

分 类 号: TP311

单位代码: 10183

研究生学号: 2017544052

密 级: 公开



吉 林 大 学

硕士学位论文

(专业学位)

基于神经图模型的试题推荐在线考试系统设计与研究

Design and Research of Online Examination System of Test Questions
Recommendation Based on Neural Graph Model

作 者 姓 名: 邢瑞

类 别: 工程硕士

领域(方向): 软件工程

指 导 教 师: 王岩 教授

培 养 单 位: 软件学院

2020 年 8 月

基于神经图模型的试题推荐在线考试系统设计与研究

Design and Research of Online Examination System
of Test Questions Recommendation
Based on Neural Graph Model

作者姓名：邢瑞

领域（方向）：软件工程

指导教师：王岩 教授

类别：工程硕士

答辩日期：2020 年 8 月 25 日

吉林大学硕士学位论文原创性声明

本人郑重声明：所呈交学位论文，是本人在指导教师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：邢瑞

日期：2020 年 9 月 7 日

关于学位论文使用授权的声明

本人完全了解吉林大学有关保留、使用学位论文的规定，同意吉林大学保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权吉林大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

（保密论文在解密后应遵守此规定）

论文级别： ☒ 硕士 ☐ 博士

学科专业： 软件工程

论文题目： 基于神经图模型的试题推荐在线考试系统设计与研究

作者签名： 邢瑞

指导教师签名：



2020 年 9 月 7 日

摘 要

基于神经图模型的试题推荐在线考试系统设计与研究

考试对于学生的整个学习过程来说是尤为重要的环节。然而,传统的纸质化考试在人工出题、试卷印刷、学生考试、人工阅卷和成绩统计等过程中,会产生大量的人力消耗和时间开销,而且试题选择固定或随机,缺乏学生个体针对性。

针对传统纸质化考试存在的问题,本文设计开发了在线考试系统。目前,虽然在线考试系统已经应用非常普遍,但是这些系统往往忽略了学生个性化的学习需求。市面上存在的考试系统,主要以营利为目的,更加注重界面的美观设计和功能的扩展;而很多高校的考试系统,则是为了测评学生近期的学习情况,所以更加关注核心的线上考试功能,相对比较简单。在如今数据量巨大、信息过载的时代,推荐系统作为一种个性化的解决方案,已经广泛应用在各个领域;而个性化的学习方式才能更好地激发学生的学习兴趣,更针对性地帮助学生,提高成绩。

为实现学生个性化的学习需求,并满足学校和平台的要求,本文实现的考试系统除了具备基本的自动组卷、阅卷功能外,还具有模拟试卷、知识点练习和错题重做等题目练习功能;同时将推荐算法引入所开发的考试系统中。本文主要聚焦于练习试题的推荐,在学生题目练习功能模块的知识点练习中设计了两种练习方式,分别为自主随机抽题练习和利用推荐算法推荐题目练习。针对题目推荐功能,本文研究了一种基于神经图模型的推荐算法,用于进行试题的推荐,可根据学生历史的错题记录,为学生推荐一些适合他们的易错题和知识点掌握不牢的题目。而实验也表明,本文研究的推荐算法相比其他类似的几种算法在推荐效果上有所提升。

本文主要工作如下:

(1) 针对学校和平台的需求,开发了一款在线考试系统,并且就学生个性化的学习需求,对推荐系统领域的一些典型应用进行了学习和了解,并探索了基于内容推荐、基于模型推荐和协同过滤等经典的传统推荐算法的原理。

(2) 针对考试系统试题推荐练习的功能,进行了推荐算法研究。在神经图协

同过滤算法（NGCF）的基础上进行了改进，提出一种基于深度学习和注意力机制（Attention）的推荐算法模型 NGCF-Att。首先，在 NGCF 的交互层采用多层神经网络代替了内积；其次，在传播层的消息构建中引入了注意力机制。

（3）使用 Tensorflow 实现推荐算法，并在可映射为本系统数据库格式的 Amazon-book 和 Gowalla 等公开数据集上进行实验来验证推荐算法的效果。

关键词：

在线考试，题目练习，试题推荐，神经图协同过滤算法，NGCF-Att，注意力机制

Abstract

Design and Research of Online Examination System of Test Questions Recommendation Based on Neural Graph Model

The examination is a particularly important link for the whole learning process of students. However, in the process of manual question setting, paper printing, student examination, manual marking and score statistics, traditional paper examination will incur a large amount of manpower and time expenditure. Moreover, the test questions are fixed or random and lack of individual pertinence.

To solve the problems of traditional paper-based examination, this paper designs and develops an online examination system. At present, online examination systems have been widely used, but these systems often ignore students' personalized learning needs. The existing examination system in the market, mainly for the purpose of making profits, pays more attention to the aesthetic design of the interface and the expansion of functions; Many colleges and universities have their own test systems, which are designed to measure students' recent learning, so they pay more attention to the core online test function, which is relatively simple. In this era of huge data and information overload, Recommendation system as a personalized solution, has been widely used in various fields. Personalized learning methods can better stimulate students' interest in learning, more targeted to help students learn and improve their performance.

In order to realize students' personalized learning needs and meet the needs of the school and the platform, this paper has realized the functions of automatic paper formation, marking, and the function of simulating test paper, practicing knowledge points and redoing wrong questions. At the same time, the recommendation algorithm is introduced into the developed examination system. This paper designs two ways to select questions in the test exercise function module, which are the traditional random selection and the recommendation algorithm. For the function of topic recommendation, this paper studies a kind of efficient recommendation algorithm, which can be used for the recommendation of test questions. According to the history records of students' wrong questions, students can be recommended some questions that are easy to be wrong and difficult to master. The experiment also shows that the

recommendation algorithm studied in this paper has improved the recommendation effect compared with other similar algorithms.

The main work of this paper is as follows:

(1) According to the needs of the school and the platform, an online examination system has been developed. And for students' personalized learning needs, this paper studied and understood some typical applications in the field of recommendation system, and explored the principles of classical traditional recommendation algorithms such as content-based recommendation, model-based recommendation and collaborative filtering.

(2) For test question recommendation exercise, the recommendation algorithm is studied. A recommendation algorithm model NGCF-Att based on deep learning and graph structure is proposed, which is based on the neural graph collaborative filtering algorithm (NGCF). Firstly, the inner product is replaced by the multi-layer neural network in the interaction layer of NGCF. Secondly, attention mechanism is introduced in the message construction of the propagation layer.

(3) Tensorflow was used to implement the recommendation algorithm, and experiments were carried out on public data sets such as Amazon-book and Gowalla that could be mapped into the database format of this system to verify the effect of the recommendation algorithm.

Key Words:

online test, practice of question, recommendation of test question, neural graph collaborative filtering algorithm, NGCF-Att, attention mechanism

目 录

第 1 章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 在线考试系统研究现状.....	2
1.2.2 推荐系统研究现状.....	3
1.3 研究内容和方法.....	6
1.4 论文结构安排.....	7
第 2 章 相关理论和技术.....	9
2.1 推荐系统.....	9
2.2 经典推荐算法.....	10
2.2.1 基于内容的推荐算法.....	10
2.2.2 基于协同过滤的推荐算法.....	11
2.2.3 混合推荐算法.....	14
2.2.4 基于深度学习的推荐算法.....	15
2.3 推荐算法常用评测指标.....	16
2.3.1 评分预测.....	16
2.3.2 TopN 推荐.....	17
2.4 相似度计算方法.....	17
2.5 系统开发框架.....	18

2.5.1	前端框架 Dojo.....	18
2.5.2	后端框架 SSH	19
2.6	本章小结.....	20
第 3 章	神经图模型试题推荐算法研究	21
3.1	神经图协同过滤算法.....	21
3.1.1	高阶连通性的概念.....	21
3.1.2	神经图协同过滤模型.....	22
3.2	神经图协同过滤算法改进.....	24
3.2.1	传播层改进.....	24
3.2.2	预测层改进.....	25
3.2.3	NGCF-Att 模型.....	27
3.3	本章小结.....	30
第 4 章	实验验证及结果分析	31
4.1	实验环境与数据集.....	31
4.1.1	实验环境.....	31
4.1.2	实验数据集.....	31
4.2	模型训练.....	34
4.3	评估指标及超参数设置.....	35
4.3.1	评估指标.....	35
4.3.2	超参数设置.....	35
4.4	结果对比分析.....	36

4.5 本章小结.....	37
第5章 试题推荐考试系统的设计与实现	38
5.1 系统需求分析和开发环境	38
5.1.1 开发环境介绍	38
5.1.2 功能需求分析	38
5.1.3 系统可行性分析	40
5.2 系统概要设计	41
5.2.1 系统分层结构	41
5.2.2 系统功能设计	42
5.2.3 数据库设计	43
5.3 系统详细功能的设计与实现	49
5.3.1 课程负责人端的功能实现	50
5.3.2 教师端的功能实现	57
5.3.3 学生端的功能实现	57
5.4 系统测试	61
5.4.1 测试目的及概述	61
5.4.2 系统功能模块测试	61
5.5 推荐算法应用	65
5.5.1 学生知识点练习推荐练习功能分析	65
5.5.2 推荐算法应用可行性	65
5.5.3 冷启动问题的解决	67

5.6 本章小结.....	68
第6章 总结与展望.....	69
6.1 工作总结.....	69
6.2 未来研究展望.....	69
参考文献	71
作者简介及科研成果.....	75
致 谢	76

第1章 绪论

1.1 研究背景及意义

互联网的发展和互联网+战略的不断深入,带动了教育领域的改革,突破了传统教育方式在时间和空间上的限制,减少了大量人力物力消耗。例如,考试对于学生的整个学习过程中是尤为重要的一个环节,但传统的纸质化考试包含了人工出题、试卷印刷、学生考试、人工阅卷和成绩统计等过程,整个过程会产生大量的人力消耗和办公经费开销。目前,在线教育蓬勃发展,涌现出了大批的在线考试系统,很好地解决了传统纸质化考试存在的弊端。然而在大数据时代下,在线教育领域也存在着信息过载、大量资源难以得到充分利用的问题。无论是市面上还是很多高校的一些在线平台,虽然提供练习和考试功能,但对所有学生用户提供同样的题库题目,忽略了学生个性化的学习需求。市面上存在的考试系统,主要以营利为目的,更加注重界面的美观设计和功能的扩展;而很多高校的考试系统,则是为了测评学生近期的学习情况,所以更加关注核心的线上考试功能,相对比较简单。而且对于这些系统来说,随着题库的积累,学生随机抽题练习的方式会特别盲目,学生不能在庞大的题库中找到适合一些自己的题目进行练习,甚至大量的题目没有机会抽取到,从而导致事半功半的效果。

而信息技术的发展和互联网的普及带来了大规模的信息,我们已经进入大数据时代。信息的普及和增长给用户带来便利的同时,也带来了信息过载的问题^[1]。面对如此海量的数据,无论是用户还是信息服务商都将遇到很大的挑战。用户很难检索出对自己有用或者感兴趣的信息^[2];信息服务商想让自己的信息脱颖而出同样不容易。为了解决信息过载的问题,很多优秀的方案被提出,典型的有分类目录和搜索引擎^[3]。分类目录将网站进行分类,为用户查找提供了一定便利性,但随着信息规模的扩大,分类目录很难覆盖全部信息。因此,随着搜索引擎诞生,它允许用户输入搜索的关键词筛选出自己所需要和感兴趣的信息,弥补了分类目录存在的缺陷。但是,搜索引擎需要用户主动提供准确的关

关键词进行搜索，一旦用户无法明确定位自己的真实需求，搜索引擎也就失效了。此外，搜索引擎不够个性化，不同的用户输入相同的关键词时，会得到相同的结果。作为搜索引擎的互补工具，推荐系统^[4]很好地解决了以上存在的问题。推荐系统也是一种帮助用户快速发现有用信息的工具，它不需要用户提供明确的需求，而是通过分析用户的个人信息和历史行为，分析出用户的兴趣、偏好和习惯，然后将数据信息进行过滤排序，生成列表展现给用户，从而更好地为用户提供个性化的服务。目前，推荐系统的身影随处可见，已是各大主流网站不可缺少的一部分，包括电子商务、新闻、视频、社交网络等。这些也为个性化教育提供了技术支持和研究契机。

而推荐算法是推荐系统的核心成分。传统的推荐算法包括基于内容的推荐算法、协同过滤算法^{[5][6]}、混合推荐等，其中协同过滤的应用最为广泛。随着深度学习的发展和数据集规模的增加，传统的推荐算法已经不能很好地解决数据稀疏、冷启动等问题，此外学得的模型准确率达不到要求。因此，目前学术界和工业界研究的推荐系统算法大多基于深度学习^[7]。

如今，个性化教育是一个重要的研究课题。基于对以上问题的思考，本文将推荐算法应用到所开发的在线考试系统中。主要针对在线考试系统的试题练习功能模块，研究了一种个性化的推荐算法，可根据学生用户历史的错题情况，为不同学生推荐不同的并且适合他们的一些优质题目供其练习，使得练习不再盲目，更有针对性，从而提高学习效率。这样既对学生也对老师在一定程度上起到了帮助的作用，同时对个性化教育的发展也起到了一定的推动作用。

1.2 国内外研究现状

1.2.1 在线考试系统研究现状

对计算机辅助教学和在线教育的研究，国外是领先于国内的。全球第一个在线考试系统诞生于美国，由美国科技公司于 1957 年设计。1971 年，国外的一些计算机科学家首次提出了自适应考试 CAT 并且得到了快速地发展。Novell 采用了 CAT 考试形式，考生人数已突破百万人次。1993 年，美国教育考试服务中心 ETC 也采用了 CAT 考试机制，典型的考试项目包括 TOEFL 和 GRE。1995 年，美国和一些英国的一些大学都相继研发了自己的网络教学平台，之后美国 ATA

研发了第一个智能化的考试系统，具有在线考试和试题管理等功能。在此基础上，加拿大 OCHE 又增加了考试分析和成绩评定等功能。目前在国外，CAT 已经应用相当广泛，知名的 IBM 认证、托福、Oracle 认证等均采用在线考试的方式进行。

相比于国外在线考试的蓬勃发展，国内对在线考试系统的研究要晚很多。直到 1998 年，在线教育平台才开始兴起，随着在线考试系统的引入，避免了传统纸质化考试大量的人力物力消耗，提高了管理效率，简化了复杂的组织流程。计算机和互联网技术的不断发展，带动了在线考试系统在国内的飞速发展，全国各大高校已经相继开发和使用了自己的在线考试平台。此外各大企业在员工招聘考核当中，也都采用在线笔试和远程面试的形式进行，大大提升了人才引进的效率。除了校园和企业之外，国内有一些知名的社会化认证考试，如计算机等级考试、英语四六级考试、普通话水平考试也都是采用线上的方式进行。还有很多专注于打造线上教育平台的公司和企业，利用互联网技术和人工智能技术，已经把在线考试系统设计的功能相当强大，界面更加美观大方，操作起来更加方便和智能，为学校、企业和社会的线上考核提供了极大的便利，推动了网络教学和在线考试的发展。即便在线考试系统发展很好，目前在大数据背景下，很多平台仍然缺乏对个性化的学习方式的考虑。

1.2.2 推荐系统研究现状

推荐系统作为互联网时代的一种信息搜索工具^[8]，自 20 世纪 90 年代被提出，至今已经应用非常广泛，它已经成为一门独立的学科，不管是工业界还是学术界都取得了很多成果，尤其在电子商务、社交网络及娱乐影音等领域，已经成为不可或缺的一部分。

最早的推荐系统可以追溯到基于协同过滤算法的邮件系统 Tapestry^[9]，由 Xerox 公司在 1992 年设计，为了解决 Palo Alto 研究中心信息过载的问题。同时，Goldberg 第一次提出了 CF (Collaborative Filtering，简称 CF) 这一概念，但是并未实现主动推荐。1994 年，Minnesota 大学的 GroupLens 小组开发了第一个基于主动协同过滤推荐算法的自动化推荐系统 GroupLens^[10]，并应用在 Usenet 新闻组中，通过协同过滤算法根据计算用户评分来预测用户感兴趣的新闻，推荐

效果有所提升。后来，GroupLens 研究组又推出了广为人知、影响巨大的 MovieLens 电影推荐系统，基于 Web，其数据集 MovieLens 更是成为了学术界研究推荐算法的常用典型数据集。随之而来的还有 MIT 媒体实验室开发的音乐推荐系统 Ringo，BellCore 开发的视频推荐系统 Video Recommender 以及著名的 Youtube^[11] 视频推荐系统。

1997 年，Varian 和 Resnick 指出推荐系统有助于电商为用户提供商品购买建议，这也是推荐系统首次在学术界被正式提出。从此推荐系统开始成为一个重要领域，也从探索阶段逐步迈入商业化阶段，渗透到各个领域，典型的有亚马逊和 Facebook 社交网站。2003 年，亚马逊 Linden 等人发表论文^[12]，公布了基于物品的协同过滤算法。

2005 年，Adomavicius 等人将推荐系统分为基于内容的推荐、基于协同过滤的推荐和混合推荐 3 类，目前被学术界广泛接受。基于内容的推荐^[13]是建立在物品的内容信息之上的，不太依赖用户的评分，更多的是通过机器学习的方法从物品的内容特征中得到用户的兴趣资料，没有冷启动问题和稀疏问题，而且能为具有特殊兴趣爱好的用户进行推荐，效果也很不错，但是对物品内容特征的抽取和结构要求较高。传统的协同过滤算法建立在用户已经对物品产生过交互行为的基础上，一般采用最近邻技术，根据相似度公式计算出用户或者物品之间关联，将关联最紧密的进行推荐，它包括基于用户的协同过滤、基于物品的协同过滤和基于模型的协同过滤。基于用户^[14]的协同过滤的思想是通过计算用户之间的相似度将相似用户的物品推荐给对方，相当于猜你喜欢，可能推荐结果不是那么准确。而基于物品^{[15][16]}的协同过滤则是通过计算物品之间的相似度，直接将类似的物品推荐给用户。相比于前者，对于用户来说，后者推荐的结果可能更准确一些。最后基于模型的协同过滤是利用机器学习的方法，根据用户和物品的交互行为进行建模，如矩阵分解模型^[17]等，然后利用建立好的模型对用户的行为进行预测。事实上，没有任何一种推荐是万能的，适合于任何场景，而且在一些复杂的场景下，一些现有的推荐模型可能不满足需求。混合推荐^{[18][19]}算法就是将两种或者更多的推荐算法融合在一起，大大提升了效果，目前应用在各大商业领域中。混合是一种策略，将现有模型融合，没有更深入地改进和挖掘算法，但是在加强模型，提高准确率上有很大的意义。

2006 年 10 月，北美视频服务商 Netflix^{[20][21]}举办了一项竞赛引起了学术界和工业界极大地关注，任何人只要能将它现有的电影推荐算法 Cinematch 的准确度提升 10%，就能获得 100 万美元奖金，期间诞生了许多优秀的推荐算法，极大地推动了推荐系统的发展。2007 年第一届推荐系统顶级会议 RecSys 开始举办，为学术界和工业界广泛的推荐系统学术交流提供了平台，将推荐系统的研究推向高潮。

三类推荐系统相比之下，协同过滤推荐算法的研究和应用更为广泛。协同过滤需要面临的最大问题就是冷启动和稀疏性问题，所以传统的协同过滤需要尽可能多的用户交互数据，如果在评分数据很少的情况下，效果会很不尽人意，所以研究者们更多采用隐式评分数据。后来的研究中，大多都是针对以上的两大问题进行创新和改进。Sarwar 等人应用线性代数中的奇异值分解^[22](SVD)提出一种矩阵分解方法，可以降低评分矩阵维度，但是降维就会有损失重要信息的风险。此外由于奇异值分解要求矩阵稠密，所以首先需要对评分矩阵进行填充，时间复杂度太大。为了提高效率，Simon Funk 提出了一种新的 SVD 方法，也称为隐语义模型，模型简单，将每个用户和物品映射成一个向量，且用内积表示评分，为后来 embedding 的研究做了一定铺垫，但是没考虑到每个用户和物品的差异性，所以通常都会为每个用户和物品添加一个 baseline。Koren 提出了 SVD++^[23]的方法，综合考虑到用户本身和交互的物品对用户特征的一个表示，提升了效果。

自从 2006 年，深度学习被 Hinton 等人提出之后，就逐渐被科学界广泛关注和应用，它在图像处理、自然语言处理等方面取得了突出的成就，尤其是近些年，深度学习的研究和应用相当广泛。Salakhutdinov^[24]等人较早将深度学习应用到推荐算法当中，使用限制玻尔兹曼机构建模型，但是神经网络层与层之间连接规模太大。目前推荐算法的研究大多都是建立在深度学习之上。Covington^[25]等人提出了一个基于多层神经网络的推荐算法，并应用到了 YouTube 视频推荐。Travis^[26]等人提出了一种基于记忆的神经网络协同过滤算法，利用到了用户物品的交互关系，通过记忆层将相邻用户以非线性的方式结合到用户的表示中，使得用户的表示更加准确，效果上有一定的提升。但是在近邻用户结合的时候只考虑到了一阶连通性，没有挖掘到更深层的用户特征信息。Wang^[27]等人

为了解决传统协同过滤所面临的稀疏性问题，提出了一种基于深度学习的分层贝叶斯模型，添加了一定的辅助信息，将内容信息用神经网络进行表示，考虑到了复杂的非线性关系，增强了向量表示性能。但在预测层采用的仍然是内积的形式，不足以揭示用户和物品之间交互的复杂的非线性关系。He^[28]等人针对Wang的研究，采用非线性神经网络结构代替内积操作来增强交互功能，以捕获用户和物品之间的非线性特征交互，提出了一种NCF模型。PinSage^[29]是工业上的应用，采用多个卷积层在物品图上进行Pinterest图像推荐，因此，是在物品的水平上捕获关系，而不是在集体的用户行为上进行。SpectralCF^[30]提出了一种频谱卷积操作，以发现用户和频谱域中的物品之间的所有可能的连通性，通过图邻接矩阵的特征分解，可以发现用户-物品对之间的里联系，但是复杂度很高，非常耗时，不支持大规模推荐方案。Hop-Rec^[31]是一个基于图形的最新的模型，利用了随机游走探索衍生的高阶邻居来丰富用户-物品交互数据，但只用来丰富了训练数据，预测模型仍然是MF，通过优化损失进行训练，而损失会因为高阶连通性而增大。最近Wang^[32]等人提出了一个神经图协同过滤模型，是一种将高阶连通性显式地集成到预测模型中的新技术，通过堆叠多层embedding传播层，充分捕获高阶连通性中的协作信号，获得了较好的效果。这项工作是在基于模型的CF中利用消息传递机制开发结构知识的初步尝试，开辟了新的研究可能性，推动了深度学习在推荐系统领域应用的发展。

1.3 研究内容和方法

本文的主要目标是完成在线考试系统的开发设计，并为所开发的在线考试系统的试题练习功能研究一种个性化的推荐算法，来完成试题的推荐。通过分析学生用户历史考试的错题情况，最终为能为学生用户生成一个推荐试题答题卡，从而帮助学生高效地进行试题的练习。为此，论文的主要研究内容包括：

- 1、开发了一个在线考试系统，并针对考试系统试题练习功能模块的推荐练习方式，进行了基于深度学习的推荐算法研究。详细研究了神经图协同过滤（NGCF）算法，对NGCF算法进行了改进，最终提出一个引入注意力机制（Attention）的推荐模型NGCF-Att。

- 2、本文使用Tensorflow和python实现推荐算法部分。由于本文实现了系统

原型，仍在测试阶段，真实学生做题记录数据正在搜集过程中，所以选取了可映射到本系统原型数据库结构的 Amazon-book 和 Gowalla 两套公开数据集进行实验来验证推荐算法的效果。推荐算法研究基于隐式数据，主要关注推荐算法本身的准确性和推荐功能，在进行数据处理之后，忽略了数据集本身的领域和内容属性，只保留 id 属性。每行首个 id 表示 user，其余 id 表示该用户交互 item 集合，对于任何数据集，经数据处理之后格式一致。因此，可以将公开数据集作为学生做题记录数据的测试数据使用，进行算法研究。

本文内容主要组织结构如图 1.1 所示。

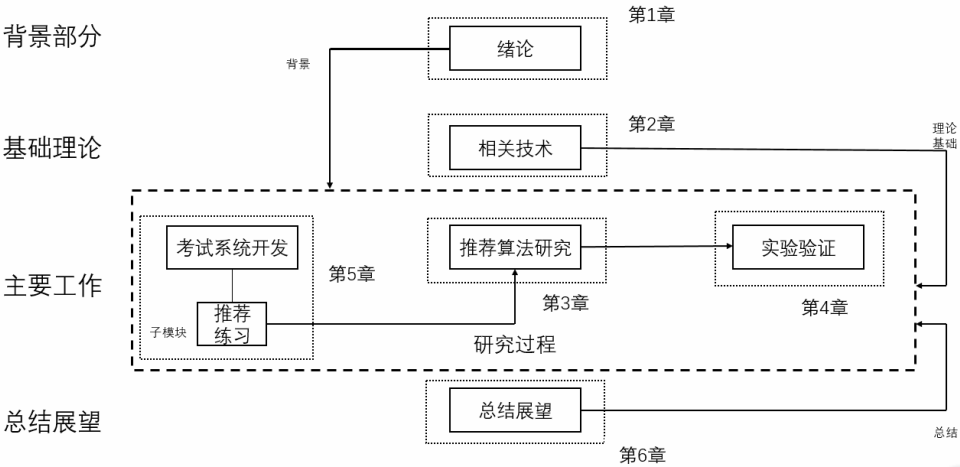


图 1.1 论文内容组织结构

1.4 论文结构安排

本文共分为六章，各章节的安排如下：

第一章为绪论。首先介绍了推荐系统研究相关的背景以及个性化教育研究的意义，然后详细分析了推荐系统的发展历程和国内外关于推荐系统的研究做出的贡献及研究状况，最后概述了本文的主要研究内容以及给出了各章节的安排说明。

第二章为相关理论和技术。首先对推荐系统的概念进行了简单介绍，然后就一些推荐系统经典的推荐算法以及评价指标等进行了概述，最后对系统主要的前后端框架做了说明。

第三章为神经图模型试题推荐算法研究。基于目前深度学习推荐领域前沿的神经图协同过滤算法和神经协同过滤算法，进行了推荐算法的研究和改进。

主要在神经图协同过滤算法的基础之上，借鉴了神经协同过滤算法的思想，用多层神经网络代替了神经图协同过滤算法交互层的内积方式，然后又引入了注意力机制（Attention），最终提出改进的神经图结构模型 NGCF-Att。

第四章为实验验证与结果分析。根据一些常用的推荐算法指标设计实验。由于本文实现了系统原型，仍在测试阶段，不能收集到大量的学生真实做题数据，但算法的研究主要关注推荐算法本身的性能，所以在数据处理之后，忽略掉了数据自身的内容属性和领域性，只保留了 id 属性，方便了对推荐算法进行性能研究和实验验证。因此，选取了公开且广泛用于推荐算法研究的数据集 Amazon-book 和 Gowalla 来进行实验，验证推荐算法的有效性。

第五章为试题推荐考试系统的设计与实现。主要对在线考试系统的实现做了整体的说明，包括系统的需求分析，各个功能模块的实现及测试、数据库表设计以及所用到的开发技术。然后针对系统推荐试题的题目练习方式，探究了推荐算法应用的可行性，明确了系统可搜集的数据以及推荐算法所需数据类型。

第六章为总结与展望。对本文所做的工作做出总结，针对工作中的问题和未能实现的地方做出说明，并对未来的工作方向做出展望。

第 2 章 相关理论和技术

2.1 推荐系统

互联网的快速发展，带来了大量的信息，同时产生信息过载的问题。分类目录和搜索引擎都是解决信息过载问题而诞生的，推荐系统^[33]也是相应的产物，可以作为搜索引擎的互补技术。当用户需求明确的时候，需要搜索引擎获取信息，没有明确需求的情况下，推荐系统可针对用户进行个性化推荐，也就是推荐系统更加个性化^[34]。

推荐系统是一种信息过滤系统^[35]，可以预测用户对物品的偏好。推荐系统的整体流程如图 2.1 所示。

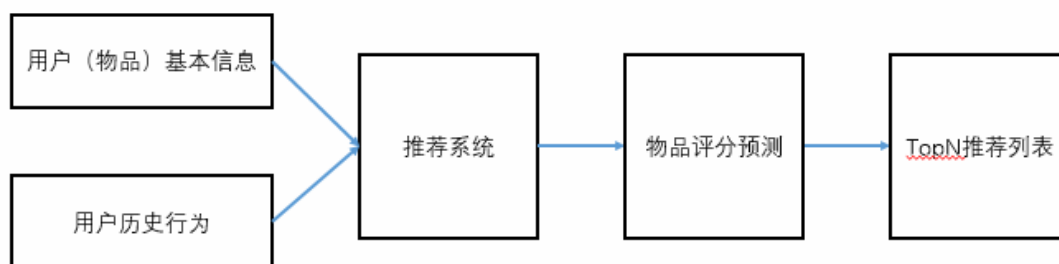


图 2.1 推荐系统整体流程

推荐系统解决了很多问题，包括经典的长尾问题^[36]。长尾指的是用户关注和购买更多的是一些热门的物品，但这些物品占商家物品总和的比例很小，大多数物品都不会被关注到，而这些物品的销售额却是一个很客观的数字，推荐系统正是帮助商家把这些物品推荐出去，从而获得更大收益。同样，对用户来说，可以得到自己意想不到的一些物品和信息，找到自己的兴趣，提升惊喜度和体验，实现个性化。

推荐系统的本质就是针对用户需求不明确的情况，通过一定方式将用户和物品连接起来，既帮助用户发现有用信息，又使得信息能展现给对它感兴趣的用户，达成双赢。不同的推荐系统使用了不同的推荐方式，但整体上都包括前端推荐页面、后台日志及推荐算法三部分，其中，推荐算法是最核心的部分。

2.2 经典推荐算法

推荐系统的广泛应用和发展，衍生出了很多经典的推荐算法，包括热度推荐、基于内容的推荐、协同过滤推荐、关联规则^[37]推荐和基于标签^[38]的推荐等。

自 2005 年推荐系统被分为基于内容的推荐、协同过滤和混合推荐，已被学术界广泛接受。直至今，已经有好多人在传统协同过滤的基础上进行了研究和改进，包括基于图模型的协同过滤和基于神经网络的协同过滤等。

推荐系统的一个分类如图 2.2 所示。

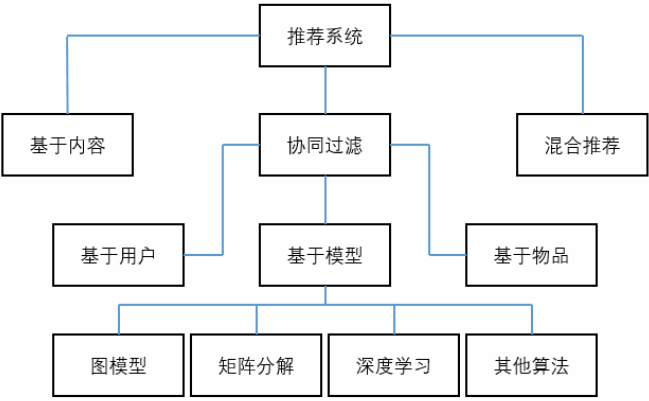


图 2.2 推荐算法分类

2.2.1 基于内容的推荐算法

基于内容的推荐算法是利用用户过去一段时间喜欢的物品推算出用户偏好，从而进行推荐。即根据用户之前浏览过的物品的基本信息，提取用户的偏好特征，然后将与用户感兴趣的物品内容最相似的物品推荐给用户。一个基于内容推荐的例子如图 2.3 所示。

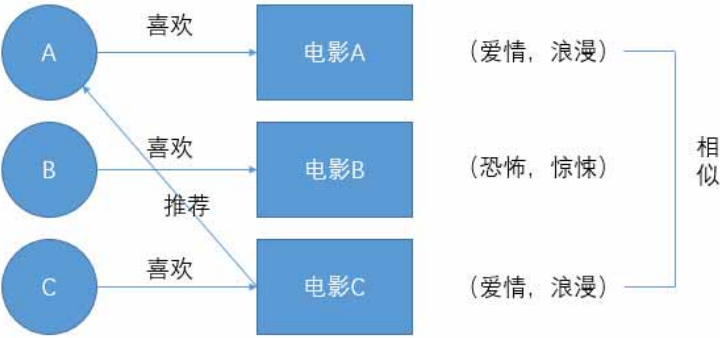


图 2.3 基于内容的电影推荐

基于内容的推荐算法实现通常包含三个过程：

- 1、构造物品特征。这个过程就是对物品的属性进行分析，然后用恰当的格式进行表示。
- 2、用户特征提取。根据观察一个用户历史喜欢的物品的特征数据，从而学习出该用户的偏好特征。可以利用 KNN、决策树和朴素贝叶斯等特征学习算法来完成这个过程。
- 3、生成推荐列表。产生一个最符合用户特征的物品列表，推荐给用户。

基于内容的推荐算法具有几个优点：

- 1、推荐结果具有可解释性。用户的兴趣可以被标签化，每一个推荐结果都可以带标签展示，用户可以很清楚的知道为什么推荐这些物品。
- 2、推荐结果稳定且不依赖其他用户数据。为某个用户推荐物品，只需要观察和分析该用户的历史行为，然后提取兴趣特征，进而完成推荐即可，无需借助其他用户历史行为数据。
- 3、不存在物品冷启动^{[39][40]}问题。适用于推荐系统刚建立的阶段，对于一个新的物品，直接可以计算与之前所有物品的相似度，从而推荐给用户。

基于内容的推荐算法也存在一些缺点：

- 1、缺乏用户惊喜度。推荐物品都是与之前类似的物品，变化不大，不会推荐一些新鲜的东西，无法挖掘用户潜在的兴趣，长期下去，导致用户兴趣丢失。
- 2、存在用户冷启动问题。对于新用户，缺乏历史的数据，所以不能很好地发现和提取出该用户兴趣特征，从而也就不能准确推荐物品。
- 3、物品特征抽取较困难。

2.2.2 基于协同过滤的推荐算法

协同过滤基于用户与物品的交互数据来产生推荐列表，可分为基于用户的协同过滤算法（User-Based CF）、基于物品的协同过滤算法（Item-Based CF）和基于模型的协同过滤算法（Model-Based CF）。

1、基于用户的协同过滤算法

基于用户的协同过滤算法的思想是通过用户的历史行为，找到用户喜欢的物品，然后观察其他用户，找到与该用户喜欢类似物品的用户，将其所喜欢的且该用户不曾购买的物品推荐给该用户。一个基于用户的协同过滤的例子如图 2.4 所示。

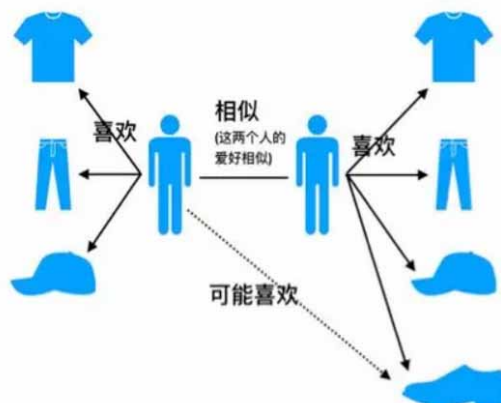


图 2.4 基于用户的协同过滤

基于用户的协同过滤算法通常包含三步：

(1) 构建用户物品评分表。每一行代表一条用户数据，每一列代表各个用户对某个物品的打分。

(2) 计算用户相似度。根据构建的评分表数据，计算用户之间的相似度，计算相似度的方法有很多，例如余弦相似度。

(3) 生成推荐结果。根据(2)的计算结果，为用户推荐和他相似度最高的几个用户喜欢的且该用户没有购买过的物品。

2、基于物品的协同过滤算法

基于物品的协同过滤算法是通过计算物品之间的相似性，然后根据用户历史发生过交互行为的物品，为其推荐类似的物品。与基于内容不同，它衡量相似度的方法也不是基于物品内容属性上的相似，而是取决于同时购买两个物品的用户，如图 2.5 所示。

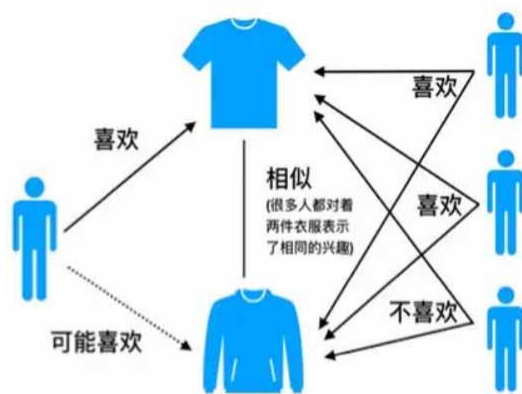


图 2.5 基于物品的协同过滤

基于物品的协同过滤算法主要有三步：

- (1) 计算物品之间的相似度。
- (2) 生成推荐结果。

(3) 对热门物品进行惩罚。由于一些热门物品的影响，在计算物品之间相似度的时候就会出现任何物品都和热门物品有很高的相似度，因此需要采取一定的措施，降低热门物品的权重。

上述两种协同过滤算法应用广泛，但也具有局限性：

(1) 评分稀疏性^[41]。利用用户-物品评分表构造的评分矩阵，每个用户只是对少数的物品有过交互，大部分位置都是空值，因此矩阵非常稀疏，计算偏差较大，导致丢失准确性，推荐效果不好，这也正是协同过滤面对的最大问题。

(2) 冷启动问题。同时存在物品冷启动和用户冷启动问题，对于新加入的物品，既没有过用户历史交互，也没有办法计算物品相似度，因此协同过滤算法不会将它推荐给用户；而对于新注册的用户，没有历史交互，也找不到相同兴趣的用户，所以，协同过滤算法不会给用户推荐物品。

3、基于模型的协同过滤算法

基于模型的协同过滤推荐算法与机器学习密切相关，它构建用户和物品的特征表示，然后利用这些特征作为训练数据，学得一个评分预测模型，输入用户和物品可计算出评分，然后按照评分排序由高到低，将排序靠前的推荐给用户即可。常见的模型有矩阵分解、隐语义模型、图模型以及深度学习模型等。

基于矩阵分解的隐语义模型 (LFM) 可以看作是将高维的评分矩阵 R 分解成两个低维的矩阵，分别是用户矩阵 P 和物品矩阵 Q ，如图 2.6 所示。

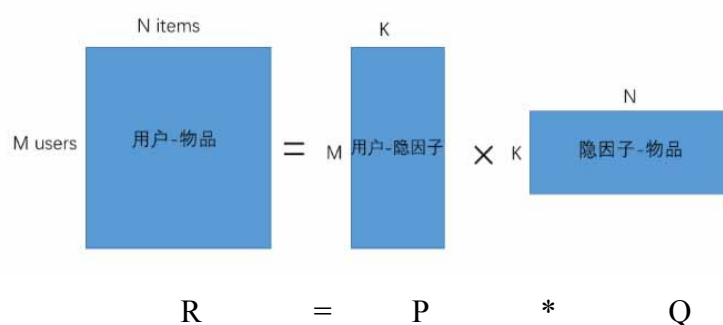


图 2.6 评分矩阵分解

图 2.6 中，矩阵 R 是用户对物品的评分， P 的每一行可以看成是一个向量，用来表示一个用户， Q 的每一列是一个用来表示物品的向量， P 的每一行与 Q 的每一列的乘积，也就是两个向量的内积，对应 R 位置处的评分。从而可以得出用户对物品的兴趣度公式，如公式 2.1：

$$R(u,i) = \sum_{k=1}^K P_{u,k} Q_{i,k} \dots\dots\dots (2.1)$$

LFM 最终求的是 P 和 Q ，通常采用均方误差作为模型的最优化损失函数，函数公式，如公式 2.2：

$$Loss = \sum_{(u,i) \in S} (R_{ui} - \sum_{k=1}^K P_{uk} Q_{ik})^2 + \lambda ||P_u||^2 + \lambda ||Q_i||^2 \dots\dots\dots (2.2)$$

上式中， λ 是正则项系数，后两项为正则项^[42]，为了避免过拟合。损失函数表示的是用户 u 对物品 i 的真实评分与模型计算出来的评分的误差平方和，要使得模型最优， $Loss$ 的值应该接近 0。

2.2.3 混合推荐算法

混合推荐算法利用模型融合策略，将两种或两者以上的推荐算法组合到一起，充分发挥了每一种推荐算法的优势，从而提升了准确率，提高了推荐系统的性能。

通常有以下几种模型融合的策略：

1、加权型。将多个模型的结果通过加权累加到一起，最简单的就是线性加权和，开始可以将所有模型的结果赋予相同的权重，然后根据实际情况不断调整。这种融合策略有如下的形式，如公式 2.3：

$$R(u,i) = w_1 R_1(u,i) + w_2 R_2(u,i) + \dots + w_n R_n(u,i) \dots\dots\dots (2.3)$$

上式中， R_i 表示各个推荐模型， w_i 表示各个模型的权重。

2、瀑布型。利用后一个模型优化前一个模型，是一个阶段性的过程。首先用一个模型得出结果，然后将这个结果作为第二个模型的输入，依次进行下去，最终得到一个比较精确的结果。形式如公式 2.4:

$$R_2(u,i) = g_1(R_1(u,i)), \dots, R_n(u,i) = g_{n-1}(R_{n-1}(u,i)) \dots \quad (2.4)$$

上式中， g_i 表示表示下一个算法模型。

3、特征组合型。将不同模型的数据的特征组合起来，作为另外一种推荐模型的训练数据。这种方式一般将协同过滤算法的特征数据作为增加的特征向量，最后用基于内容的推荐算法，计算这些最终组合后的特征数据的相似度。使得整个推荐系统不只是依赖协同过滤的那些数据特征，降低对评分矩阵数据量的敏感度。

此外，还有转换型、合并型、元层次型、特征递增型等多种模型融合的策略。实际应用中要根据不同的场景应用不同的策略，这样才能更好地提升推荐系统的准确率，从而满足需求。

2.2.4 基于深度学习的推荐算法

近几年，深度学习^[43]发展迅速，在推荐领域中，也应用非常广泛。例如阿里的 TDM 算法，将树结构与神经网络相结合，在数据量很大的情况下，会对用户兴趣进行层次切分，避免运算超载，提升推荐结果新颖度召回效果。在广告推荐应用中，利用注意力机制模型^[44]新增 DIN 结构，对用户历史数据和物品进行匹配并计算权重，最终实现推荐。

深度学习应用到推荐系统领域主要体现了以下几方面的优势：

- 1、提取特征的能力强，能够直接从内容中获取；
- 2、很容易处理噪声数据，抗噪声能力强；
- 3、循环神经网络^[45]可以方便地对序列数据进行建模；
- 4、较传统的协同过滤，能更好地捕捉数据的非线性特征，学得更深层次的特征，从而结果更准确。

基于深度学习的推荐算法输入用户和物品信息数据和历史评分数据，训练模型，学得用户和物品的表示，然后通过这些表示，计算出评分，最终生成推荐列表。典型的一些深度学习推荐模型有 CNN、RNN、RBM、GAN、AE、

NADE 等。

著名视频网站 YouTube 的推荐系统模型，是一个经典的深度学习在推荐系统中的应用，也可以算是深度学习在推荐领域的首个成功的应用，它的模型结构如图 2.7 所示。

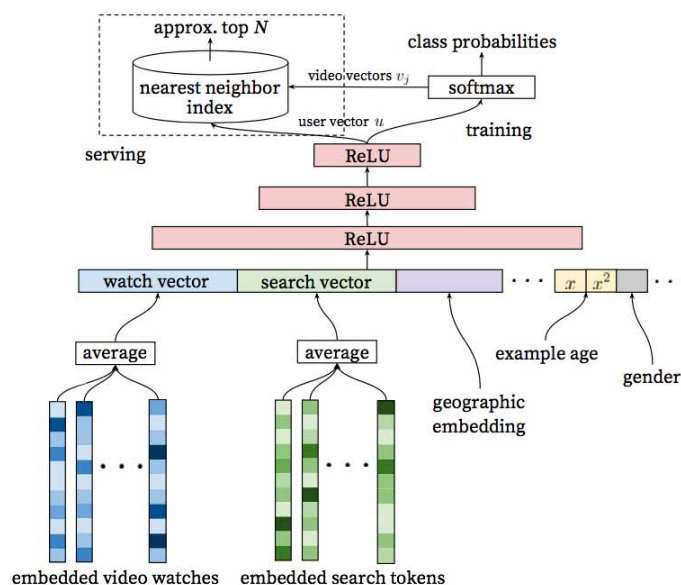


图 2.7 YouTube 视频推荐模型结构

2.3 推荐算法常用评测指标

对于推荐算法的研究，通常采用离线评估的方式进行。而评估指标主要分为两大类：准确度指标和非准确度指标。非准确度指标包括推荐惊喜度、覆盖率、信任度、新颖性、健壮性、实时性、多样性等，但这些都比较抽象，对于研究本身来说，更加注重的是准确度指标。而准确度指标有可分为评分准确度和推荐准确度。

2.3.1 评分预测

评分预测准确度衡量算法预测评分和用户实际评分之间的偏差，非常重要的一个指标，如果推荐系统只能向用户提供推荐列表，而不能预测评分，往往推荐结果不尽人意。衡量指标通常有：平均绝对误差（MAE）、均方误差（MSE）、均方根误差（RMSE）。

1、平均绝对误差。表示的是预测值和真实值之间的差值的绝对值的和，再

取平均值。如公式 2.5:

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i| \dots\dots\dots (2.5)$$

2、均方误差。表示的是预测值和真实值之间的差值的平方和，再取平均值。如公式 2.6:

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \dots\dots\dots (2.6)$$

3、均方根误差。表示均方误差的开方。计算公式如公式 2.7:

$$RMSE = \sqrt{MSE} \dots\dots\dots (2.7)$$

2.3.2 TopN 推荐

实际推荐系统中，最终应该为用户推荐评分最高的前 N 项物品，也就是 TopN 列表，而推荐准确度衡量指标衡量的正是推荐算法是否能正确预测到用户是否喜欢或不喜歡某些物品。通常包括召回率（Recall）、精确率（Precision）、准确率（Accuracy）、AUC 等指标。

1、召回率。表示在原始样本的正样本中，最后被正确预测为正样本的概率。计算方式如公式 2.8:

$$Recall = \frac{TP}{TP + FN} \dots\dots\dots (2.8)$$

2、精确率。表示预测为正样本的样本中，预测正确的概率。计算方式如公式 2.9:

$$Precision = \frac{TP}{TP + FP} \dots\dots\dots (2.9)$$

3、准确率。表示分类正确的样本占样本总数的比例，计算方式如公式 2.10:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \dots\dots\dots (2.10)$$

公式 2.8、2.9 和 2.10 中，TP 表示实际为正样本，预测也是正样本的样本数。TN 表示实际为负样本，预测也为负样本的样本数。FP 表示实际为负样本，但预测为正样本的样本数。FN 表示实际为正样本，但预测为负样本的样本数。

2.4 相似度计算方法

相似度是一个非常重要的概念，相似度计算在协同过滤算法中也是最为关

键的一步，无论是在推荐系统中，还是在数据挖掘的过程中，都离不开相似度的计算。在协同过滤算法中，利用相似度计算来计算出用户之间或者物品之间的相似性；在 k-means 聚类算法中，利用相似度计算某个个体到各个簇的距离，从而划分个体所属类别。计算相似度的方法也有很多，下面是常见的几种。

1、欧式距离。用于计算 n 维空间中，两个点之间的真实距离，假设两个向量分别为 P, Q ，则计算公式如公式 2.11：

$$\text{Dist}(P,Q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \dots\dots\dots (2.11)$$

上式中， p_i 和 q_i 分别代表 P, Q 的分量。

2、余弦相似度。表示的是两个向量夹角的余弦值，这个值越小，相似度越大反之，相似度越小。余弦相似度计算公式如公式 2.12：

$$\cos(P,Q) = \frac{\sum_{i=1}^n p_i q_i}{\sqrt{\sum_{i=1}^n p_i^2} \sqrt{\sum_{i=1}^n q_i^2}} \dots\dots\dots (2.12)$$

3、皮尔逊相关系数。衡量随机变量 X 与 Y 相关程度的一种方法，结果取值范围是 $[-1,1]$ ，取值的绝对值越大，相关性越强，取值为 1 代表正线性相关，-1 代表负线性相关，不适合计算两个布尔型向量之间的相似性。计算公式如公式 2.13：

$$\text{pearson}(X,Y) = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum (X - \bar{X})^2} \sqrt{\sum (Y - \bar{Y})^2}} \dots\dots\dots (2.13)$$

4、Jaccard 距离。表示的是两个集合的交集元素在这两个集合的并集中所占的比例，也称为 Jaccard 相似系数，适合计算两个布尔型向量之间的相似性。假设两个集合为 A 和 B ，则计算公式如公式 2.14：

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \dots\dots\dots (2.14)$$

2.5 系统开发框架

2.5.1 前端框架 Dojo

Dojo 是一个开源的 JavaScript 框架，或者严格来说，是一个工具包，是框架的一部分，主要用于构造 web 应用程序。Dojo 有一个重要的特点是面向对象，

根据功能分成很多个 module,每个 module 有包含多个 package,可以根据实际需要导入所需的 package,这一点特别类似于 Java。它引入包机制进行动态加载对于 JavaScript 编程来说是一个新想法,只需要一段很小的加载代码就可以加载各种包,从基础的 js 到功能界面无所不包,包含了很多常用的包,足够使用。

Dojo 是重量级框架,适合企业级开发,并且异步 js 模块加载机制避免了企业级开发中大量加载 js 导致浏览器崩溃的问题。本身提供的 UI 组件完全可以满足所有需求,不需要第三方框架。而国内常用的 jQuery 则是一个很灵活很轻量级的框架,适合轻量级的 web 应用开发,只提供 js 的优化操作,不提供 UI 组件,需要借助第三方框架实现。

Dojo 具有以下几个优势:

- (1) 本身是开源的,方便扩展,而且全部东西都是免费的,比较经济;
- (2) 有专门的团队和大公司的支持,控件很多,文档齐全,全世界有很多志愿者贡献代码,生命力强大;
- (3) 提供的 UI 组件,可以很方便使用,不需要第三方框架,并且编写自己的 UI 也很方便;
- (4) 面向对象设计模式,代码层次清晰,便于调试管理。

Dojo 也有一些缺点:

- (1) 学习曲线很高,很庞大;
- (2) 教程全英文,学习起来比较困难。

2.5.2 后端框架 SSH

SSH 是 Spring, SpringMVC 和 Hibernate 的集成框架,是用来做 web 应用程序的后端开源框架。Spring 管理对象的实例化,把对象的创建和获取放到外部; SpringMVC 实现了 web 分层,并负责处理请求和视图管理; Hibernate 用于进行数据库操作。

Spring 是一个全方位的整合框架,在项目中对 Hibernate 和 SpringMVC 进行整合,解决层与层之间的耦合问题。核心是控制反转 (IOC) 和面向切面编程 (AOP), IOC 使得创建对象不需要自己去 new,而是交给 IOC 容器实例化,并通过依赖注入方式注入对象属性。AOP 将业务逻辑和事务分离,应用对象只负

责完成业务逻辑，而不需关心事务的处理。**Spring** 是一个轻量级的容器框架。

SpringMVC 是 **Spring** 框架的一个模块，无需通过中间整合层即可与 **Spring** 进行整合，是一个基于 **MVC** 模式的请求驱动类型的轻量级 **web** 框架，解决了前端页面和后端逻辑的分离。它将用户发出的所有请求发送到前端控制器，前端控制器调用各个处理器完成请求并返回给用户。

Hibernate 应用于数据持久化层，是对 **JDBC** 的轻量级封装，是一种对象关系映射工具，将 **Java** 类与数据库表建立映射关系，可以自动生成 **SQL** 语句，是一个全自动框架，使得 **Java** 程序员可以使用面向对象编程思维操作数据库。

2.6 本章小结

本章首先对推荐系统做了简单介绍。接着介绍了几种推荐系统领域最基本的推荐算法，包括算法原理和优缺点，并将传统的推荐算法过程中涉及到的两个关键问题：相似度计算和推荐算法评估，进行了解释和说明。最后，介绍了系统开发所需的前后端框架。

第3章 神经图模型试题推荐算法研究

本章旨在针对在线考试系统中推荐练习模块，研究一种个性化的推荐算法，从而完成试题推荐功能。本章是本文的核心算法部分，首先研究了神经图协同过滤算法（NGCF）。然后在此基础上，借鉴了神经协同过滤算法（NCF）用多层神经网络代替内积的思想，并引入注意力机制（Attention），对 NGCF 进行了改进，提出新的算法模型 NGCF-Att。

3.1 神经图协同过滤算法

神经图协同过滤算法（NGCF）是 Wang^[32]等人在 2019 年发表的论文中提出的算法。算法基于隐式反馈数据，核心思想是将深度学习技术应用在推荐系统算法当中，并且利用了用户-物品交互图的高阶连通性，提取出用户和物品更深层次的特征，用来增强用户和物品的 Embedding 特征表示。相比传统协同过滤算法，更准确的表示了用户和物品，从而提升算法性能。

3.1.1 高阶连通性的概念

为了提取协同信号，可以利用用户和物品交互的高阶连通图结构来实现。高阶连通示意图如图 3.1 所示。

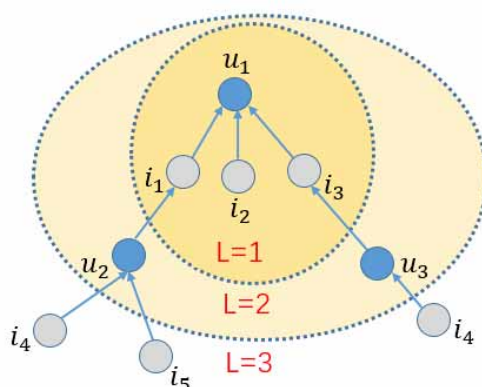


图 3.1 u_1 的高阶连通性^[32]

图 3.1 显示的是 u_1 展开的树形结构。高阶连通性指的是从任何节点到达 u_1 且

长度大于等于 1 的路径，这种结构的连通性包含了丰富的协同信号。例如，路径 $u_1 \leftarrow i_2 \leftarrow u_2$ 包含了用户 u_1 和 u_2 的行为相似性，因为都与 i_2 产生了交互；而路径 $u_1 \leftarrow i_2 \leftarrow u_2 \leftarrow i_4$ 则暗示了 u_1 具有与 i_4 发生交互的可能性，因为和 u_1 相似的用户 u_2 与 i_4 发生过交互行为。此外，总的来说 u_1 更有可能与 i_4 发生交互行为比 i_5 ，因为 i_4 有两条到达 u_1 的路径，而 i_5 只有一条。

可以看出，高阶连通结构中包含了很多有价值的信息。因此，将其利用起来集成到模型表示中，具有重要的意义。

3.1.2 神经图协同过滤模型

神经图协同过滤模型设计了一种神经网络方法传播进行递归，从而实现将高阶连通信息集成到 Embedding 中。具体讲，就是通过设计一个 Embedding 传播层，主要负责将交互信息融合到用户和物品的 Embedding 中，然后通过增加多个这样的传播层，就可以捕获到高阶连通的协同信号了。图 3.1 中，通过增加两层传播层，可以捕获 u_1 和 u_2 的行为相似性。增加三层，可以捕获 u_1 对 i_4 的潜在兴趣的信息。

神经图协同过滤模型主要包含三个部分：一个 Embedding 层，多个 Embedding 传播层和一个预测层。模型的整体结构如图 3.2 所示。

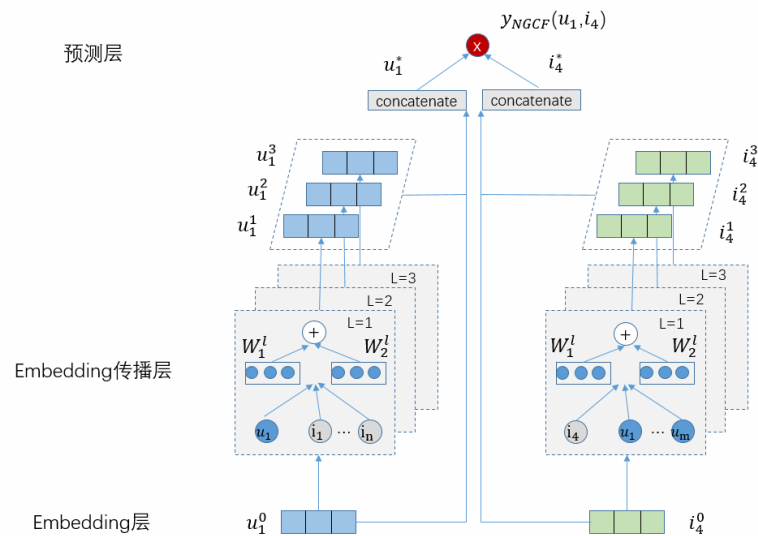


图 3.2 神经图协同过滤模型结构^[32]

(1) Embedding 层。主要是将用户和物品的稀疏特征向量映射成稠密型，并产生初始的用户和物品 Embedding。可以构造一个矩阵，作为 Embedding 查找

表，如公式 3.1:

$$E = [e_{u_1}, \dots, e_{u_N}, e_{i_1}, \dots, e_{i_M}] \quad (3.1)$$

其中, $e_u \in R^d$ ($e_i \in R^d$), 表示用户和物品的初始 Embedding, d 表示维度。上面集合中, 前 N 项表示用户, 后 M 项表示物品。

(2) Embedding 传播层。基于 GNN 的消息传递结构^{[46][47]}, 沿着高阶连通图捕获协同过滤信号并优化 Embedding。每一层传播层包含两个阶段: 消息构建和消息合成。

1) 消息构建。对于用户 u , 与发生过交互的物品 i , 构成用用户-物品对 (u, i) , 定义从 i 到 u 的消息为公式 3.2:

$$m_{u \leftarrow i} = \frac{1}{\sqrt{|N_u||N_i|}}(W_1 e_i + W_2(e_i \odot e_u)) \quad (3.2)$$

其中, $\frac{1}{\sqrt{|N_u||N_i|}}$ 是拉普拉斯范数, 表示传播过程的衰减因子, N_u 和 N_i 分别表示用户 u 发生过交互的物品数和与物品 i 发生过交互的用户数, W_1, W_2 表示两个权重矩阵, 是模型需要学习的参数, e_i 和 e_u 表示初始的 Embedding, \odot 表示向量逐个元素的乘积运算符。

2) 消息合成。为了获取传播之后优化的用户 u 的 Embedding, 需要将用户

u 发生过交互的所有物品到 u 的消息合成, 合成函数定义如公式 3.3:

$$e_u^{(1)} = \text{LeakyRelu}(m_{u \leftarrow u} + \sum_{i \in N_u} m_{u \leftarrow i}) \quad (3.3)$$

其中, $e_u^{(1)}$ 表示在经过一层传播之后用户 u 的 Embedding, LeakyRelu 为激活函数, $m_{u \leftarrow u}$ 表示自连接信号保留了传播之前的特征, $m_{u \leftarrow u} = W_1 e_u$ 。

与上面获取用户 u 的 Embedding 类似, 也可以获取到经过一层传播之后物品 i 的 Embedding 表示 $e_i^{(1)}$, 同样为以上两个阶段。

基于一层传播层，可以添加更多的传播层来探索高阶的连通信息，第 L 层的用户 u 的 Embedding 表示如公式 3.4:

$$e_u^{(L)} = \text{LeakyRelu}(m_{u \leftarrow u}^L + \sum_{i \in N_u} m_{u \leftarrow i}^L) \dots\dots\dots (3.4)$$

其中,

$$\begin{cases} m_{u \leftarrow i}^L = \frac{1}{\sqrt{|N_u||N_i|}}(W_1^L e_i^{(L-1)} + W_2^L(e_i^{(L-1)} \odot e_u^{(L-1)})), \\ m_{u \leftarrow u}^L = W_1^L e_u^{(L-1)}, \dots\dots\dots \end{cases} (3.5)$$

公式 3.5 中, W_1^L, W_2^L 表示第 L 层传播层的权重矩阵, 也是模型需要学习的参数。 $e_i^{(L-1)}$ 和 $e_u^{(L-1)}$ 表示经过第 $L-1$ 层传播层之后获得的物品和用户 Embedding。可以看出, 每一层都是基于上一层的输出作为输入, 是一个递归的实现形式。同样也可以获得物品 i 第 L 层传播之后的表示 $e_i^{(L)}$ 。

(3) 预测层。在经过 L 层传播之后, 可以获得各层用户 u 的表示 $\{e_u^1, \dots, e_u^L\}$, 而每层获得的表示从不同层面表现了用户 u 的偏好。因此, 需要将各层获得的表示连接起来, 构成最终的 Embedding。对物品 i 也是同相同的操作。最终的形式如公式 3.6:

$$e_u^* = e_u^0 || \dots || e_u^L, \quad e_i^* = e_i^0 || \dots || e_i^L \dots\dots\dots (3.6)$$

最后将上面最终的用户 u 和物品 i 的 Embedding 进行内积, 计算出评分。

3.2 神经图协同过滤算法改进

3.2.1 传播层改进

注意力机制 (Attention) 开始提出为了解决机器翻译问题, 如今已经发展成为神经网络中非常热门的研究方向, 尤其在计算机视觉和自然语言处理领域。它主要模拟了人对环境的生理感知, 合理地假设人类的视觉不会立即识别处理整个影像的信息, 而是只专注于整个感知空间需要的那部分信息, 然后进行分析, 暂时忽略掉无关信息。在文本翻译任务中, 下一个预测值可能仅与输入的

单词序列的部分单词有关，因此这些相关的单词应该赋予更高的权重。注意力模型具有直观、可解释性强的特点，因此将注意力机制引入到模型当中，是一种不错的方案，可以提高神经网络的效率。此外，目前很多关于神经网络的研究中，也都融合了注意力机制，例如 pooling 操作就可近似地看作是一种注意力机制。

注意力机制是机器学习的一种数据处理方法，是机器对人处理信息的一种模拟。人在阅读文字和浏览图片时，大脑总会专注于获取有用的信息，而舍弃掉一些次要的信息，注意力不是均衡地分配给每一个文字和像素。计算机如果不能模拟这种情景，将很多不重要甚至无关的信息予以考虑，与重要信息同等处理，就可能会干扰结果，与实际情况形成很大偏差。注意力机制可以教会计算机关联上下文，选择性地保留和舍弃信息，从而更加适应真实场景。

注意力机制没有严格意义上的数学定义，是一种方法论，在神经网络算法的研究中，注意力机制可以理解为一种神经网络隐藏层的加权模型。NGCF 模型中，充分考虑和利用了 user 和 item 的交互信息，通过传播层将这些信息显式地编码，最终获取的 Embedding 增强了 user 和 item 本身的特征表示。但是，NGCF 在对传播层进行消息构建时，没有考虑每一个消息的重要程度，而是直接将所有的消息进行累加集成，相当于为每一个消息赋予了相同且为 1 的权重。事实上，对于一个 user 的 Embedding 传播过程中，每一个消息是与 user 发生过交互行为的 item 信息，而 user 对于每一个 item 并不一定是具有相同的偏好程度，所以应当给予不同的关注，赋予不同的注意力权重，这样更符合实际情况。

因此，本文在 NGCF 的传播层引入了注意力机制，在消息的构建中，添加了一个可变权重，每一个消息具有不同的权重，权重大小表示偏好程度，权重本身取决于 user 和 item 的 Embedding 信息。

3.2.2 预测层改进

NGCF 和传统的矩阵分解 (MF) 在预测层的表达类似，利用用户和物品的特征向量 p_u 和 q_i 的内积表示用户 u 和物品 i 的交互，如公式 3.7:

$$\hat{y}_{ui} = f(u, i | p_u, q_i) = p_u^T q_i = \sum_{k=1}^K p_{uk} q_{ik} \dots \dots \dots (3.7)$$

其中, K 表示特征向量的维度, 用户和物品的特征向量具有相同的维度。

所以, 内积的交互方式是把用户和物品的特征看成是各个维度之间相互独立的, 并且权重相同, 这样看起来更像是一个线性特征模型。但简单的内积实际上会限制模型的表现, 如图 3.3 所示。

	i_1	i_2	i_3	i_4	i_5
u_1	1	1	1	0	1
u_2	0	1	1	0	0
u_3	0	1	1	1	0
u_4	1	0	1	1	1

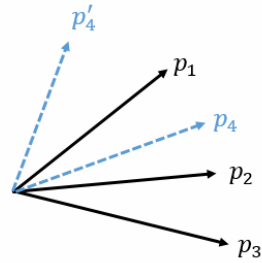


图 3.3 a 用户-物品矩阵^[28]

图 3.3b 用户特征向量空间^[28]

由于用户和物品映射到了相同的特征空间, 所有可以用内积或者余弦值衡量用户相似度。但通常使用 Jaccard 系数作为 MF 模型两个用户之间的真实相似性。图 3.3a 中, 可方便计算出 u_2 和 u_3 的相似性 s_{23} 为 0.66。同理, s_{12} 为 0.5, s_{13} 为 0.4, 则 $s_{23} > s_{12} > s_{13}$ 。这样, u_1 、 u_2 、 u_3 这三个用户的特征向量 p_1 、 p_2 、 p_3 在特征空间中应该具有图 3.3b 的关系。如果再考虑到用户 u_4 , 可类似计算出, $s_{41} > s_{43} > s_{42}$ 。也就是说 u_4 和 u_1 最相似, 然后依次 u_3 和 u_2 。但图 3.3b 中, 将 p_4 最靠近 p_1 后 (虚线表示), 无论如何也没办法使得 p_3 比 p_2 更靠近 p_4 。

因此, 内积表示低维空间中的用户-物品交互, 具有一定局限性。可以通过增加特征向量的维度来解决, 但是又会带来数据过拟合的问题, 尤其是在数据稀疏^[48]的情况下。

神经协同过滤算法 (NCF) 是 He^[28] 等人在 2017 年发表的论文中提出的算法。算法基于隐式反馈数据, 在传统的协同过滤基础上, 利用多层神经网络对用户和物品的交互进行建模, 代替了传统的内积形式。神经网络是被数学证明了的可以逼近任意连续函数的结构, 规避了内积所存在的局限性。神经协同过滤框

架的结构如图 3.4 所示。

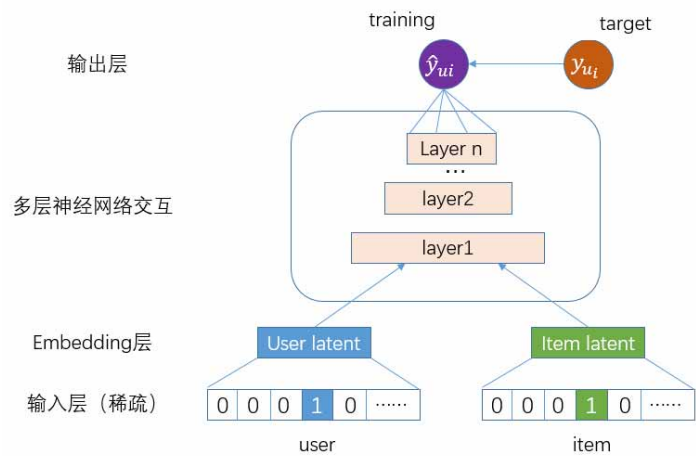


图 3.4 神经协同过滤通用框架^[28]

NGCF 和 MF 会在获得 user 和 item 的最终 Embedding 之后，直接在预测层将两者做内积，计算出评分。而上图 3.4 的结构中，首先将 User latent 和 Item latent 进行结合，合并成一个向量，然后经过一个塔式结构的多层神经网络，最后与输出层相连接。输出层只有一个神经元，就是用户对物品的评分。其中合并的方式有多种，通常包括直接连接和逐个元素乘积等方式。

本文借鉴了 NCF 的结构和思想，在 NGCF 的预测层，首先将获取到的最终的 user 和 item 的 Embedding 直接连接作为神经网络的输入，然后设置多层神经网络，最终到输出层的一个神经元作为输出，获得得分。这样就将 NGCF 预测层的内积替换成了神经网络的形式。

3.2.3 NGCF-Att 模型

以上两个小节中，本文分别在 NGCF 传播层和预测层的基础上，对 NGCF 做了改进。首先，相比于传统的一些协同过滤算法，本文基于的 NGCF 通过交互信息的显式编码，增强了 user 和 item 的 Embedding 表示。其次，本文在 NGCF 传播层引入了注意力机制，对用户关联的物品进行了可变权重的学习，代替了原先的相等权重或启发式权重，使得模型具有更高的效率和更好的解释性。最后，本文在 NGCF 预测层采用神经网络代替内积，捕获了 user 和 item 之间复杂的非线性交互关系，提升了模型准确率。最终本文提出了一种注意力（Attention）模型结构 NGCF-Att，如图 3.5 所示。

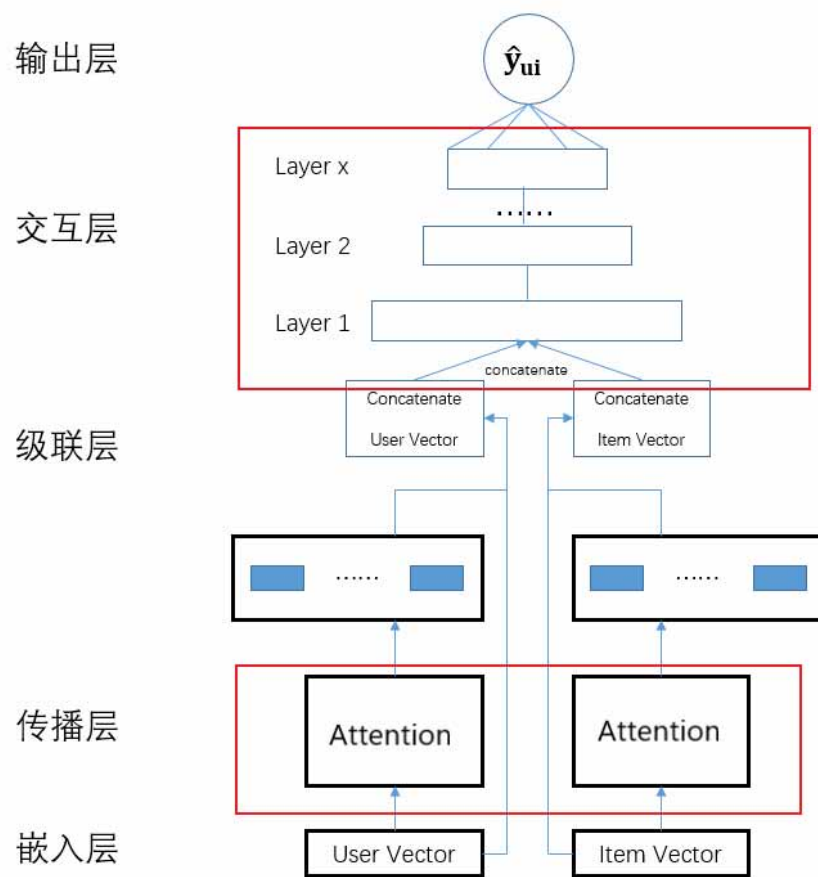


图 3.5 NGCF-Att 模型结构

(1) 嵌入层

对用户和物品的特征表示初始化成向量的表示形式，同时相当于构建起一个用户和物品的 Embedding 查询表。如前面公式 3.1

(2) 传播层

在 NGCF 原来的规则中，认为所有存在交互行为的都能表示偏好，并且权重相同。但是事实上，一组正面的反馈并不一定表示偏好，可能受其他因素的影响。而且，一个用户对所有存在交互的物品也不一定具有完全相同的偏好。所以对于不同物品，应该给予不同的关注。注意力机制可以很好地学习可变权重，将其分配给不同的物品或者不同成分。权重的高低反应相关性和偏好程度的大小。

1、在消息构建阶段，将消息重新定义为公式 3.8:

$$m_{u \leftarrow i} = \frac{1}{\sqrt{|N_u| |N_i|}} (W_1 \alpha_i e_i + W_2 (\alpha_i e_i \odot e_u)) \dots\dots\dots (3.8)$$

其中, $\frac{1}{\sqrt{N_u N_i}}$ 是拉普拉斯范数, 表示传播过程的衰减因子, N_u 和 N_i 分别表

示用户 u 发生过交互的物品数和与物品 i 发生过交互的用户数, α_i 是可变权重,

W_1, W_2 表示两个权重矩阵, 是模型需要学习的参数, e_i 和 e_u 表示初始的

Embedding, \odot 表示向量逐个元素的乘积运算符。

上面消息构建的过程相比于 NGCF 模型, 为不同交互物品 i 传递的消息增加了一个权重 α_i , 可变权重 α_i 表示了用户对物品有不同的偏好程度。 α_i 通过用户 e_u 和物品 e_i 计算获得, 如公式 3.9:

$$\alpha_i = w_1^T \varphi(W_3 e_u + W_4 e_i + b_3) + b_4 \dots \dots \dots (3.9)$$

其中 W_3, W_4 和 b_3 代表第一层的权重参数, w_1 和 b_4 代表第二层的权重参数, $\varphi(x) = \max(0, x)$ 表示 ReLU 激活函数。上式的计算相当于用户 u 对物品 i 的打分, 由网络自动学习获得。

最后需要将上面的 α_i 通过 Softmax 进行正则化, 获得最终的 α_i , 计算如公式 3.10:

$$\alpha_i = \frac{\exp(\alpha_i)}{\sum_{j \in N_i} \exp(\alpha_j)} \dots \dots \dots (3.10)$$

2、然后遵循 NGCF, 在消息合成阶段将所有的消息进行合并, 获得经过一层传播层之后优化的用户 Embedding, 如前面公式 3.3。最终通过多层传播, 获得各层的用户的 Embedding, 如前面公式 3.4 和 3.5。同样的计算获得物品的各层的 Embedding。

(3) 级联层

将各层传播层获取到的用户和物品 Embedding 进行连接, 最简单的方式就是直接连接, 也可以采取别的方式, 如加权等。但是直接连接不需要训练额外的

参数，获取到连接之后的最终表示 e_u^* 和 e_i^* ，如前面公式 3.6。

(4) 交互层

塔式结构的多层全连接网络，首先将 e_u^* 和 e_i^* 进行连接，然后经过 L 层神经网络，代替了内积，捕获非线性关系，如公式 3.11。

$$\varphi = a_L \left(h_L^T \left(a_{L-1} \left(\dots a_2 \left(h_2^T \begin{bmatrix} e_u^* \\ e_i^* \end{bmatrix} + b_2 \right) \dots \right) + b_L \right) \right) \dots \dots \dots (3.11)$$

其中， h_n^T 为各层神经网络的权重， b_n 为各层神经网络的偏置， a_n 为各层神经网络的激活函数， φ 为输出层之前最后一层神经网络。

(5) 输出层

只有一个神经元，将交互层的最后一层与该神经元进行全连接，最终得到一个结果，就是用户对物品的评分 \hat{y}_{ui} ，如公式 3.12。

$$\hat{y}_{ui} = a_{out}(h^T \varphi) \dots \dots \dots (3.12)$$

其中， h^T 为输出层权重， a_{out} 为输出层激活函数。

3.3 本章小结

本章首先详细介绍了 NGCF，然后对 NGCF 算法进行了改进。在 NCF 算法的启发下，用多层神经网络取代了 NGCF 预测层的用户物品内积交互形式，规避了内积的局限性，捕获了用户物品之间复杂的非线性交互关系。然后在 NGCF 传播层引入了注意力机制（Attention），提升算法性能的同时，使得模型具有更好的可解释性。最终提出一种新的模型结构 NGCF-Att。

第4章 实验验证及结果分析

上一章主要针对系统的推荐练习功能模块，对个性化推荐算法进行了详细的研究。基于 NGCF 算法，提出了一种改进的模型 NGCF-Att。本章针对提出的新模型进行了实验验证，并与相关的几种模型进行了结果的对比，最终得出结论。

4.1 实验环境与数据集

4.1.1 实验环境

本文研究的算法基于深度学习，使用深度学习框架 Tensorflow 来实现，由于实验数据集规模较大，考虑到时间和性能的问题，使用实验室服务器来进行实验，并且主要运行在 GPU 上，Tensorflow 用的是 GPU 版本。实验环境如表 4.1 所示。

表 4.1 实验环境

处理器	AMD Ryzen 5 2600 Six-Core Processor 3.40GHz
内存	32.0 GB
操作系统	Windows 10 专业版 64 位操作系统，基于 x64 的处理器
显卡	NVIDIA GeForce RTX 2080
显示内存	8010 MB
实验工具	JetBrains PyCharm 2019.2.3
编程语言	Python

4.1.2 实验数据集

1、数据集选取

为了完成试题的推荐功能，首先需要进行数据的获取。数据可以从考试系统数据库错题记录表中获取，需要收集学生 id、题目 id、题目是否做错（0 或 1）

这些所有学生的错题记录数据。本文在考试系统推荐练习模块，预留了推荐数据接口，可从数据库获取数据并调用推荐算法。但是由于本文实现的系统原型仍在测试阶段，真实学生做题记录数据正在搜集过程中，不能获取到大量的数据。为了保证推荐算法的准确性和有效性，本文选取了可映射到系统原型数据库结构的 Amazon-book 和 Gowalla 两套公开数据集进行实验来验证推荐算法的效果。

数据集选取说明如下：

- (1) 这两个数据集公开可用，并且被广泛用于推荐算法的研究。
- (2) 事实上，本文的推荐算法专注于推荐功能的实现和算法推荐的准确性，不针对于某个具体的场景。如表 4.2 所示。

表 4.2 收集数据格式

数据集	收集数据格式
学生错题数据	学生 id、题目 id、是否做错
Gowalla	用户 id、位置 id、是否签到
Amazon-book	用户 id、图书 id、是否购买

上表中展示了系统中真实学生错题的记录数据格式和两个数据集的记录格式。对于每一个数据集，为了进行算法实验，需要首先进行数据处理。本文算法所需数据格式如图 4.1 所示。

```

0 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
1 127 128 129 130 131 132 133 134 135 136 137 138 139
2 176 177 178 179 180 181 182 183 184 185 186 187 188
3 259 260 244 243 261 262 263 264 265 266 246 267 268
4 192 205 277 176 278 279 280 281 282 283 284 285 286
5 290 208 277 291 292 293 294 295 296 297 206 298 176
4 535 536 537 538 539 540 541 160 542 543 544 204 192
6 176 294 295 324 554 555 556 197 525 557 374 373 293
7 571 572 573 543 574 575 576 577 578 579 580 581 582
8 372 619 205 620 621 622 623 624 625 626 627 628 629

```

图 4.1 数据处理之后的文本格式

上图的文本格式中，每一行第一个数字对应学生 id，每一行除了第一个数字之外的其他所有数字对应学生做错的题目 id 集合。对于选取的两个数据集来说，也是同样的对应关系。

可以看出，算法本身不关注具体的场景内容信息，而关注的是这种格式和对应关系。在这种意义下，学生错题记录数据和选取的两个实验数据集格式是完全一致的。在缺少真实学生用户错题记录数据的情况下，可将选取的两个公开推荐算法数据集当成学生错题记录数据使用，来验证算法本身的推荐功能和准确性。

2、数据集介绍

两个数据集的大小和稀疏性等各不相同，二者的统计数据如表 4.3 所示。

表 4.3 数据集统计信息

Dataset	Users	Items	Interactions	Density
Gowalla	29858	40981	1027370	0.00084
Amazon-book	52643	91599	2984108	0.00062

(1) Gowalla 数据集：此数据集是一个签到数据集，来源于 Gowalla 网站，记录社交网站用户签到的时间和位置信息，包括用户 ID，签到时间，纬度，经度和位置 ID 等信息，用户可通过签到可共享他们的位置。

(2) Amazon-book 数据集：此数据集是一个被广泛使用作为产品推荐的数据集，来自于 Amazon 的产品评论和元数据，在相关实验数据集网页上收集，主要记录的是用户对图书的评分数据，包括用户 ID，物品 ID，评分和时间戳等信息。

为了保证数据集的质量，两个数据集都进行了 10-core 提取，保证每个用户和每个物品至少都有 10 次交互，去掉少于 10 次交互的用户和物品。例如，在 Gowalla 数据集中，交互表示用户签到过的位置，保留签到过 10 个位置以上的用户和至少被签到过 10 次的位置；而同理在 Amazon-book 数据集中，交互表示用户购买过图书。对于每个数据集，随机选择每个用户 80% 的历史交互作为训练集，其余的为测试集。在训练集中，对于可以观察到的用户-物品交互视为正例，训练过程中，每个正例都会采用负采样策略为其配对一个此用户之前没有进行过交互的物品。

模型基于用户隐式反馈数据进行研究，即用户对物品有过历史交互行为可记为 1，没有记为 0。需要对数据集进行简单的预处理，保留用户 ID 和物品

ID。对于每个数据集，最终形成四个文本文件，分别为 train.txt，test.txt，user_list.txt，item_list.txt。

user_list.txt 是对原始数据集的用户 ID 进行了映射，将原先不连续的用户 ID 映射成了一组从 0 开始的连续自然数序列。item_list.txt 是对物品 ID 进行映射，操作同用户一样。文件格式如图 4.2 所示。

```
org_id remap_id
343 0
5258 1
3973 2
5254 3
6686 4
138 5
253 6
254 7
494 8
221 9
3229 10
4676 11
24330 12
```

图 4.2 ID 映射截图

train.txt 和 test.txt 分别是训练集和测试集，格式为每行第一个 ID 为用户 ID，后面所有的 ID 为用户产生过交互行为的物品 ID 集合。文件格式如前面提到的图 4.1 所示。

4.2 模型训练

第三章中详细介绍了推荐算法模型研究，并且对改进的算法模型结构和思想以及算法过程都做了详尽的说明。

对于数据集的划分，上一小节中也提到了，随机选择每个用户 80% 的历史交互作为训练集，剩下的 20% 交互作为测试集。

模型基于 NGCF 进行改进。首先初始化用户和物品 Embedding，并在传播层引入注意力机制。然后，对交互信息采用了可变权重加权平均的结合方式，经过多层传播层之后，获得各层优化之后的 Embedding。最后，将这些 Embedding 连同初始的 Embedding，通过级联层进行连接，获得了用户和物品的最终表示 e_u^* 和 e_i^* 。最终将 e_u^* 和 e_i^* 进行连接，通过多层全连接网络层，输出用户 u 对物品 i 的评分。

本文的目标是使得这个评分的预测值与真实值之间的差距尽可能的小。为

了学习模型参数，此外考虑到本文的主要任务是进行 TopN 推荐，这里优化 BPR^[49] 损失，即贝叶斯个性化排序算法。它采用了目前比较广泛使用的成对损失 pairwise 的思想。具体来说，BPR 主要考虑了观察到的用户物品交互和未观察到的用户物品交互之间的相对顺序，它假设观察到的交互行为更能反映用户的偏好，应该分配比未观察到的交互行为更高的预测评分值。BPR 损失函数如公式 4.1:

$$L_{\text{loss}} = \sum_{(u,i,j) \in O} -\ln \sigma(y_{u_i} - y_{u_j}) + \lambda ||\Theta||_2^2 \quad (4.1)$$

其中， $O = \{(u,i,j) | (u,i) \in R^+, (u,j) \in R^-\}$ 表示成对的训练数据， R^+ 是观察到的交互， R^- 是未观察到的交互， σ 是 sigmoid 函数， Θ 表示所有的需要学习的训练参数， λ 是正则项系数。模型使用 Adam^[50] 来优化损失，更新模型参数。

4.3 评估指标及超参数设置

4.3.1 评估指标

对于测试集中的所有用户，未观察到用户交互的物品视为负例，本文将输出用户在除了训练集中的正例物品之外的其他所有物品上的评分，然后选出前 K 个评分最高的物品推荐给用户。为了评估这前 K 个物品排序的效果，本文主要使用了两个在推荐系统中被广泛使用的指标 recall@K 和 ndcg@K。实验中使用 K=20 来评估结果。

(1) recall: 评估推荐的 K 个物品的列表与测试集中正例的交集在测试集的正例中所占的比例，如公式 4.2:

$$\text{recall@k} = \frac{|testP \cap topK|}{|testP|} \quad (4.2)$$

其中，testP 是测试中的正例，topK 是推荐的 K 个物品。

(2) ndcg: 评估 topK 推荐列表排序质量的好坏，如公式 5.3:

$$\text{ndcg@K} = \frac{1}{Z} \sum_{i=1}^K \frac{2^{r_i} - 1}{\log_2(i+1)} \quad (4.3)$$

其中，Z 表示最理想状况下的推荐列表排序值， r_i 表示用户对推荐列表中第 i 个物品实际的评分等级。由于是隐式反馈数据，所以 r_i 的取值为 0 或 1， $r_i=1$ 说明这个物品在测试集正例中， $r_i=0$ 说明此物品不在测试集正例中。

4.3.2 超参数设置

本文在 Tensorflow 中实现了模型算法，而模型超参数的设置对于模型的性能有着很大的影响，非常关键。想要提升模型的效果，需要不断进行参数的搭配选择并经过训练调整，超参数包括初始 Embedding 的尺寸大小、传播层数量（通常不超过 5）、训练数据 batch 的大小以及学习率等。模型使用 Adam 优化器优化损失，使用 Xavier 初始化器来初始化模型的参数，并且采用了早停策略，如果连续 10 轮训练结果没有提升，就结束训练。最终选择的超参数设置如表 4.4 所示。

表 4.4 超参数设置表

参数	值	参数说明
embed_size	32	初始嵌入维度
layer_size	[32,32,32]	每层传播层嵌入维度
batch_size	512	批处理尺寸
regs	0.00001	正则项系数
lr	0.0001	学习率
node_dropout	0.1	节点丢失率
mess_dropout	[0.1,0.1,0.1]	每层消息丢失率

4.4 结果对比分析

为了验证算法模型的效果，将本文的模型 NGCF-Att 与以下的三种算法进行了比较。为了公平对比，所有算法都采用 BPR 损失函数进行优化。

（1）BPRMF：基于 BPR 损失函数优化的传统的矩阵分解模型。该模型直接训练用户和物品的 Embedding，然后将内积作为交互的评分值。最后用 BPR 损失函数优化训练。

（2）NCF：第四章提到的神经协同过滤模型。该模型将用户和物品的 Embedding 进行连接作为输入，然后使用了多层神经网络，最终输出评分。最后利用 BPR 损失函数进行训练。

(3) NGCF: 第四章提到的神经图协同过滤模型。该模型通过 Embedding 的传播, 将 Embedding 进行了优化, 捕捉到了各阶的连通性交互特征, 然后将内积作为评分值。最后使用 BPR 损失函数训练。

四种算法在 Gowalla 和 Amazon-book 两个数据集上的 recall@20 和 ndcg@20 评价指标如表 4.5 所示。

表 4.5 四种算法的结果比较

数据集	算法名称	recall@20	ndcg@20
Gowalla	BPRMF	0.1283	0.1864
	NCF	0.1317	0.1993
	NGCF	0.1502	0.2198
	NGCF-Att	0.1514	0.2211
Amazon-book	BPRMF	0.0239	0.0526
	NCF	0.0251	0.0533
	NGCF	0.0315	0.0617
	NGCF-Att	0.0327	0.0636

通过上表结果的对比可以看出, 相比传统的矩阵分解 MF 模型, 其他三种基于深度学习的模型效果会更优一些, 反映了从不同的程度上, 优化了传统 MF。其次, NGCF-Att 由于结合了 NCF 和 NGCF 两种模型的优势, 并且引入注意力机制, 提高网络效率, 所以指标会略高于它们。

因此, 我们可以得出如下结论:

(1) 基于深度学习的推荐算法的研究, 对于提升推荐系统的性能来说, 是一个很有价值的方式。

(2) 内积存在一定的局限性, 利用神经网络代替内积, 可捕获用户和物品之间复杂的非线性交互关系。

(3) 注意力机制针对每个用户, 对不同的物品赋予了不同的权重, 更加符合现实中用户对不同物品的偏好程度。可变权重的学习, 在提升模型性能的同时, 使得模型具备较好的可解释性。

4.5 本章小结

本章主要针对前面章节提出的推荐算法模型, 着手进行了实验验证, 由于

缺乏真实的用户数据，所以使用了推荐领域常用的，比较有说服力的 Gowalla 和 Amazon-book 两个公开数据集来进行实验。模型基于隐式反馈数据，经数据处理之后，对任何的推荐数据集来说，格式都是一致的，仅仅保留用户 ID 和物品 ID。最后，将本文提出的模型与相关的几种算法模型进行了实验结果的对比，验证了模型的有效性。

第5章 试题推荐考试系统的设计与实现

前面 1、2 章是本文基础部分，从研究背景、系统开发框架、推荐系统概述以及一些传统的推荐算法进行了介绍，是后续章节的理论基础，3、4 章为本文核心算法及算法实验验证。而本章主要对在线考试系统整体设计和功能实现做了详细介绍，并对系统学生端的试题练习功能模块的题目推荐练习功能进行了分析思考，明确了系统数据和推荐算法所需数据类型，分析了推荐算法应用的可行性。

5.1 系统需求分析和开发环境

5.1.1 开发环境介绍

在线考试系统在 windows 系统环境下，使用 Eclipse 作为开发工具。基于 Java 语言进行开发，开发工具包是目前比较稳定的版本 JDK1.8，同时配置了 Tomcat 服务器。Java web 开发通常搭配 Tomcat 服务器，由于 Tomcat 简单好用，而且是轻量级的，在访问量不是特别大的时候，具备不错的负载能力。系统使用 B/S 结构和 MVC 设计模式分层开发，将系统分成数据库表设计、前端数据可视化界面和后端逻辑的实现三部分。数据库使用的是 MySQL 来存取数据；前端框架使用 Dojo，是一个面向对象的 JavaScript 工具箱，包含的功能和界面很全面；后端框架使用 SSH，即 Spring，SpringMVC 和 Hibernate，是一种比较流行 Web 应用程序开源框架，实现了 MVC 模式开发。此外，使用 maven 工具来管理项目，非常方便，尤其是对项目依赖包的获取，只需要配置好路径，可自动从仓库获取，不需要人为去开源网站寻找。

5.1.2 功能需求分析

在线考试系统开发主要为了规避传统纸质化考试需要人工出卷、阅卷以及统计，工具量巨大，此外还需要大量的试卷印刷费用等缺点，通过使用网络技术实现无纸化、自动化，使得考试过程变得方便快捷高效，并且节约大量人力物力成本。所以对于考试负责人来说，在线考试系统应该具备可以根据设置的

条件，自动从题库抽取试题组成试卷，以及在学生提交试卷完成的时候自动阅卷评分等核心功能。对于学生来说，不仅可以参加考试，还可以日常做题练习，更能实现个性化的学习方式。所以考试系统，主要可以设计三类不同权限的用户：课程负责人、教师和学生，然后针对三个端的权限功能分别进行需求分析。

1、课程负责人主要功能需求

(1) 管理用户。可以对系统的教师用户进行管理和权限分配，添加和删除教师用户信息。

(2) 试题审核。对题库的试题进行维护和更新，对新上传的试题进行审核，通常由教师负责上传。

(3) 管理考试。为每一次正式考试和模拟考试进行组卷；设置每次考试的具体安排信息，包括试卷试题的组成、试题难度、考试类型、考试时间以及考生的安排；监控每次考试中实考人数、已交卷人数和未交卷人数并查看考试详情和学生得分情况。

(4) 管理课程。对所负责的课程信息进行管理，编辑和删除所负责课程包含的所有知识点信息。

根据以上对课程负责人的角色功能需求分析，主要包含 4 个用例，分别为用户管理、试题审核、考试管理和课程管理，得到课程负责人的用例如图 5.1 所示。

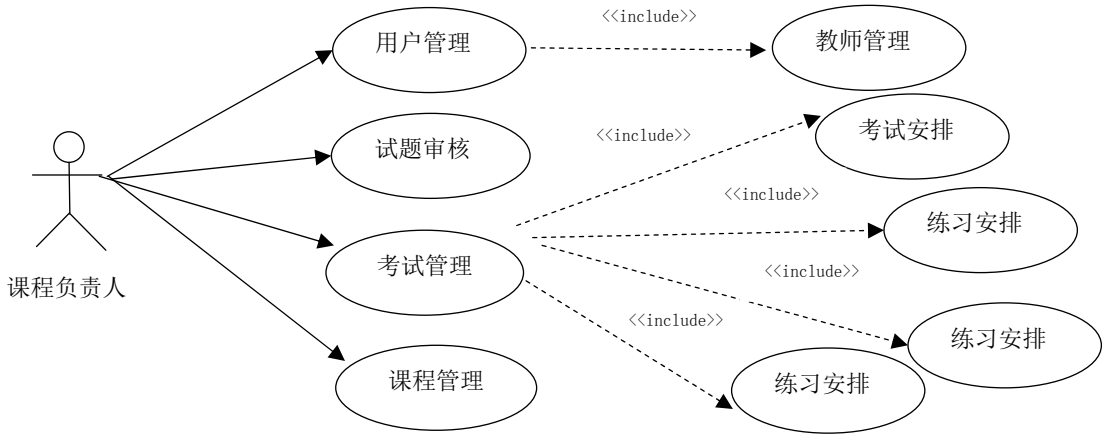


图 5.1 课程负责人用例图

2、教师主要功能需求

(1) 管理用户。可以管理系统的所有学生用户，添加和删除学生用户信

息。

(2) 上传试题。可以针对平时的教学经验和学生们的知识掌握程度，自己编辑或者发现一些优质的试题，然后保存到题库中，但需要管理员的审核。

(3) 管理考试。可以监控学生考试情况，包括实考人数和得分情况等；查看学生试卷详情，针对系统对部分主观题判断不合理的情况做出得分调整。

根据以上对教师的角色功能需求分析，主要包含 3 个用例，分别为用户管理、试题上传和考试管理，得到教师的用例如图 5.2 所示。

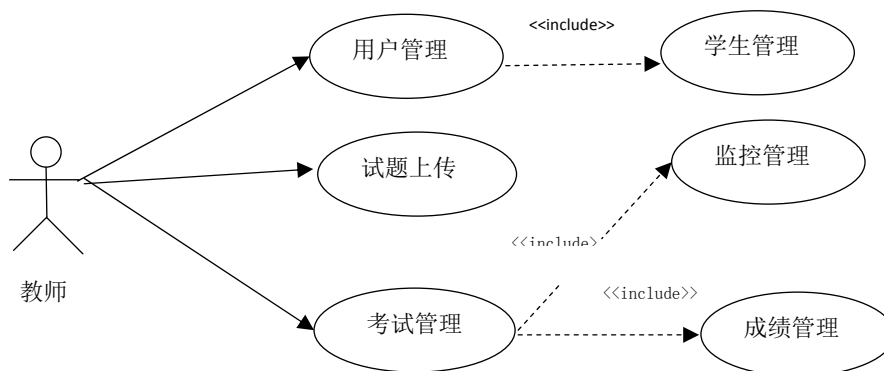


图 5.2 教师用例图

3、学生主要功能需求

(1) 在线考试。可以根据课程负责人的考试安排，按时登录考试系统进行在线考试答卷。

(2) 考试练习。根据课程的教学任务和安排，按时进行模拟考试的练习。

(3) 题目练习。学生可自行进行题目练习，针对一些错题、难题和重要知识点题目进行做题训练。

根据以上对学生的角色功能需求分析，主要包含 3 个用例，分别为参加考试、模拟考试和题目练习，得到学生的用例如图 5.3 所示。

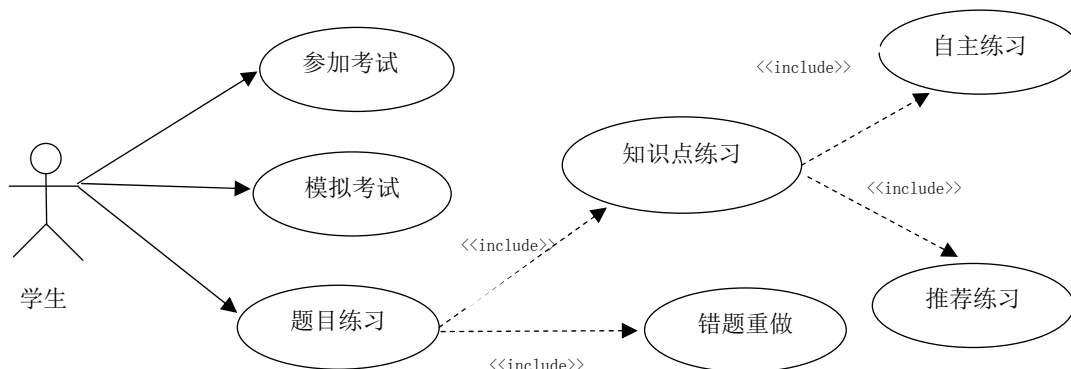


图 5.3 学生用例图

5.1.3 系统可行性分析

1、技术可行性分析

系统基于 B/S 结构，MVC 设计模式。数据库是 MySQL，而 MySQL 语句简单易学，直观明了，很容易理解，开发系统只需要掌握一些基本的 MySQL 技术，便可对数据库进行简单的增、删、改、查操作。开发整个系统使用 Java 语言和 Eclipse 工具，Java web 和 JavaEE 对于企业级项目和网站开发应用已经相当广泛和成熟。后端框架 SpringMVC 很好的实现了 MVC 的模式，而且 Java 中有专门的和数据库交互连接的工具包 JDBC，配合 Hibernate 等框架方便地实现了数据库的连接查找。此外 Eclipse 工具很适合新手练习和学习，使用起来难度不大。因此，技术方面是可行的。

2、经济可行性分析

首先，需要一台普通的电脑设备，随着技术的发展，电脑现在已经很普及，价格也很低，一台普通的电脑不需要多高成本。然后，开发工具使用 Eclipse，开发语言使用 Java，数据库使用 MySQL，网站解析服务器为 Tomcat，前端框架使用 Dojo，所有这些都是开源免费的，无需投入购买软件工具成本。最后涉及到上线问题的话，需要购买服务器部署网站，对于学校用户来说，并不需要太高配的服务器，但是能节省大量人力物力，收益肯定是大于投入的。因此，经济可行性很高。

5.2 系统概要设计

5.2.1 系统分层结构

系统基于 B/S 结构，采用 MVC 的设计模式，将系统分为三层结构，分别为数据访问层、业务逻辑层和用户界面层，顺序是从下至上，也是目前软件体系架构设计最为流行和重要的结构。分层结构是基于“高内聚低耦合”的思想。数据访问层主要存放系统数据文件和表，以便系统读取，前端展示；业务逻辑层为系统的核心部分，系统整体的功能实现主要在这一层完成和处理，并且直接与数据库交互；用户界面层将后台业务逻辑处理后的数据进行直观展现给用

户，很少一部分简单的逻辑在前端实现。

三层结构的设计，使得系统整体分工明确，提高了可维护性，大大节约了维护和更新成本。整体架构图如图 5.4 所示。

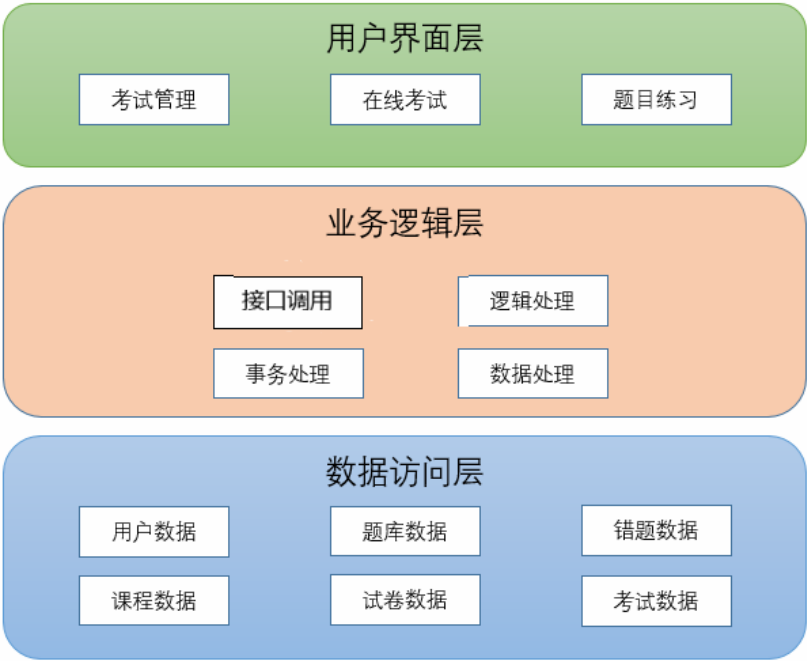


图 5.4 系统分层架构图

5.2.2 系统功能设计

根据在线考试系统整体的功能需求分析，系统主要包括课程负责人、教师和学生三类不同角色权限的用户，需要分别针对三类角色进行三个端的功能模块设计。

1、课程负责人主要负责教师用户的管理、课程的管理、考试试卷的生成、题库的更新维护以及考试的具体安排等。课程负责人端的功能设计如图 5.5 所示。

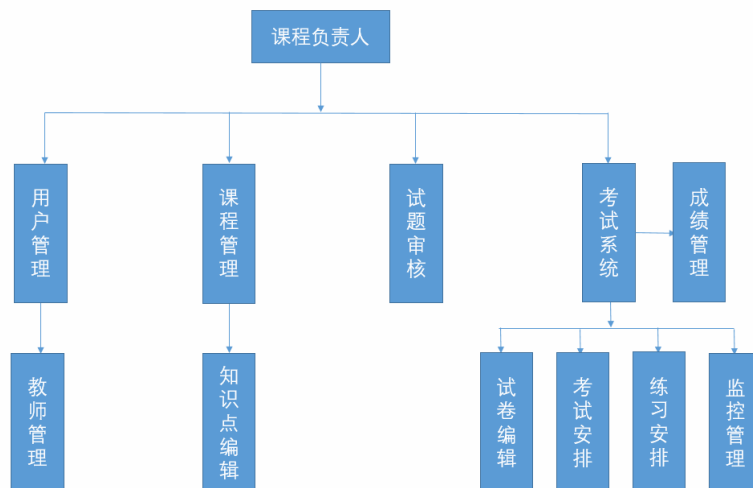


图 5.5 课程负责人功能模块

2、教师主要负责学生用户管理、学生考试成绩管理、题库试题上传和学生考试状况监控等。教师端的功能设计如图 5.6 所示。

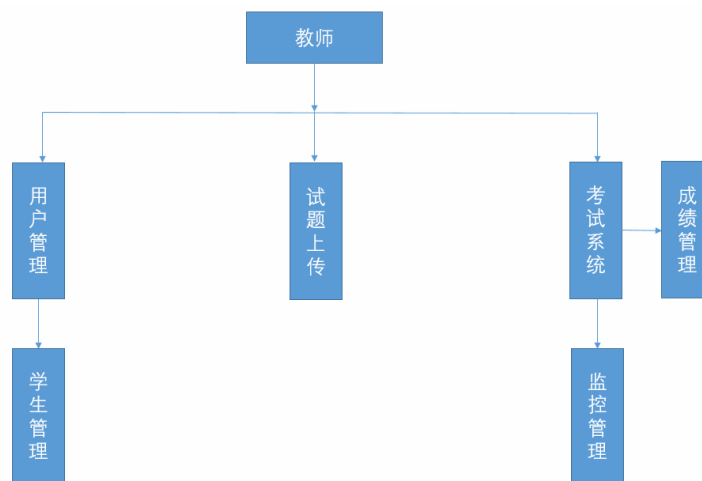


图 5.6 教师功能模块

3、学生端的主要功能是参加考试和做题练习。学生端的功能设计如图 3.7 所示。

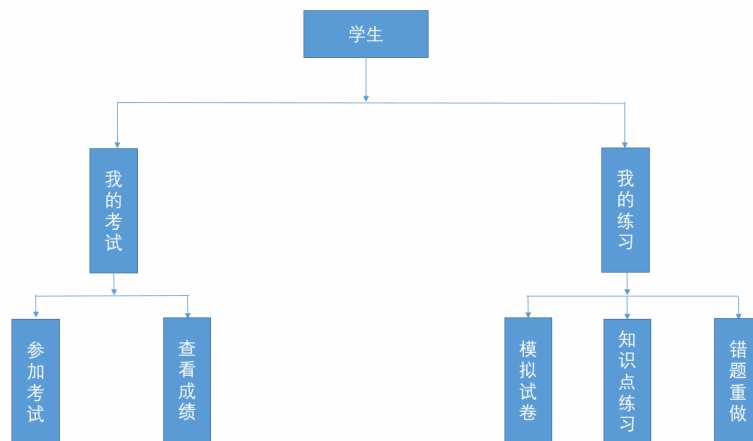


图 5.7 学生功能模块

5.2.3 数据库设计

1、E-R 图的设计

通过对系统整体的需求分析和功能设计，可以得到系统中涉及到的主要实体有用户、题库、试卷、考试记录、考试安排、课程、试卷难易规则等。

用户包含课程负责人、教师、学生，实体属性图如图 5.8 所示。

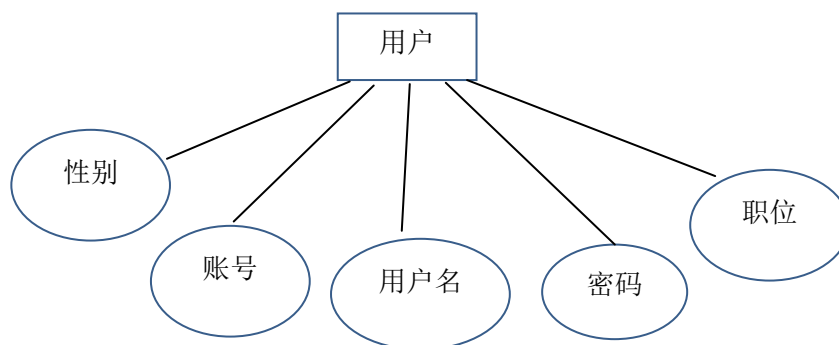


图 5.8 用户实体属性图

题库包含了所有试题题目信息，实体属性图如图 5.9 所示。

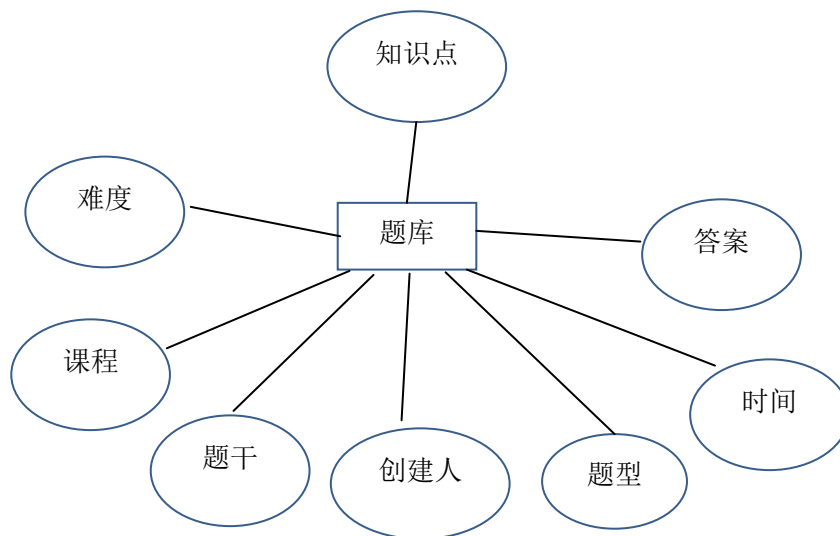


图 5.9 题库实体属性图

试卷包含试卷的分数、时间、题目数量等信息，实体属性图如图 5.10 所示。

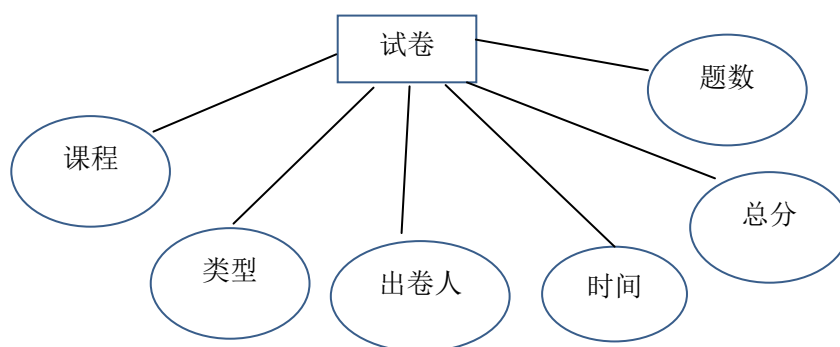


图 5.10 试卷实体属性图

考试记录包含考生、考试状态、答案、考试成绩等信息，实体属性图如图 5.11 所示。

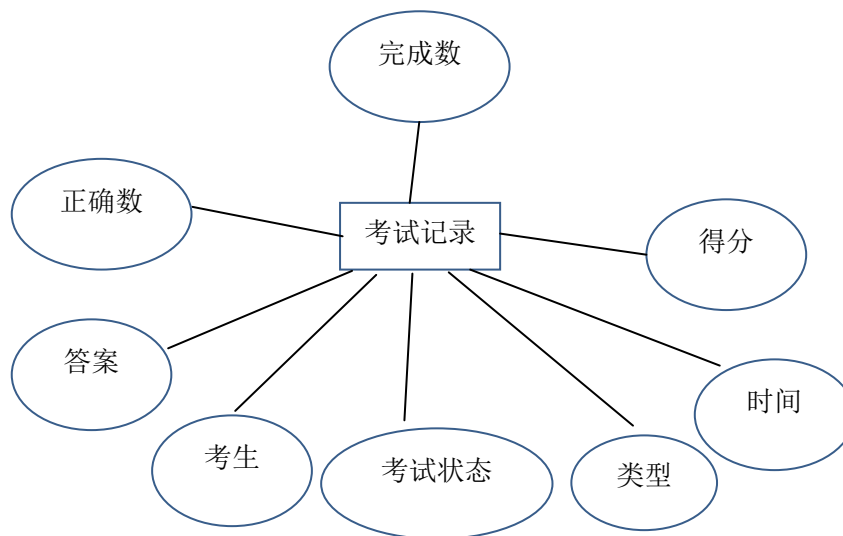


图 5.11 考试记录实体属性图

考试安排包含考试时间、考试试卷、考生范围等信息，实体属性图如图 5.12 所示。

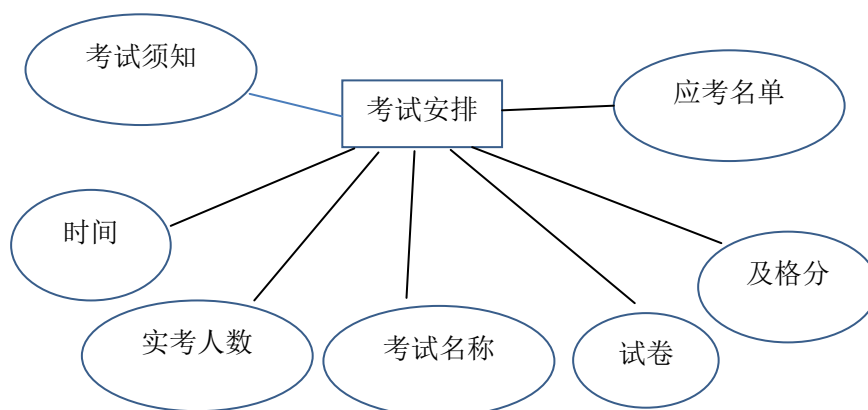


图 5.12 考试安排实体属性图

课程包含课程的基本信息，实体属性图如图 5.13 所示。

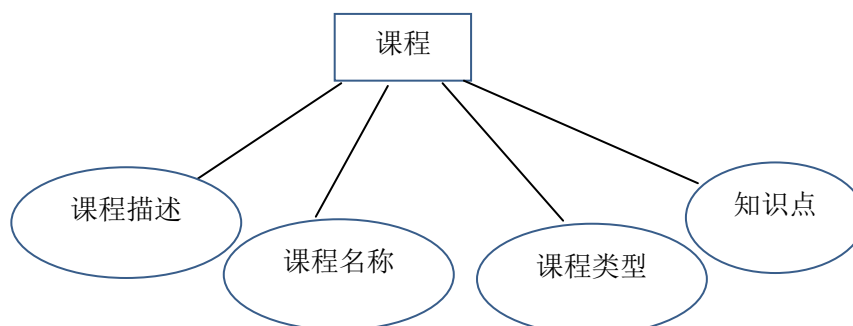


图 5.13 课程实体属性图

试卷难易规则是针对课程负责人和教师对学生考试自动组卷、随机抽题设置的规则，实体属性图如图 5.14 所示。

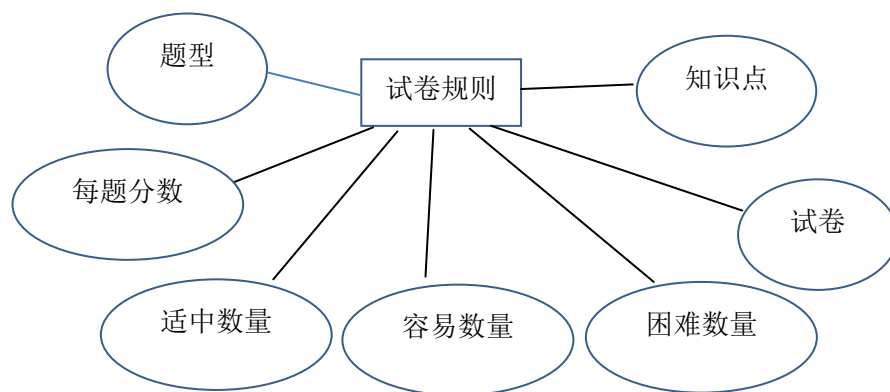


图 5.14 试卷难易规则实体属性图

根据设计的实体属性，系统整体的实体关系 E-R 图如图 5.15 所示。

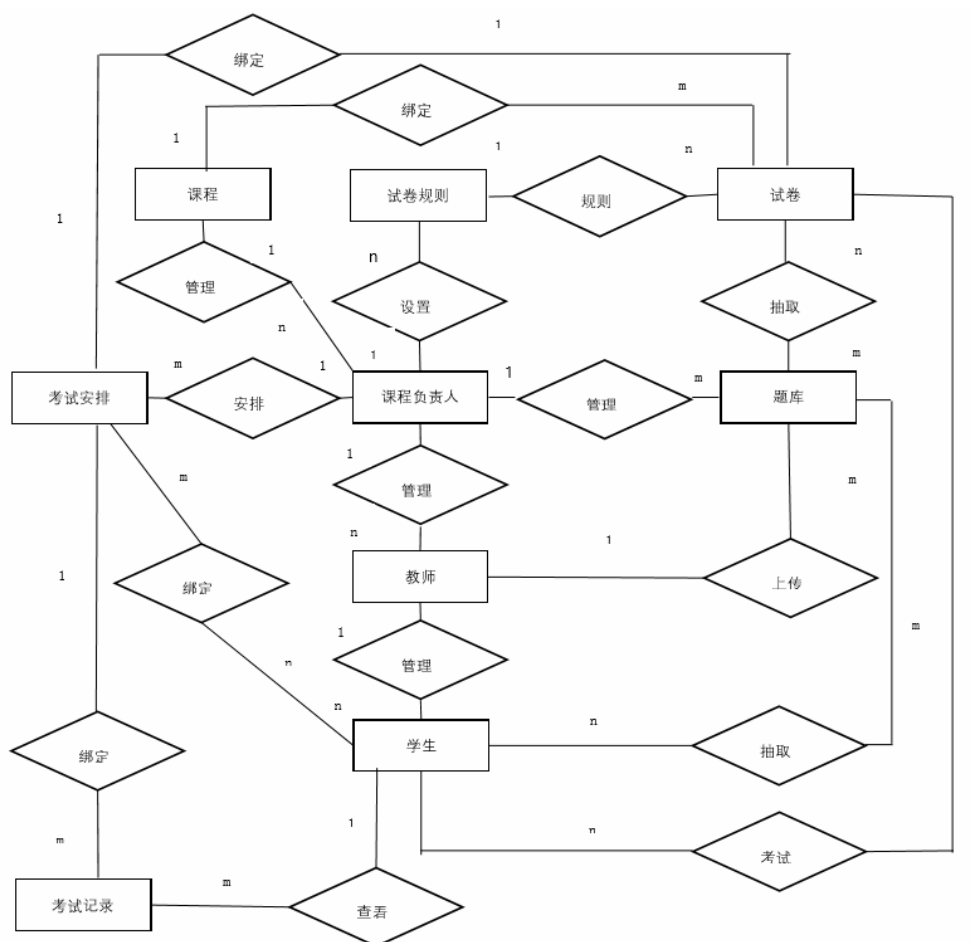


图 5.15 系统 E-R 图

2、数据库表设计

在对系统进行功能需求分析和功能模块设计的基础之上，仔细分析了在线考试系统涉及到的实体和关系，并设计了如下的 MySQL 数据库表：

(1) 题库表：用来存取所有题目的信息，表的设计如表 5.1。

表 5.1 题库表

序号	字段名称	字段类型	字段长度	允许空	主键	字段说明
1	id	varchar	255	Not null	Y	主键
2	stem	varchar	255		N	题干
3	type	varchar	255		N	题型
4	create_time	datetime	0		N	创建时间
5	subject_id	varchar	255		N	所属课程
6	difficulty	varchar	255		N	题目难度
7	bank_type	varchar	255		N	题库类型
8	examine	varchar	255		N	审核状态
9	answer	varchar	255		N	题目答案
10	user_id	varchar	255		N	创建人
11	knowledge	varchar	255		N	知识点

(2) 试卷信息表：用来存储组好的试卷信息，表的设计如表 5.2。

表 5.2 试卷信息表

序号	字段名称	字段类型	字段长度	允许空	主键	字段说明
1	id	varchar	255	Not null	Y	主键
2	type	varchar	255		N	试卷类型
3	create_time	datetime	0		N	创建时间
4	subject_id	varchar	255		N	所属课程
5	name	varchar	255		N	试卷名称
6	total_num	varchar	255		N	题目总数
7	total_score	int	11		N	试卷总分
8	user_id	varchar	255		N	组卷人

(3) 试卷题目信息表：用来存取试卷每一道题目信息，表的设计如表 5.3。

表 5.3 试卷题目信息表

序号	字段名称	字段类型	字段长度	允许空	主键	字段说明
1	id	varchar	255	Not null	Y	主键
2	paper_id	varchar	255		N	所属试卷
3	question_id	varchar	255		N	关联题库
4	score	varchar	255		N	题目分数

(4) 考试记录表：记录每个学生在某次考试的得分信息，表的设计如表

5.4。

表 5.4 考试记录表

序号	字段名称	字段类型	字段长度	允许空	主键	字段说明
1	id	varchar	255	Not null	Y	主键
2	user_id	varchar	255		N	考试学生
3	type	varchar	255		N	考试类型
4	subject_id	varchar	255		N	所属课程
5	state	varchar	255		N	考试状态
6	finish_num	int	11		N	完成题数
7	paper_id	varchar	255		N	考试试卷
8	score	double	0		N	试卷得分
9	stu_answer	longblob	0		N	学生答案
10	sys_answer	longblob	0		N	正确答案
11	total_num	int	11		N	题目总数

(5) 试卷题难易分配表：一份试卷中某题型的难易题分配，表的设计如表 5.5。

表 5.5 试卷题难易分配表

序号	字段名称	字段类型	字段长度	允许空	主键	字段说明
1	id	varchar	255	Not null	Y	主键
2	easily	varchar	255		N	容易数量
3	moderate	varchar	255		N	适中数量
4	difficulty	varchar	255		N	困难数量
5	score	int	11		N	每题分数
6	paper_id	varchar	255		N	所属试卷
7	type	varchar	255		N	题目类型
8	knowledge	varchar	255		N	知识点

(6) 课程信息表：用来存放课程的信息，表的设计如表 5.6。

表 5.6 课程信息表

序号	字段名称	字段类型	字段长度	允许空	主键	字段说明
1	id	varchar	255	Not null	Y	主键
2	name	varchar	255		N	课程名称
3	description	varchar	255		N	课程描述
4	type	varchar	255		N	课程类型
5	update_time	datetime	0		N	更新时间

(7) 考试安排表：存储考试安排信息，表的设计如表 5.7。

表 5.7 考试安排表

序号	字段名称	字段类型	字段长度	允许空	主键	字段说明
1	id	varchar	255	Not null	Y	主键
2	name	varchar	255		N	考试名称
3	paper_id	varchar	255		N	所用试卷
4	start_time	varchar	255		N	开始时间
5	end_time	datetime	0		N	结束时间
6	note	varchar	255		N	考试须知
7	submit_time	int	11		N	早提交时间
8	displayInfo	tinyint	1		N	是否屏蔽信息
9	exam_state	tinyint	1		N	是否结束
10	atten_num	int	11		N	实考试人数
11	sub_num	int	11		N	交卷人数
12	nosub_num	int	11		N	未交卷人数
13	pass_score	int	11		N	及格分数

(8) 错题记录表：记录学生在考试试卷中出现错误的题目，表的设计如表 5.8。

表 5.8 错题记录表

序号	字段名称	字段类型	字段长度	允许空	主键	字段说明
1	id	varchar	255	Not null	Y	主键
2	user_id	varchar	255		N	错题学生
3	question_id	varchar	255		N	错误题目
4	type	varchar	255		N	题目类型
5	report_id	varchar	255		N	考试记录

5.3 系统详细功能的设计与实现

通过对在线考试系统的需求分析和各个用户端的功能设计，对系统整体做了功能实现，系统的整体流程如图 5.16 所示。

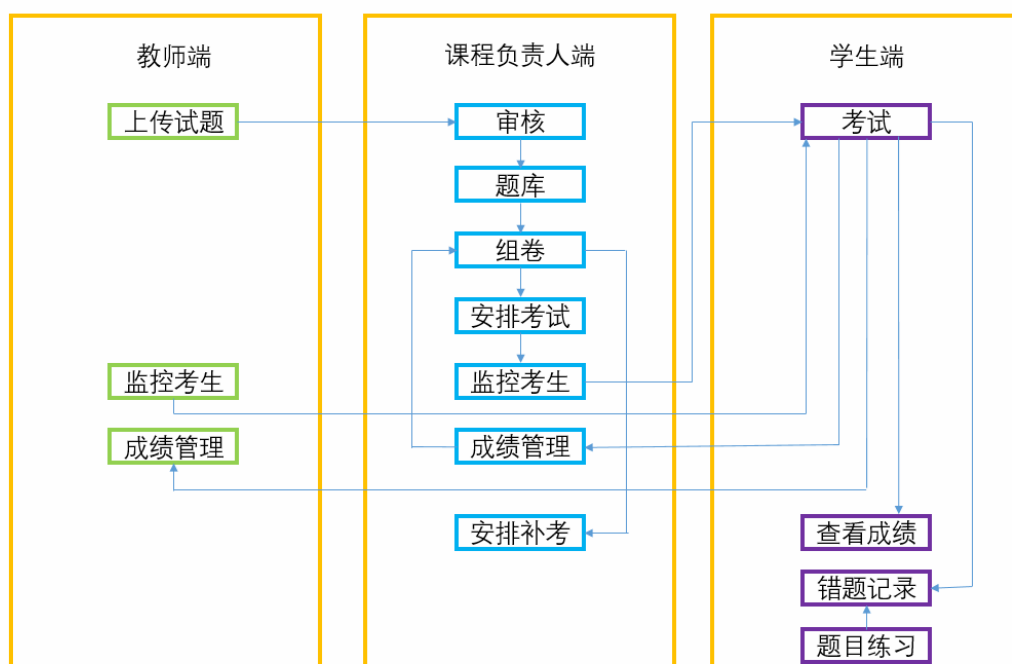


图 5.16 考试系统整体流程图

以下详细展示了在线考试系统各个主要功能界面的设计实现。

首先，输入网址进入在线考试系统登录界面，如图 5.17 所示。

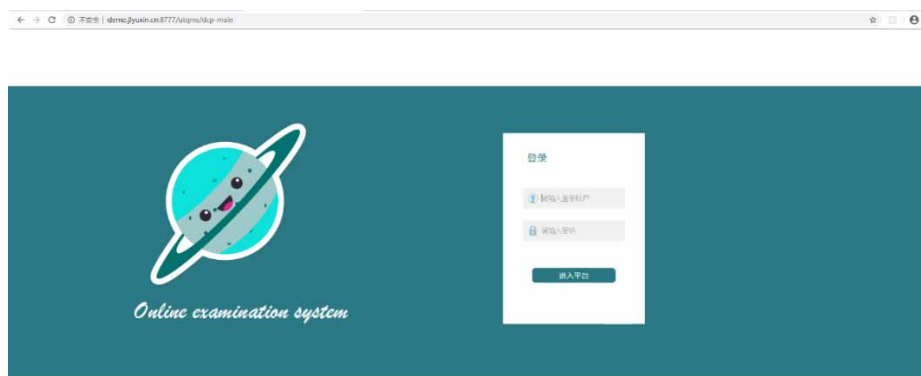


图 5.17 考试系统登录界面

这个界面是统一的登录界面，课程负责人、教师和学生都是通过这个界面登入系统，系统会根据不同的角色权限，显示不同的功能界面。

5.3.1 课程负责人端的功能实现

当课程负责人输入用户名和密码登录系统，页面会跳转到课程负责人的功能界面，课程负责人的功能模块包括用户管理、课程管理、试题审核和考试系统。

1、用户管理

课程负责人可以对教师用户进行管理，添加删除教师信息，如图 5.18 所示。

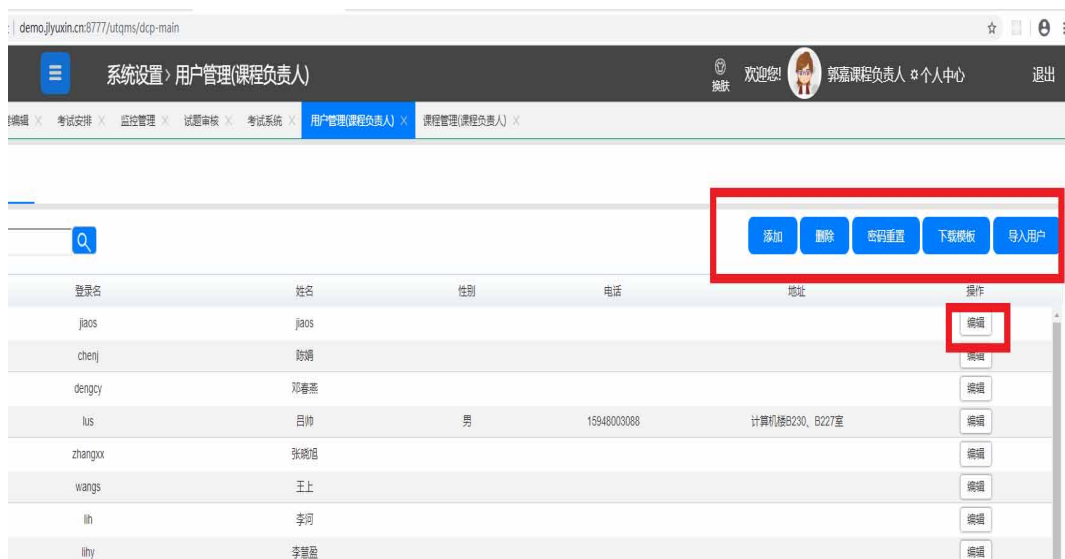


图 5.18 用户管理页面

2、课程管理

课程负责人可以对自己所负责的课程信息进行编辑，主要对课程所包含的知识点进行添加和删除，如图 5.19 所示。

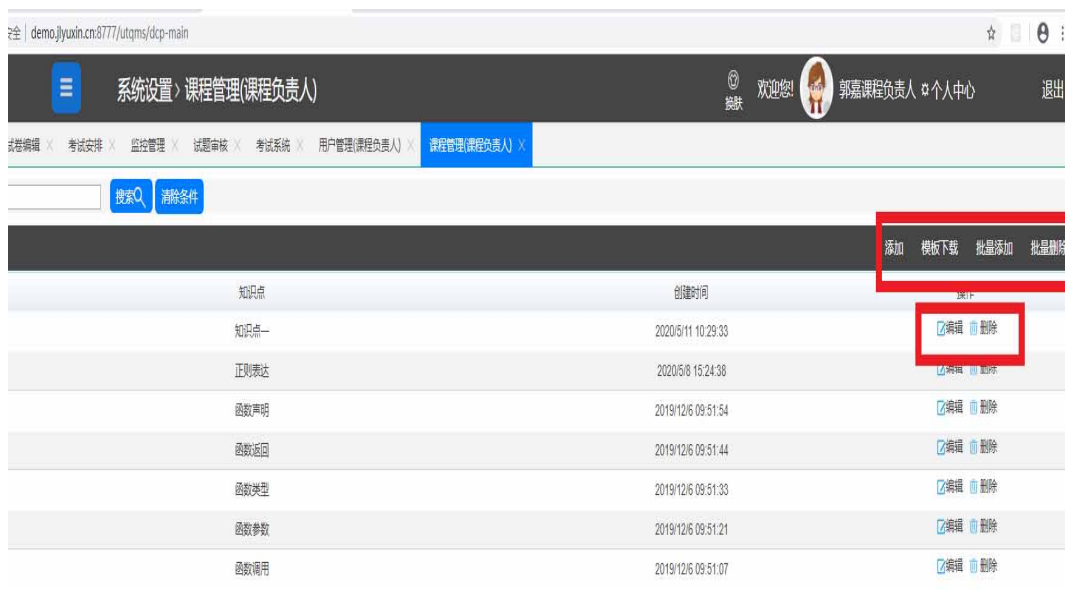


图 5.19 课程管理页面

3、试题审核

课程负责人对上传到题库的题目进行试题审核，所有的题目是由教师上传

的题目列表，如图 5.20 所示。

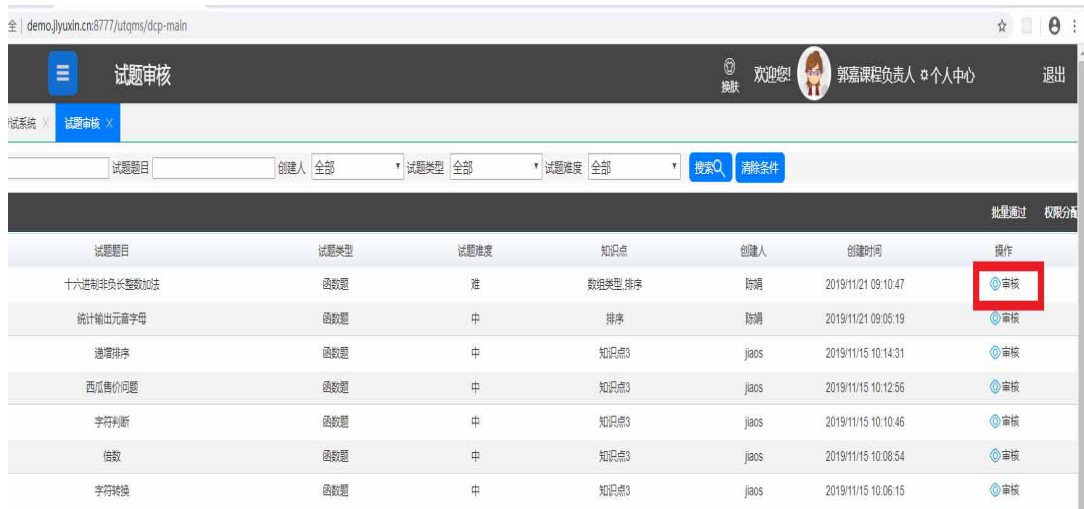


图 5.20 试题审核页面

选择某个题目，点击审核，可以选择审核通过题目，或者不通过，如图 5.21 所示。

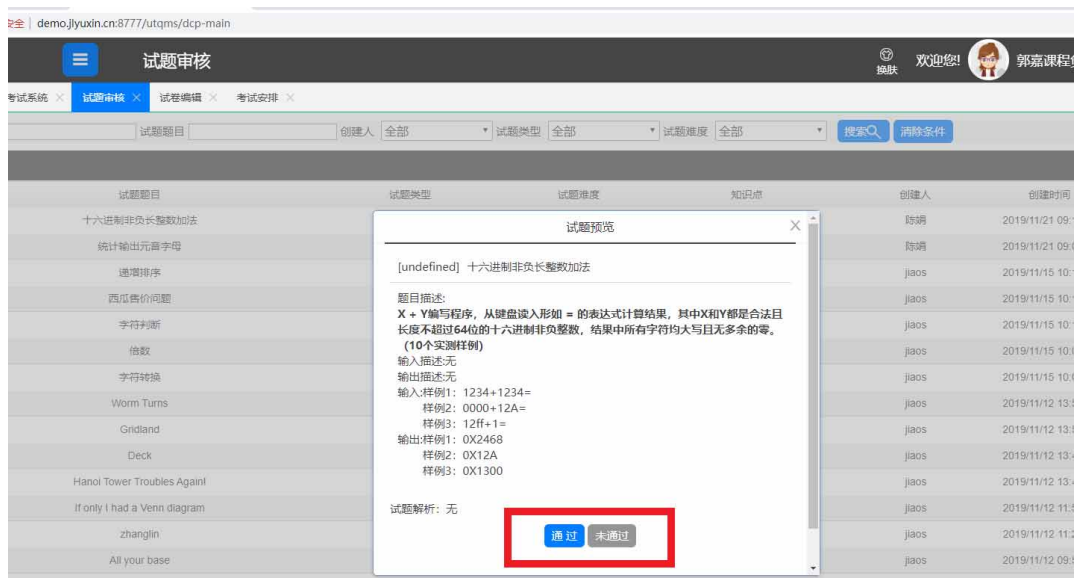


图 5.21 审核预览页面

当课程负责人选择通过，则审核成功，试题录入题库；如果选择不通过，需要课程负责人返回一个理由，如图 5.22 所示。

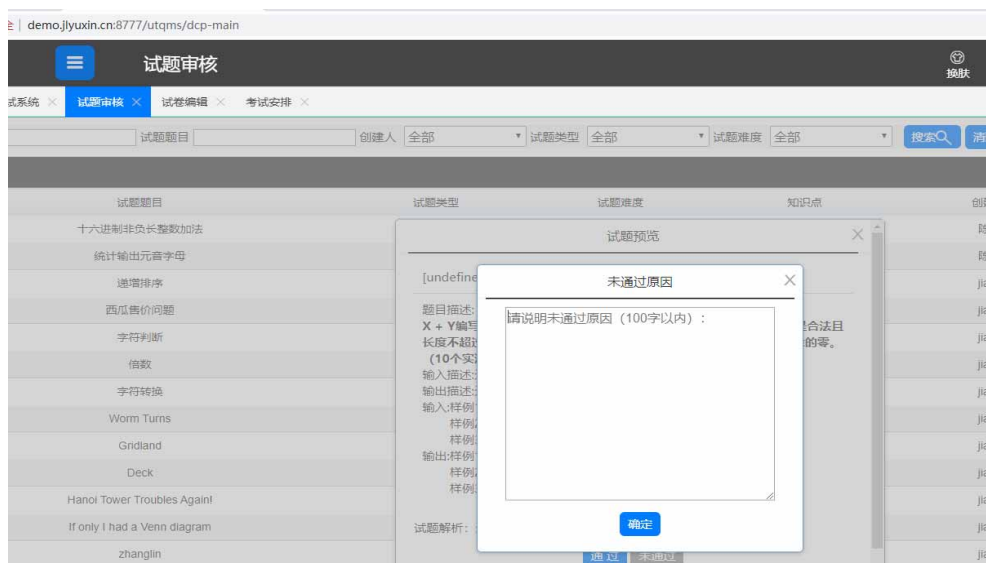


图 5.22 审核未通过页面

课程负责人试题审核流程如图 5.23 所示。

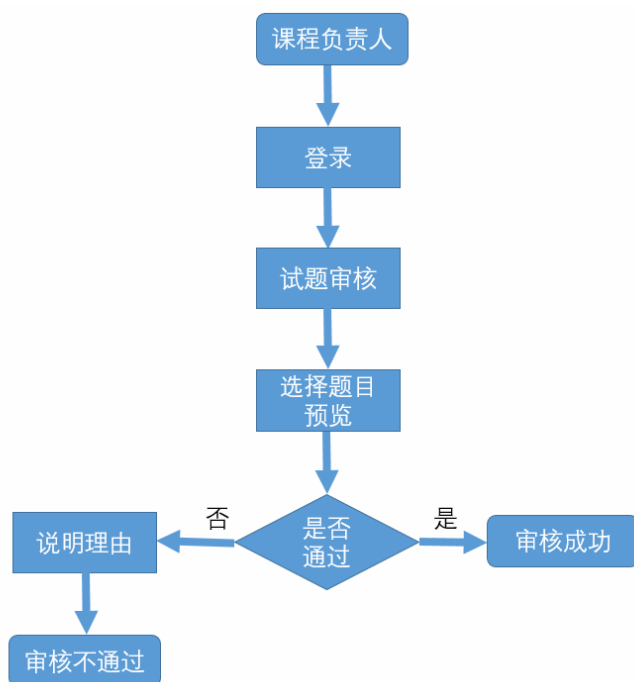


图 5.23 试题审核流程图

4、考试系统

(1) 试卷编辑

课程负责人负责考试组卷，包括正式考试卷和模拟练习试卷。组卷有两种方式，一种是自动组卷，只需设定好抽题规则，学生进入考试，系统会根据规则，随机从题库抽取题目。另一种是手动组卷，需要课程负责人手动从题库选

取题目，组成试卷。两种组卷方式的页面如图 5.24 和 5.25 所示。

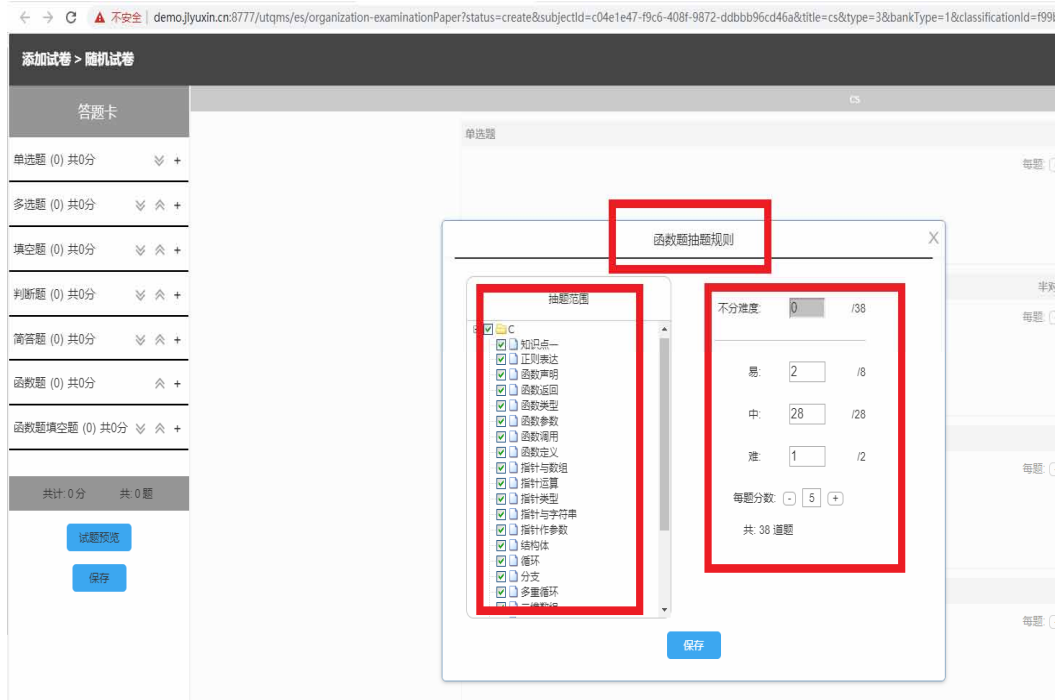


图 5.24 自动组卷页面



图 5.25 手动组卷页面

课程负责人组卷流程如图 5.26 所示。

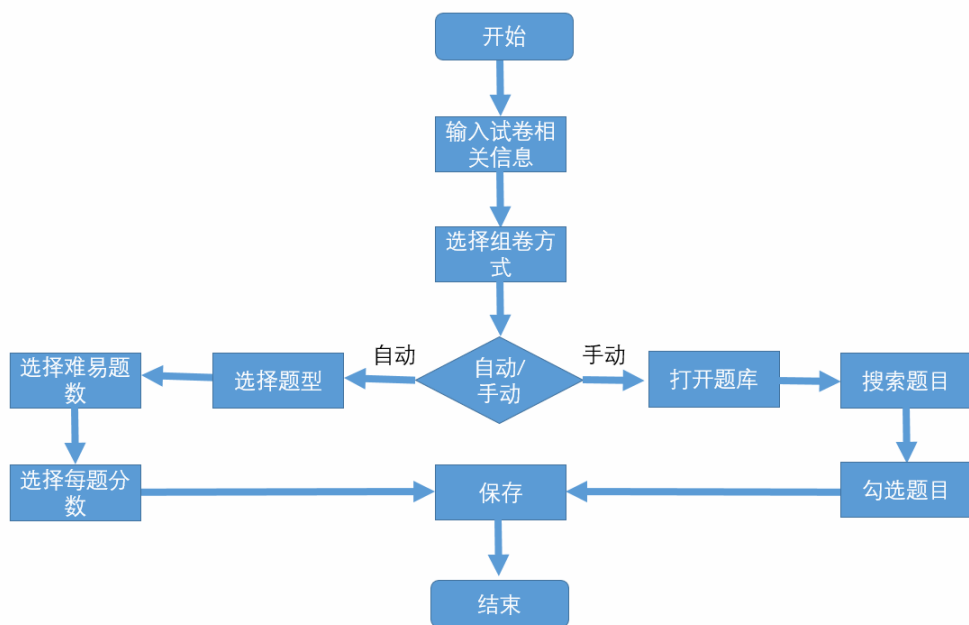


图 5.26 组卷流程图

(2) 考试安排

课程负责人完成组卷后，可以对本次考试进行安排，包括考试时间、考试人员信息等，如图 5.27 所示。

基础信息	
*选择试卷:	<input type="text"/>
*练习名称:	<input type="text"/>
*通过分数:	<input type="text"/>

场次设置	
*练习开始日期:	<input type="text"/>
*练习开始时间:	[请选择]
*练习时长:	[请选择]

分配任务	
<input type="button" value="选择参加人员"/>	
<div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div>	

图 5.27 考试安排页面

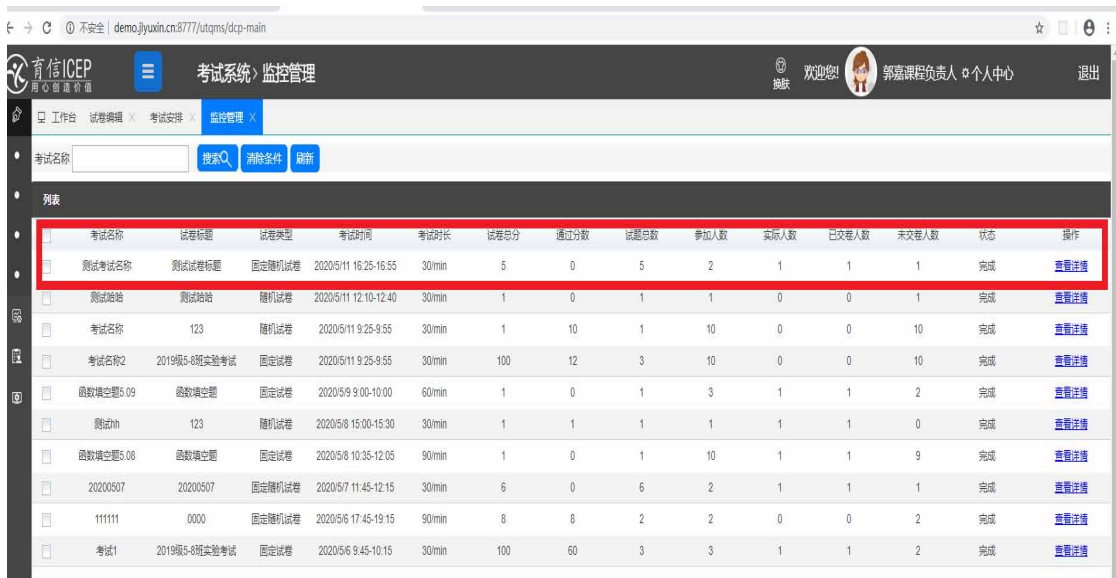
(3) 练习安排

课程负责人完成模拟考试练习卷组卷之后，可对本次练习进行安排，包括考试时间、考试人员信息等，与考试安排界面一致。

(4) 监控管理

课程负责人可以查看每场考试的信息，包括考试名称、试卷标题、考试时

间、考试状态、实考人数等，如图 5.28 所示。

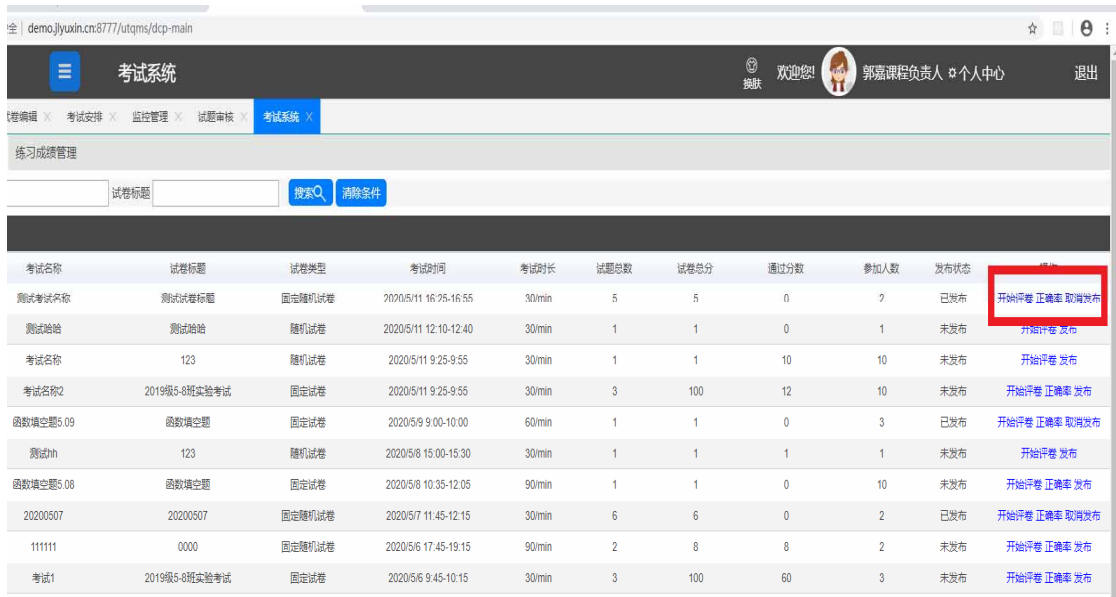


考试名称	试卷标题	试卷类型	考试时间	考试时长	试卷总分	通过分数	试题总数	参加人数	实考人数	已交卷人数	未交卷人数	状态	操作
测试考试名称	测试试卷标题	固定随机试卷	2020/5/11 16:25-16:55	30min	5	0	5	2	1	1	1	完成	查看详情
测试试卷	测试试卷	随机试卷	2020/5/11 12:10-12:40	30min	1	0	1	1	0	0	1	完成	查看详情
考试名称	123	随机试卷	2020/5/11 9:25-9:55	30min	1	10	1	10	0	0	10	完成	查看详情
考试名称2	2019级5-8班实验考试	固定试卷	2020/5/11 9:25-9:55	30min	100	12	3	10	0	0	10	完成	查看详情
函数填空题5.09	函数填空题	固定试卷	2020/5/9 9:00-10:00	60min	1	0	1	3	1	1	2	完成	查看详情
测试h	123	随机试卷	2020/5/8 15:00-15:30	30min	1	1	1	1	1	1	0	完成	查看详情
函数填空题5.08	函数填空题	固定试卷	2020/5/8 10:35-12:05	90min	1	0	1	10	1	1	9	完成	查看详情
20200507	20200507	固定随机试卷	2020/5/7 11:45-12:15	30min	6	0	6	2	1	1	1	完成	查看详情
111111	0000	固定随机试卷	2020/5/6 17:45-19:15	90min	8	8	2	2	0	0	2	完成	查看详情
考试1	2019级5-8班实验考试	固定试卷	2020/5/6 9:45-10:15	30min	100	60	3	3	1	1	2	完成	查看详情

图 5.28 监控管理页面

(5) 成绩管理

学生提交试卷，系统可自动阅卷，给出得分。本文聚焦于练习试题推荐算法研究，在自动阅卷的实现逻辑上比较简单，客观题采用直接答案比对的方式，而主观题采用关键字比对的方式。所以针对系统对部分主观题判断不合理的情况，课程负责人可查阅试卷，修改分数，给予更合理的得分。评卷完成即可发布试卷成绩，学生端便可查看。如图 5.29 所示。



考试名称	试卷标题	试卷类型	考试时间	考试时长	试题总数	试卷总分	通过分数	参加人数	发布状态	操作
测试考试名称	测试试卷标题	固定随机试卷	2020/5/11 16:25-16:55	30min	5	5	0	2	已发布	开始评卷 正确率 取消发布
测试试卷	测试试卷	随机试卷	2020/5/11 12:10-12:40	30min	1	1	0	1	未发布	开始评卷 发布
考试名称	123	随机试卷	2020/5/11 9:25-9:55	30min	1	1	10	10	未发布	开始评卷 发布
考试名称2	2019级5-8班实验考试	固定试卷	2020/5/11 9:25-9:55	30min	3	100	12	10	未发布	开始评卷 正确率 发布
函数填空题5.09	函数填空题	固定试卷	2020/5/9 9:00-10:00	60min	1	1	0	3	已发布	开始评卷 正确率 取消发布
测试h	123	随机试卷	2020/5/8 15:00-15:30	30min	1	1	1	1	未发布	开始评卷 发布
函数填空题5.08	函数填空题	固定试卷	2020/5/8 10:35-12:05	90min	1	1	0	10	未发布	开始评卷 正确率 发布
20200507	20200507	固定随机试卷	2020/5/7 11:45-12:15	30min	6	6	0	2	已发布	开始评卷 正确率 取消发布
111111	0000	固定随机试卷	2020/5/6 17:45-19:15	90min	2	8	8	2	未发布	开始评卷 正确率 发布
考试1	2019级5-8班实验考试	固定试卷	2020/5/6 9:45-10:15	30min	3	100	60	3	未发布	开始评卷 正确率 发布

图 5.29 成绩管理页面

5.3.2 教师端的功能实现

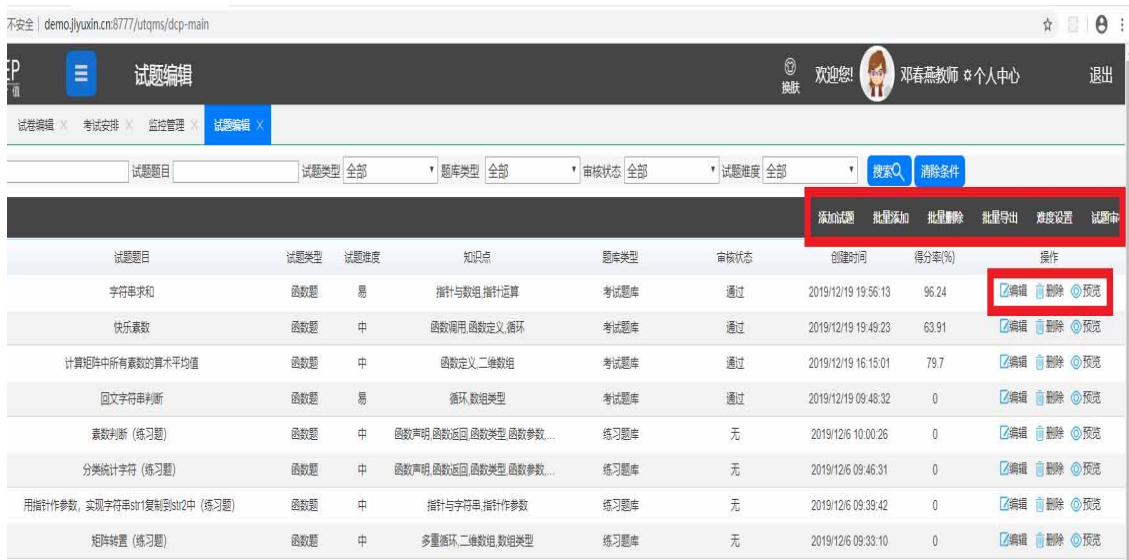
当教师输入用户名和密码登录系统，页面会跳转到教师的功能界面，教师的功能模块包括用户管理、试题上传和考试系统。

1、用户管理

教师用户管理功能与课程负责人管理功能类似，教师主要负责学生用户信息的添加和删除。

2、试题上传

教师在负责题目的编辑和上传，编辑完成之后需等待课程负责人审核。教师可查看试题审核状态，如图 5.30 所示。



试题题目	试题类型	试题难度	知识点	题库类型	审核状态	创建时间	得分率(%)	操作
字符串求和	函数题	易	指针与数组, 指针运算	考试题库	通过	2019/12/19 19:56:13	96.24	编辑 删除 预览
快乐素数	函数题	中	函数调用, 函数定义, 循环	考试题库	通过	2019/12/19 19:49:23	63.91	编辑 删除 预览
计算矩阵中所有素数的算术平均值	函数题	中	函数定义, 二维数组	考试题库	通过	2019/12/19 16:15:01	79.7	编辑 删除 预览
回文字符串判断	函数题	易	循环, 数组类型	考试题库	通过	2019/12/19 09:48:32	0	编辑 删除 预览
素数判断 (练习题)	函数题	中	函数声明, 函数返回, 函数类型, 函数参数...	练习题库	无	2019/12/6 10:00:26	0	编辑 删除 预览
分类统计字符 (练习题)	函数题	中	函数声明, 函数返回, 函数类型, 函数参数...	练习题库	无	2019/12/6 09:46:31	0	编辑 删除 预览
用指针作参数, 实现字符串str1复制到str2中 (练习题)	函数题	中	指针与字符串, 指针作参数	练习题库	无	2019/12/6 09:39:42	0	编辑 删除 预览
矩阵转置 (练习题)	函数题	中	多重循环, 二维数组, 数组类型	练习题库	无	2019/12/6 09:33:10	0	编辑 删除 预览

图 5.30 试题编辑页面

3、考试系统

(1) 监控管理

教师点击监控管理，可以查看每场考试的信息，包括考试名称、试卷标题、考试时间、考试状态、实考人数等，与课程负责人功能一致。

(2) 成绩管理

教师可查看每场考试学生试卷，并根据实际情况，对部分系统判断不合理的主观题进行得分修正，与课程负责人功能一致。

5.3.3 学生端的功能实现

当学生输入用户名和密码登录系统，页面会跳转到学生的功能界面，学生

的功能模块包括我的考试和我的练习。

1、我的考试

学生在点击我的考试，可以看到已完成的考卷和即将要进行的考卷信息。对于已考完的试卷，需等待课程负责人或教师发布成绩。发布完成，可以查看考卷信息和成绩；若未发布，则显示未发布。对于即将要进行的考卷显示未开始，禁止进行操作；进入考试时间段，可进行开始答卷的操作，然后进入答题界面。如图 5.31 所示。



图 5.31 我的考试页面

学生参加考试和查看成绩的流程如图 5.32 所示。

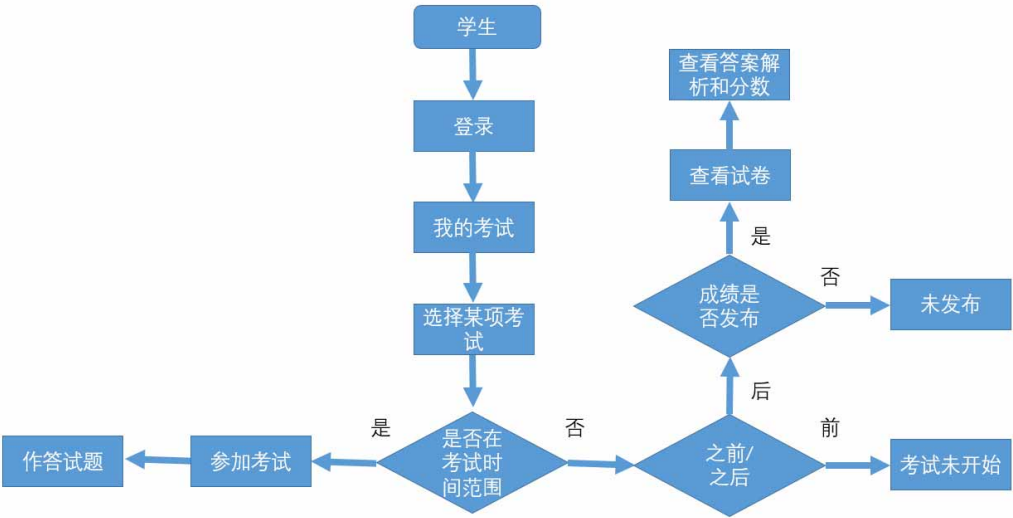


图 5.32 考试和查看成绩流程图

2、我的练习

(1) 参加练习

学生点击参加练习，可以看到所有已参加和未参加的模拟考试试卷信息，与我的考试页面一致，显示所有练习试卷。

(2) 知识点练习

学生进入知识点练习页面下，可自行进行试题练习。包括两种方式，一种是自主组卷，学生可自主选择想要练习的课程和知识点，然后选择各种题型数量，提交即可随机从题库抽取题目组成答题试卷；另外一种推荐练习，系统会调用推荐算法，根据学生历史的错题记录数据，自动为学生推荐一组试题，组成答题卷。知识点练习界面如图 5.33 所示。



图 5.33 知识点练习页面

知识点练习之自主组卷的页面如图 5.34 所示。

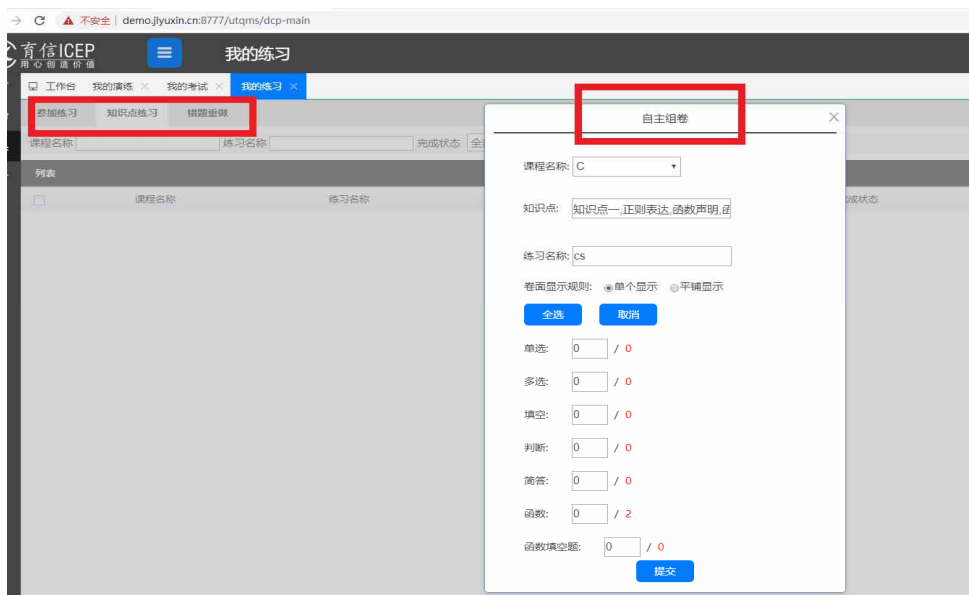


图 5.34 自主组卷练习页面

知识点练习之推荐练习的页面如图 5.35 所示。

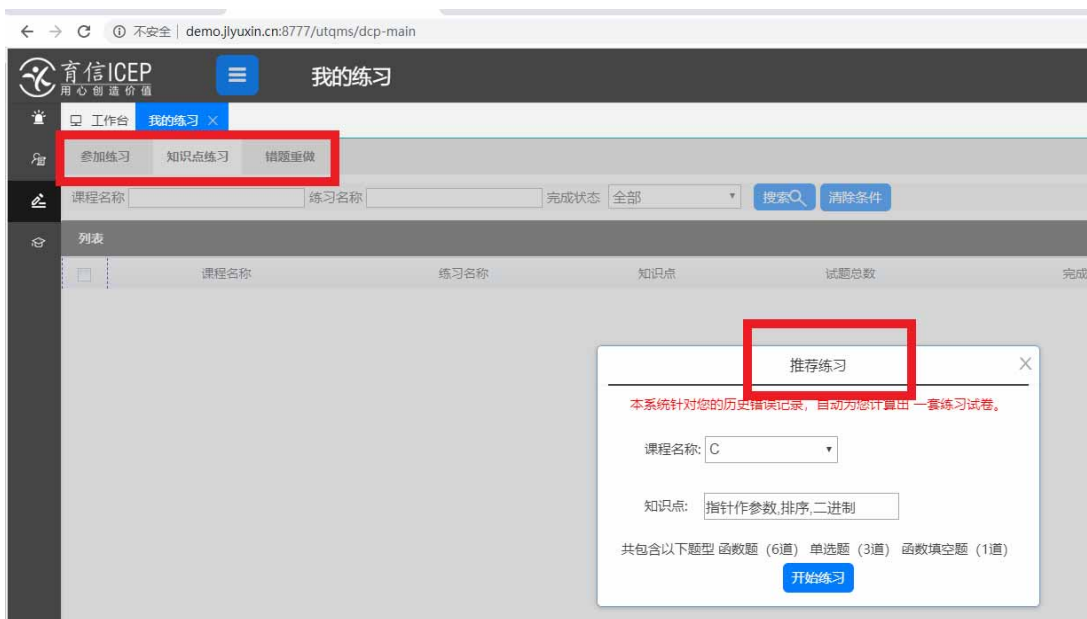


图 5.35 推荐练习页面

点击开始练习，会展示推荐题目的答题卡界面，进行答题。如图 5.36 所示。



图 5.36 答题卡页面

(3) 错题重做

学生在考试和题目练习过程中做错的题，都会记录到错题库中，学生点击错题重做，可针对题型进行错题练习，如图 5.37 所示。

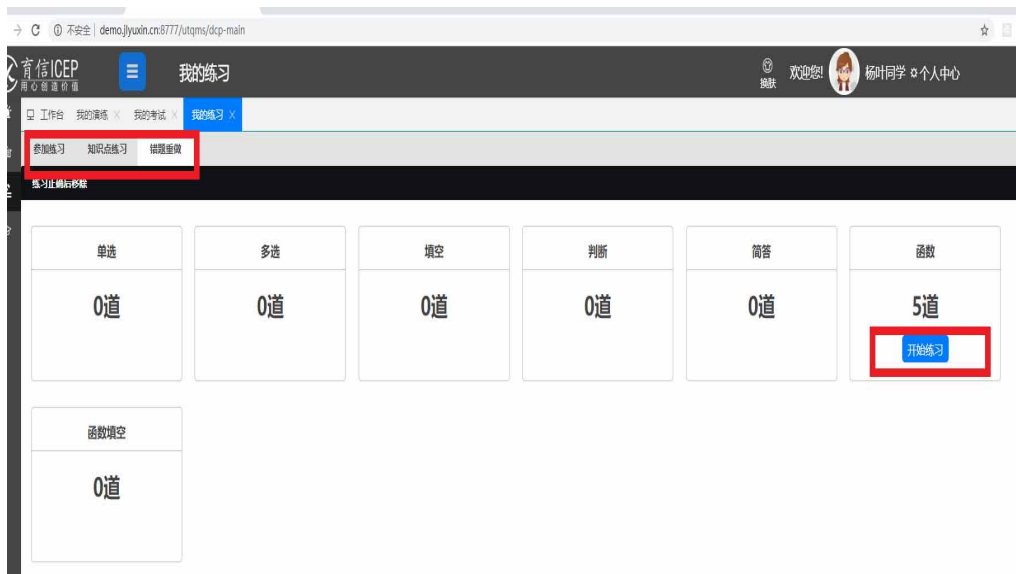


图 5.37 错题重做页面

5.4 系统测试

5.4.1 测试目的及概述

系统测试是整个系统设计开发过程中一个重要的步骤，是对系统有效性和可靠性的重要保障。通过在实际环境中对系统功能进行测试，可以发现系统中存在的问题，包括系统能否正常运行，系统功能是否满足预设需求，用例是否可操作，数据库是否能正常连接，数据是否可正确获取等。及时改正这些问题，从而保证系统质量，使其符合设计规范。

5.4.2 系统功能模块测试

为了保证系统功能的正常运行，需重点关注系统各个功能模块的正确实现，并采用黑盒测试法进行测试。由于系统包含了三个权限的用户端，因此针对各个用户端进行功能模块测试。

1、课程负责人端功能测试

课程负责人端主要包含用户管理、课程管理、试题审核、试卷编辑、考试安排、监控管理和成绩管理等功能模块。分别对课程负责人功能模块进行测试，测试表如表 5.9。

表 5.9 课程负责人功能模块测试表

测试编号	功能模块	测试方式	期望结果	测试结果
1	登录	课程负责人输入用户名和密码，点击进入平台	成功登入课程负责人端界面，正确显示课程负责人权限功能	Pass
2	用户管理	课程负责人点击添加、删除、编辑等按钮	能成功添加和删除教师用户，并成功编辑教师用户信息	Pass
3	课程管理	课程负责人点击添加、删除、编辑等按钮	能成功添加和删除课程包含知识点，并成功修改知识点信息	Pass
4	试题审核	课程负责人选择题目点击审核按钮，然后选择通过或者不通过	点击审核弹出试题预览页面，然后选择通过则提示审核成功；选择未通过弹出说明原因提示框	Pass
5	试卷编辑	课程负责人点击添加试卷，并选择随机试卷或者固定试卷	选择随机试卷则成功进入随机组卷页面；选择固定试卷则成功进入固定试卷页面	Pass

续表

测试编号	功能模块	测试方式	期望结果	测试结果
6	考试安排	课程负责人点击安排考试	成功进入考试安排信息界面	Pass
7	监控管理	课程负责人选择某次考试，点击查看详情	成功查看参加本次考试所有学生考试情况	Pass
8	成绩管理	课程负责人点击开始评卷、发布等按钮	能正确进行学生试卷查看和评阅，并且成功完成考试成绩发布	Pass

2、教师端功能测试

教师端主要包含用户管理、试题上传、监控管理和成绩管理等功能模块。分别对教师功能模块进行测试，测试表如表 5.10。

表 5.10 教师功能模块测试表

测试编号	功能模块	测试方式	期望结果	测试结果
1	登录	教师输入用户名和密码，点击进入平台	成功登入教师端界面，正确显示教师权限功能	Pass
2	用户管理	教师点击添加、删除、编辑等按钮	能成功完成添加、删除和编辑学生用户信息	Pass
3	试题上传	教师点击添加试题、删除、预览等按钮	能完成试题的添加、删除和预览	Pass

续表

测试编号	功能模块	测试方式	期望结果	测试结果
4	监控管理	教师选择某次考试，点击查看详情	成功查看参加本次考试所有学生考试情况	Pass
5	成绩管理	教师点击开始评卷、发布等按钮	能正确进行学生试卷查看和评阅，并且成功完成考试成绩发布	Pass

3、学生端功能测试

学生端主要包含我的考试、参加练习、知识点自主组卷练习、知识点题目推荐练习和错题重做等功能模块。分别对学生功能模块进行测试，测试表如表 5.11。

表 5.11 学生功能模块测试表

测试编号	功能模块	测试方式	期望结果	测试结果
1	登录	学生输入用户名和密码，点击进入平台	成功登入学生端界面，正确显示学生权限功能	Pass
2	我的考试	学生点击参加考试	成功进入考试试卷答题卡界面	Pass
3	参加练习	学生点击参加练习	成功进入练习试卷答题卡界面	Pass

4	自主组卷练习	学生点击自主组卷，设置抽取题目信息	成功完成知识点题目随机抽取并组成试卷	Pass
---	--------	-------------------	--------------------	------

续表

测试编号	功能模块	测试方式	期望结果	测试结果
5	推荐练习	学生点击推荐练习，然后点击开始练习	进入系统推荐的题目列表答题卡界面	Pass
6	错题重做	学生选择题型，点击开始练习	弹出信息框，学生可选择课程和题目数量	Pass

5.5 推荐算法应用

5.5.1 学生知识点练习推荐练习功能分析

基于以上小节对在线考试系统的分析和功能模块的实现，可以看出，在线考试系统可完成自动组卷、阅卷和在线考试等基本功能，减轻了课程负责人和教师的负担，满足了学生参加考试的基本需求。同时，具备学生进行题目的练习功能，并且提供了多种练习方式，包括模式试卷练习、知识点练习和错题重做。此外，考虑了学生个性化的学习需求，在知识点练习功能下设计了两种方式，学生可根据实际情况，选择自主随机抽题练习，也可以练习系统推荐的题目。

对于系统推荐题目的方式，需要学生历史做题数据作为支撑，并设计一种高效的推荐算法，这样对学生的学习更具有针对性。一个好的推荐算法往往可以很好地洞察到学生真正需要学习哪些知识点，哪些地方掌握有所不足，从而更具针对性地为不同学生推荐一些适合他们的题目，尤其一些易错题和掌握不牢靠的题目，最终达到学生个性化的学生学习需求。高效的学习方式往往会事半功倍。

5.5.2 推荐算法应用可行性

1、业务功能可行性

对于学生试题练习来说,纯粹的自主练习采用随机从题库抽题的方式比较盲目。首先,学生可能本身不能很好地意识到和总结出自己真正在哪些知识点掌握不足,所以在自主练习知识点选取的时候可能不太准确,导致遗漏的知识点题目不能练习全面,或者浪费很多时间练习一些不重要的知识点题目。其次,自主抽取的题目可能在难度和掌握程度等多方面不太适合自己练习,也会导致一些不必要的时间消耗。为此,本文在系统设计的时候,就在知识点题目练习模块中设计了两种练习方式,方便学生根据实际情况选择,可自主组卷,也可按照系统推荐题目练习。

推荐系统作为一个个性化的方式,在其他领域的应用已经相当普遍和成熟。同样在教育领域,个性化也是一个重要的发展方向。可以借鉴推荐系统在其他领域实现的个性化效果,将其应用到考试系统试题练习模块中。通过分析学生历史的做题数据,发现学生的薄弱点和易错题目类型,从而更加针对性地为每一个学生推荐适合他们的题目。使得学生在平时练习题目的时候,目标更加明确,学习更加高效,实现个性化发展。

2、数据可行性

推荐算法的实施需要相关的数据作为支撑,对于在线考试系统中题目练习的推荐,需要收集学生的做题记录数据。而根据在线考试系统的业务流程,学生在考试和自主练习过程中都会产生错题,并且会将错题记录到错题集中。对应的数据库设计中,也相应对错题记录表进行了设计,错题表存储的是学生所有的错题,包括学生 id、题目 id、题目类型等字段。

通常推荐算法研究中,主要需要收集用户记录数据,包含 `userId`、`itemId` 和 `rating`, `rating` 字段表示 user 对 item 评分。电商领域中, `rating` 表示买家对商品的打分,数值反映了喜好程度;而在本文试题推荐场景中,学生 id 对应 `userId`, 题目 id 对应 `itemId`, `rating` 可以表示学生对于题目的错误程度,数值反映了学生对该题目所包含知识点掌握不足的程度。推荐算法的研究数据又分为显示评分数据和隐式评分数据。显示评分数据一般都不容易搜集,例如在电商领域中,买家购买商品,电商网站会有一个商品星级评分的功能,大部分买家都不会去进

行评分，数据收集起来非常困难，导致收集到的用户-商品评分矩阵极为稀疏，不利于推荐算法进行，影响准确率。因此，常常选取隐式评分数据来进行研究，代价小很多，相对更容易收集。隐式评分数据不需要收集具体 rating，只需关注是否发生过交互，可用 0 或 1 表示。试题推荐中，对于 rating，即题目错误程度，很难有一个计算标准，或者说不太适用显式数据。所以，可以基于隐式数据的形式，只需关注对于每一道题目，学生是否做错，用 0 或 1 表示。

经过以上分析，系统错题记录数据库表中的数据记录了每个学生所有的错题，所以，可以很容易的收集到推荐算法研究所需数据。试题推荐算法研究中，需要收集学生 id、题目 id、是否做错，这样格式的做题记录数据，系统的错题记录数据库表可以满足算法需求。推荐算法将学生 id、题目 id 作为模型的输入，是否做错作为目标输出进行算法模型训练，通过分析学生的错题记录数据，可以学习到学生做题的特点，从而发现学生对哪些题目掌握不足，每道题目会包含几个知识点，也就是进而可总结出学生对哪些知识点还不太熟悉。对于某个学生，将该学生和一道没有做过的题目输入模型，模型可预测该学生是否能正确完成该题目。如果预测为错误，可将该题目作为易错题推荐给该学生。

由于本文实现了系统原型，仍在测试阶段，不能收集到大量真实的学生做题记录数据，为了保证算法的准确性，需要大量的数据训练作为支撑。因此，本文在算法研究和实验中，选取了公开的广泛用于推荐算法研究的 Gowalla 数据集和 Amazon-book 数据集来进行验证。系统已预留了从数据库学生做题记录数据表中获取数据的接口，和推荐功能实现的接口。事实上，本文对推荐算法的研究主要关注的是推荐模型本身的性能和能否完成考试系统试题推荐功能，不针对某个具体领域和场景，因此在数据集的选取中，不关注数据集本身的内容和领域属性。算法基于隐式数据，在对数据进行处理之后，任何数据集都具有相同格式，只保留了 id 属性，每一行第一个 id 表示 user，对应学生记录数据中的学生 id；每一行除去第一个，后面所有的 id 都表示用户有过历史行为的 item，对应学生记录数据中该学生的错题集合。所以，在不能收集到足够量的真实学生做题记录数据的情况下，可以将两个公开数据集进行处理之后作为做题记录数据使用，来进行推荐算法的研究。

5.5.3 冷启动问题的解决

对于推荐算法的冷启动问题，即新注册在线考试系统的学生，在没有历史错题信息数据的情况下，无法通过推荐算法模型为其推荐试题。为此本文采取了热门推荐的策略，即通过统计其他学生的历史错题情况，挑出错误次数最多的排名前面的题目推荐给新学生。事实上对于一道题目，如果做错的人数和次数很多，说明该题目有一定的难度，一者可能是题目隐藏了很多的机关技巧，也有可能是教师在教学过程中没有很好地将该题目所涉及知识点讲透彻。这样的题目值得考究，推荐给学生也很有意义。

5.6 本章小结

本章主要介绍了在线考试系统的整体设计和主要功能模块实现。首先对系统的开发环境做了简单说明，并对系统做了需求分析和可行性分析。然后介绍了系统整体的架构设计。接着详细展示了三种权限用户的关键功能模块的实现并进行了相关测试。最后对学生知识点练习的推荐功能模块进行了分析，尝试将推荐算法应用进来，可实现学生个性化的学习。

第6章 总结与展望

6.1 工作总结

本文通过在线考试系统的开发，实现了无纸化、自动化，节省了传统人工出卷而产生的大量人力消耗和时间开销；同时节约了印刷纸质试卷所需成本。为实现学生个性化的学习需求并且满足学校和平台要求，本文将推荐算法引入所开发的考试系统中，在系统知识点练习功能模块下设计了两种练习方式，分别为自主组卷和推荐练习。前者基于随机抽题的方式，后者采用了推荐算法，完成了个性化推荐。

本文主要做了以下工作：

- 1、针对考试系统试题推荐练习的功能，进行了推荐算法研究。在神经图协同过滤算法（NGCF）的基础上进行了改进，提出一种基于深度学习和注意力机制（Attention）的推荐算法模型 NGCF-Att。
- 2、在可映射为本系统数据库格式的 Amazon-book 和 Gowalla 等公开数据集上进行实验来验证推荐算法的效果。
- 3、开发了在线考试系统，实现了基本的自动组卷、阅卷以及题目练习等功能，并在题目练习功能模块下设计了试题推荐的练习方式。

6.2 未来研究展望

目前在线考试系统已初步开发完成，可以实现学生在线考试、自动组卷、自动阅卷以及试题练习的功能。但由于时间问题，有一些工作还有待完成：

- 1、由于完成了系统原型开发，仍在测试阶段，不能拿到一个真实的且有足够数据量的用户记录数据集，所以训练模型只能用网上现有的推荐模型数据集进行。后期还需收集到足够多的学生用户做题记录数据，然后基于这些数据进行实验，进一步验证算法效果。
- 2、随着用户量的增加，系统的运算量会大大增加，后期可以将 Hadoop 等分布式大数据处理技术应用到系统中，增加系统的运算能力和快速响应能力。
- 3、随着系统功能的不断增加和完善，可以考虑将推荐算法应用到更多的功

能模块上，使得系统更加个性化。

4、在推荐算法的研究上，主要将神经协同过滤框架的 MLP 模型和神经图协同过滤算法进行了结合，也可以尝试将 GMF 模型也结合进来，进一步提升性能。此外还可以探索关于用户/物品的 Embedding 和图网络结构的对抗学习网络来增强鲁棒性。

参考文献

- [1] Pennsylvania U O. Journal of the history of ideas. [M]. Journal of the History of Ideas, Inc. 1940.
- [2] 王国霞, 刘贺平. 个性化推荐系统综述[J]. 计算机工程与应用, 2012, 48(7):66-76.
- [3] 印鉴, 陈忆群, 张钢. 搜索引擎技术研究与发展%Research and Development of Search Engine Technology[J]. 计算机工程, 031(14):54-56,104.
- [4] Resnick P, Varian H R. Recommender systems. Commun ACM[J]. Communications of the ACM, 1997, 40(3):56-58.
- [5] Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model[C]//Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008. ACM, 2008.
- [6] 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.
- [7] Fan C, Xiao F, Zhao Y. A short-term building cooling load prediction method using deep learning algorithms[J]. Applied Energy, 2017, 195:222-233.
- [8] Pharo N, Jaervelin K. The SST method: a tool for analysing Web information search processes[J]. Information Processing & Management, 2004, 40(4): p. 633-654.
- [9] Goldberg D, Nichols D A, Oki B M, et al. Using collaborative filtering to weave an information TAPESTRY[J]. Communications of the ACM, 1992, 35(12):61-70.
- [10] Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of netnews[C]//Proceedings of the 1994 ACM conference on Computer supported cooperative work. 1994: 175-186.
- [11] Davidson J, Liebald B, Liu J, et al. The YouTube video recommendation system[C]//Proceedings of the fourth ACM conference on Recommender systems. 2010: 293-296.

- [12]Linden G, Smith B, York J. Amazon.com recommendations: Item-to-item collaborative filtering[J]. IEEE Internet Computing, 2003, 7(1):76-80.
- [13]Balabanović M, Shoham Y. Fab: content-based, collaborative recommendation[J]. Communications of the ACM, 1997, 40(3): 66-72.
- [14]Badsha S, Yi X, Khalil I, et al. Privacy preserving user-based recommender system[C]//2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2017: 1074-1083.
- [15]Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]. International Conference on World Wide Web. ACM, 2001: 285-295.
- [16]Deshpande M, Karypis G. Item-based top-n recommendation algorithms[J]. ACM Transactions on Information Systems (TOIS), 2004, 22(1): 143-177.
- [17]Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [18]Tran T, Cohen R. Hybrid recommender systems for electronic commerce[C]//Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press. 2000, 40.
- [19]Li Q, Kim B M. Clustering approach for hybrid recommender system[C]//Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003). IEEE, 2003: 33-38.
- [20]Koren Y. The bellkor solution to the netflix grand prize[J]. Netflix prize documentation, 2009, 81(2009): 1-10.
- [21]Töscher A, Jahrer M, Bell R M. The bigchaos solution to the netflix grand prize[J]. Netflix prize documentation, 2009: 1-52.
- [22]Sarwar B, Karypis G, Konstan J, et al. Application of dimensionality reduction in recommender system-a case study[R]. Minnesota Univ Minneapolis Dept of Computer Science, 2000.
- [23]Bell R M, Koren Y. Scalable collaborative filtering with jointly derived neighborhood interpolation weights[C]//Seventh IEEE International Conference on Data Mining (ICDM 2007). IEEE, 2007: 43-52.
- [24]Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering[C]//Proceedings of the 24th international conference on Machine learning. 2007: 791-798.

- [25]Covington P, Adams J, Sargin E. Deep neural networks for youtube recommendations[C]//Proceedings of the 10th ACM conference on recommender systems. 2016: 191-198.
- [26]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. 2018: 515-524.
- [27]Wang H, Wang N, Yeung D Y. Collaborative deep learning for recommender systems[C]//Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. 2015: 1235-1244.
- [28]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.
- [29]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.
- [30]Zheng L, Lu C T, Jiang F, et al. Spectral collaborative filtering[C]//Proceedings of the 12th ACM Conference on Recommender Systems. 2018: 311-319.
- [31]Yang J H, Chen C M, Wang C J, et al. HOP-rec: high-order proximity for implicit recommendation[C]//Proceedings of the 12th ACM Conference on Recommender Systems. 2018: 140-144.
- [32]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.
- [33]Nilashi M, Ibrahim O, Bagherifard K. A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques[J]. Expert Systems with Applications, 2018, 92: 507-520.
- [34]Zuo Y, Gong M, Zeng J, et al. Personalized recommendation based on evolutionary multi-objective optimization [research frontier][J]. IEEE Computational Intelligence Magazine, 2015, 10(1): 52-62.
- [35]王倩倩. 基于机器学习的社交网络信息过滤及推荐系统实现[D]. 2019.
- [36]张旭媛. 长尾理论在电商企业中的应用——以“淘宝”为例[J]. 电子商务, 2019(7).
- [37]Bhavithra J, Saradha A. Personalized web page recommendation using case-based clustering and weighted association rule mining[J]. Cluster Computing, 2019,

22(3): 6991-7002.

- [38] 金晶, 怀丽波. 基于标签和协同过滤的改进推荐算法研究[J]. 延边大学学报(自然科学版), 2019(3).
- [39] Hong D G, Lee Y C, Lee J, et al. CrowdStart: Warming up cold-start items using crowdsourcing[J]. Expert Systems with Applications, 2019, 138: 112813.
- [40] 王春才, 邢晖, 李英韬. 个性化推荐系统冷启动问题研究[J]. 现代计算机: 中旬刊, 2015(10):36-38.
- [41] Papagelis M, Plexousakis D, Kutsuras T. Alleviating the sparsity problem of collaborative filtering using trust inferences[C]//International conference on trust management. Springer, Berlin, Heidelberg, 2005: 224-239.
- [42] Xu H K. A regularization method for the proximal point algorithm[J]. Journal of Global Optimization, 2006, 36(1): 115-125.
- [43] Zhou G, Zhu X, Song C, et al. Deep interest network for click-through rate prediction[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 1059-1068.
- [44] 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.
- [45] Sak H, Senior A W, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling[J]. 2014.
- [46] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[C]//Advances in neural information processing systems. 2017: 1024-1034.
- [47] Xu K, Li C, Tian Y, et al. Representation learning on graphs with jumping knowledge networks[J]. arXiv preprint arXiv:1806.03536, 2018.
- [48] Rendle S. Factorization machines[C]//2010 IEEE International Conference on Data Mining. IEEE, 2010: 995-1000.
- [49] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[J]. arXiv preprint arXiv:1205.2618, 2012.
- [50] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

作者简介及科研成果

作者简介

邢瑞，男，汉族，1995 年 5 月出生于山西省忻州市。2017 年 7 月毕业于吉林市北华大学数学学院信息与计算科学专业，主修数学分析和高等代数等课程，现就读于吉林大学软件学院，攻读软件工程专业型硕士研究生，研究方向为计算智能。

科研成果

- (1) An intelligent recommendation drilling platform for online computer education.
IEEE CSEI 2020. 2020.04.25. 已经录用. (EI 检索)

致 谢

光阴似箭，三年的硕士生涯转眼间已经要结束了，真的是感慨万分，虽然短暂，但是情谊却很绵长，师生情、同门情、室友情和同学情，是这些伴我走过了三年的时光。三年之中经历了许多，很多心酸和打击，但更多是幸福和愉悦，正是因为身边这些人，这些情的存在，才支撑着我不管遇到什么困难，都能勇敢地去面对，最终克服和解决。三年来我收获了很多，成长了很多，不只是年龄，还有心态，还有知识、经验，更多的还是为人处世之道。

首先，最应该感谢的是我的导师王岩老师，是他接纳了我，认可了我，让我加入了这个大家庭中，感受到了无比的温暖，他不只是我学习上的指导老师，更是我生活中的启蒙老师。在学术上，他认真严谨，有很强的科研能力，很爱学生提出问题并且积极探讨，对待问题他绝不含糊，甚至很严格，正是这样，才锻炼了我们实践中解决问题的能力；在生活上，他却是另一番做事之道，没有丝毫学术上那种严苛，更多的是温和幽默，他很关心我们每个学生的生活，也给我们带来了许多欢乐。

然后，要感谢的是我的室友，他们是我三年生活中最亲密的伙伴，他们帮助了我太多太多，相对来说，我年龄最小，心态不是很好，又比较幼稚一些，很多事情不懂，处理起来容易慌乱，他们从来都是容忍我，开导我，帮我想办法，从他们身上，我真的是看到了和学会了很多东西，收获了很多友情和感动。每次遇到什么愉快的或者不开心的事，首先是想到和他们分享，他们性格很好又幽默，几年生活中，和他们一起经历的事情也是最多的。

接下来，还要感谢实验室的师兄师姐和同辈们，从刚入学的无知，到现在自己也成了实验室的长辈，一路上，不管是在学术还是生活中，他们都给我提供了很多帮助和指导，分享了很多经验，正是这些宝贵的东西，才使得我少走了很多弯路，真的非常感激他们。

其次，感恩我的父母，上学的过程中，和他们相隔千里，但他们心里始终惦记着我，偶尔一个电话，包括身体是否健康，学校是否习惯，学习能否跟得上等，无一不透漏着对我的关爱。虽然对我的学习上不能给予很大的帮助，但

是努力供我上学，是我学习生活最主要的支持者。

最后，感谢我的母校吉林大学，给我提供了一个安稳舒适的学习环境，让我经历了许多有趣的事，认识了很多优秀的人，了解了一些伟大的榜样和事迹，并且从他们身上学到了很多高尚的品质。以及感谢学校和学院的各位老师，不辞辛劳，对我整个研究生期间的学术进度以及生活上的一些问题都一直关注着，叮嘱着，才使我不会落下脚步。

再次衷心感谢所有这些人，以及帮助过我的每一个人！