信息管理与信息系统专业　学士
2013-09—2016-01 中国科学技术信息研究所 情报学专业　　硕士

获奖情况：
2014-11 硕士研究生国家奖学金

参加项目：
[1]中国科学技术信息研究所预研基金项目“基于海量科技文献的异构学术网络挖掘研究”（项目编号：YY-201417）
[2]国家数字复合出版系统工程28包“领域词表构建与管理系统”（项目编号：XWCB-ZDGC-FHCB/28）
[3]中国科学技术信息研究所重点工作项目“结构化知识服务平台建设及应用”（项目编号:ZD2015-2）

攻读硕士学位期间发表的学术论文：
[1]江雪琴,张志平,李琳娜. 知识获取的对数透视原理分析——以数据挖掘领域为例[J]. 情报杂志,2014,33(7):156-160.
[2]江雪琴,张志平,李琳娜. 基于互信息强度构建标签概念层次结构方法的探究[J]. 情报杂志,2014,33(12):165-169.

学位论文数据集

<table>
<tr><td>关键词*</td><td>密级*</td><td>中图分类号*</td><td>UDC</td><td>论文资助</td></tr>
<tr><td>推荐系统；隐语义模型；评分矩阵；异构信息网络；元路径</td><td>公开</td><td>TP391;G35</td><td>004</td><td></td></tr>
<tr><td>学位授予单位名称*</td><td colspan="2">学位授予单位代码*</td><td>学位类别*</td><td>学位级别*</td></tr>
<tr><td>中国科学技术信息研究所</td><td colspan="2">80901</td><td>管理学</td><td>硕士</td></tr>
<tr><td>论文题名*</td><td>并列题名*</td><td colspan="3">论文语种*</td></tr>
<tr><td>推荐系统中的隐语义模型算法研究</td><td>无</td><td colspan="3">中文</td></tr>
<tr><td>作者姓名*</td><td colspan="2">江雪琴</td><td>学号*</td><td>809011311</td></tr>
<tr><td>培养单位名称*</td><td>培养单位代码*</td><td colspan="2">培养单位地址</td><td>邮编</td></tr>
<tr><td>中国科学技术信息研究所</td><td>80901</td><td colspan="2">北京市海淀区复兴路15号</td><td>100038</td></tr>
<tr><td>学科专业*</td><td colspan="2">研究方向*</td><td>学制*</td><td>学位授予年*</td></tr>
<tr><td>情报学</td><td colspan="2">知识管理与技术</td><td>2.5年</td><td>2016年</td></tr>
<tr><td colspan="2">论文提交日期*</td><td colspan="3">2015年09月</td></tr>
<tr><td>导师姓名*</td><td colspan="2">张志平</td><td>职称*</td><td>研究员</td></tr>
<tr><td>评阅人</td><td colspan="2">答辩委员会主席*</td><td colspan="2">答辩委员会成员</td></tr>
<tr><td></td><td colspan="2">耿骞</td><td colspan="2">陈豫；徐硕；袁军鹏；曾文</td></tr>
<tr><td colspan="5">电子版论文提交格式 文本(√) 图像() 视频() 音频() 多媒体() 其它()
推荐格式：application/msword；application/pdf</td></tr>
<tr><td colspan="2">电子版论文出版（发布）者</td><td>电子版论文出版（发布）地</td><td colspan="2">权限声明</td></tr>
<tr><td colspan="2"></td><td></td><td colspan="2"></td></tr>
<tr><td colspan="2">论文总页数*</td><td colspan="3">88</td></tr>
<tr><td colspan="5">共33项，其中带*为必填数据，为22项。</td></tr>
</table>