

**学士学位论文（设计）**

**题** **目** 基于协同过滤的个性化学习资源推荐系统设计与实现

**学生姓名** 杨繁 **学号** 202121020614

**学** **院** 计算机与网络安全学院（示范性软件学院）

**专** **业** 人工智能

**指导教师** 李瑞佳 **职称** 讲师

**答辩日期** 2025 年06 月 02 日 **成绩**

**基于协同过滤的个性化学习资源推荐系统设计与实现**

作者姓名：杨繁 专业班级：人工智能 2 班 指导教师：李瑞佳

## 摘 要

教育信息化的快速发展促使在线学习资源呈现爆炸式增长，学习者在获得便利的同时也面临信息过载和资源迷航等困境。传统学习平台多采用人工分类和热门推荐机制，这类方法难以面面俱到，用户个性化需求很难得到有效满足。协同过滤算法借助解构用户行为数据（学习记录、评分、浏览时长），能够精准匹配用户兴趣与资源特征。该技术目前是解决此类问题的关键手段之一。例如，图书推荐系统基于协同过滤算法分析用户借阅记录，显著提升了推荐准确率；电影推荐领域通过混合算法优化，有效缓解了数据稀疏性问题。如今的教育领域，个性化推荐系统既要实现资源精准匹配，又要结合学习者认知水平和学习目标等动态特征，算法设计面临更高的要求和全新的挑战。

针对传统协同过滤算法所面临的数据稀疏性以及冷启动方面的难题，本研究提出前后端协同优化方案。通过余弦相似度度量用户与资源相似度，构建用户画像与资源属性相结合的混合推荐模型；采用数据填充和降维方法缓解数据稀疏性问题，同时引入基于内容的推荐机制和评分引导策略应对冷启动问题；系统架构设计运用分层模式搭建，涵盖各个功能层；前端选用Vue.js框架搭配Element UI组件库；后端基于Spring Boot框架整合MyBatis-Plus，实现用户身份认证、各类资源管理及个性化推荐等核心功能。学习环境模拟表明，该系统在推荐准确率、覆盖率和多样性等指标方面具有较好性能，有效提升了用户学习资源获取效率及学习体验。研究成果一定程度上对实现教育资源的精准匹配有帮助，为个性化学习发展提供了实践的少许参照。

**关键词：**协同过滤；冷启动难题；数据稀疏性问题。

**A** P**ersonali**z**ed Learning Reso**u**rce**

**Recommendation System Based on**

C**ollaborative** F**iltering**

**Design and Implementation**

**Abstract:** Along with the swift advancement of educational informatization, online learning resources are experiencing explosive growth, and learners are facing the dual dilemma of "information overload" and "resource confusion". Traditional learning platforms rely on manual categorization or popular recommendations, making it difficult to meet users' personalized needs. Collaborative filtering algorithms, by mining user behavior data such as learning records, ratings, browsing time, etc., can accurately match user interests and resource features, becoming a key technology to solve this problem. Personalized recommendation systems not only need to focus on resource matching, but also need to combine dynamic features such as learners' cognitive level and learning goals, which puts higher demands on algorithm design. The objective of this research is to enhance and refine traditional collaborative filtering algorithms, so as to tackle the issues of data sparsity and cold start that occur during the recommendation procedure. Using cosine similarity to calculate user and resource similarity, a hybrid recommendation model is constructed by combining user profiles and resource attributes. Data sparsity is alleviated through strategies such as data filling and dimensionality reduction. Methods such as content-based recommendation and guided scoring are used to solve the cold start problem. The design of the system makes use of a tiered architecture. It encompasses a presentation tier, a business logic tier, and a data access tier. The front-end uses Vue.js and Element UI. The back-end uses Spring Boot and MyBatis Plus. It enables functions such as user registration, login, resource management, and personalized recommendations. The system performs well in recommendation accuracy, coverage, diversity and other indicators, effectively improving the efficiency and learning experience of users in obtaining learning resources. It is of great significance to promote accurate matching of educational resources and the development of personalized learning, and provides valuable reference for the construction of personalized recommendation systems in the field of education.

**Keywords:** Collaborative filtering; Cold start problem; Sparsity problem;

目录

[摘 要 I](#_Toc31540)

[第1章 绪 论 1](#_Toc14380)

[1.1 研究背景和意义 1](#_Toc21738)

[1.2 国内外研究现状 3](#_Toc32497)

[1.2.1 协同过滤算法的主要类型 3](#_Toc20106)

[1.2.2 数据稀疏性和冷启动问题 6](#_Toc19303)

[1.3 内容介绍本文研究内容及章节安排 7](#_Toc21449)

[1.3.1 本文研究内容 7](#_Toc14312)

[1.3.2 章节安排 7](#_Toc25896)

[第2章 理论基础 9](#_Toc7226)

[2.1 个性化推荐系统 9](#_Toc6013)

[2.1.1 个性化推荐系统分类 9](#_Toc28026)

[2.1.2 个性化推荐系统发展 10](#_Toc22689)

[2.2 协同过滤算法 13](#_Toc17031)

[2.2.1 相似度计算在协同过滤中的基础概念 1](#_Toc27319)6

[2.2.2 基于用户的协同过滤算法原理 15](#_Toc18602)

[2.2.3 基于物品的协同过滤算法原理 17](#_Toc21177)

[2.2.4 协同过滤算法的优缺点 18](#_Toc4744)

[2.3 冷启动和数据稀缺性问题及解决方案 19](#_Toc7191)

[第3章 系统设计与实现 21](#_Toc15270)

[3.1 系统总体设计 21](#_Toc8884)

[3.1.1 系统目标 21](#_Toc27910)

[3.1.2 系统架构 22](#_Toc23953)

[3.1.3 系统功能模块划分 23](#_Toc29599)

[3.2 数据库设计 24](#_Toc14292)

[3.3 协同过滤算法实现 25](#_Toc24523)

[3.3.1 数据预处理 25](#_Toc13077)

[3.3.2 相似度计算 27](#_Toc7383)

[3.3.3 推荐结果生成 28](#_Toc9501)

[3.4 系统实现细节 29](#_Toc10841)

[3.4.1 前端实现 29](#_Toc13423)

[3.4.2 后端实现 33](#_Toc11516)

[3.4.3 系统页面概览 34](#_Toc4699)

[第4章 总结与展望 38](#_Toc31452)

[4.1 全文总结 38](#_Toc9836)

[4.2 未来展望 38](#_Toc10164)

[致 谢 40](#_Toc15974)

[参考文献 41](#_Toc30558)

# 第1章 绪 论

## 1.1 研究背景和意义

随着时代发展，信息量极大膨胀。用户在面对海量信息时，不能快速从中获取自己有用的信息。针对这种现象，智能算法应运而生。近年来有关个性推荐算法的应用越来越广泛，根据用户的历史行为，对用户的喜好和目标行为，为用户推送信息，极具商业价值和挖掘价值。协同过滤算法最大的优点在于对推荐的对象没有特殊要求，它能够有效处理非结构化的复杂的对象，避免了内容的分析不完全性和不精确性，根据用户的历史行为推荐个性化的信息[1]。

信息技术发展日新月异，教育信息化进程飞速发展，在线学习平台应运而生并且呈现规模化增长态势。此类平台整合了涵盖课程视频、文档资料及练习题等类型的海量学习资源，形成多元化的学习选择体系。然而学习资源的大规模增长与碎片化特征引发显著问题，用户面临信息过载困境，难以快速定位符合其学习目标、认知水平与兴趣偏好的资源。传统推荐方式（如热门推荐和人工分类）因未能充分考量用户个性化需求，其推荐效果存在明显局限性。

协同过滤算法作为一种经典推荐算法，在电子商务、电影推荐等领域都取得了突出的效果[2]。将其应用于教育领域的个性化学习资源推荐，具有理论和实际意义。研究协同过滤算法在教育场景中的应用可以丰富和拓展推荐算法的理论体系，为教育数据挖掘和个性化学习提供新的思路和方法。设计并实现一个高效的个性化学习资源推荐系统，能够帮助用户解决信息过载问题，提高学习效率和学习体验，促进教育资源的合理利用和共享[3]。

人工智能作为计算机科学领域的一个分支方向，专注于探索能够使机器效仿人类智能行为的途径。历经数十年发展，人工智能涉及多个范畴，并在图像识别、语音交互以及决策优化等方面实现重大突破。个性化推荐系统借助大数据分析与机器学习算法，能自动挖掘用户的兴趣偏好并为用户进行推荐相关物品或信息的智能应用[4]。推荐系统通过分析用户的历史行为、社交关系、上下文信息等，构建用户画像，预测其对特定物品的喜好程度，从而提供个性化的推荐列表[5]。

协同过滤是个性化推荐常用的一种方法，具备多种优势：基于用户行为数据来做推荐，不需要对物品本身的详细描述或特征进行建模；根据用户的历史行为推测潜在的兴趣点；能够处理大量用户和物品的数据，特别是当用户和物品数量庞大时，依靠用户数据而不依赖于复杂物品特征描述可以更有效地进行推荐；能够实时反应用户兴趣变化，当用户行为发生改变时，推荐系统可以通过新的行为数据实时调整推荐结果；不依赖物品的内容信息，协同过滤对于内容特征难以获取或处理的情况效果较好，尤其适用于内容不丰富的场景。与此同时，其中缺点也影响颇深：对于新用户，由于缺少行为数据导致难以找到相似的用户，缺少足够的用户反馈致使推荐系统无法准确地为用户推荐；数据的稀疏状况会致使相似度的计算出现偏差，进而对推荐成效产生不良影响；计算用户之间的相似度时，计算量非常大，且随着数据集的扩展，推荐效率会受到影响；在大多数情况下，用户对物品的评分欲望是低迷的。即使用户数量庞大以及物品数量繁多，仍然只会有极少数的用户会对极少数的物品做出评价[6]，恰恰是数据存在的稀疏性，使得相似度的计算难以精准进行，最终对推荐效果造成负面作用；并且这种推荐方式十分依赖对用户或者物品的相似度计算，不同的相似度度量方法会导致不同的推荐效果。

本课题主要讨论以协同过滤算法为核心，结合场景特性与前端后端优化，实现一个个性化学习资源推荐系统。

## 1.2 国内外研究现状

协同过滤（Collaborative Filtering，CF）是个性化推荐中较为常用的一种方法，它核心的实现思路是借助对用户过往行为以及综合喜好的深度挖掘，从而达成向用户推荐其可能感兴趣的内容或者物品的目标。近年来国内外在协同过滤的研究主要体现在以下几个方面：

### 1.2.1 协同过滤算法的主要类型

在推荐系统较常用的算法类型中，协同过滤算法是被采用率相对较高的算法类型之一。目前协同过滤算法的主要推荐类型可分为三种：

（1）基于用户的方法主要针对用户，即根据匹配相似用户的兴趣和行为进行推荐。此方式是先寻觅出和目标用户兴趣相近的其他用户，接着据此去预估目标用户有较大可能会感兴趣的物品。这种方法优点是简单直观，而缺点是计算量大，尤其在用户数量非常庞大时的计算复杂度较高。王睿[7]对基于用户的协同过滤算法结合K-means算法并且引入用户特性进行改进从而缩短时间成本，使得推荐效率有一定提升。其中主要的算法思想是通过结合用户特征与K-means算法，对协同过滤算法进行改进，以实现更高效的用户聚类。具体而言，目标是将具有相似个人信息的用户归为同一类，从而形成k个用户簇。相较于传统方法，这种聚类方式可以显著减少查找时间。首先，基于用户簇的划分,选择距离目标簇最近的簇；接着，构建邻居矩阵，从这些簇中选取合适的最近邻居；最后，对于从未被选择或者评价过的目标，预测其被选择的概率或者被评价的评分，并将预测的评分或者概率按从高到低的顺序进行排序，进而向用户进行物品的推荐。

（2）基于物品的方法主要通过分析物品间的相似性来进行推荐。该方法通过研究用户对物品的评分，预测用户可能感兴趣的其他物品[8]。与基于用户的协同过滤相比，基于物品推荐算法在大数据环境中具有更高的效率，并且计算成本较低。优点是简单直观，无需用户特征，适应性强。基于物品的协同过滤方法简单且易于实现，逻辑清晰。不同于基于用户的协同过滤，基于物品的协同过滤方法并不依赖用户的特征信息，只需要用户的评分数据。适用于用户的数量较多，且物品数量相对较少的场景。但缺点是存在冷启动问题，稀疏性问题，扩展性差[9]。对于新物品或冷门物品，由于缺少足够用户对其的评价数据，因此找到与其相似的物品会更为困难。用户-物品评分矩阵通常会十分稀疏，较大比例的用户缺乏对物品进行评分的欲望，这会导致计算物品相似度时，稀疏的数据会造成影响。当物品数量非常庞大时，计算物品之间的相似度会过程会非常耗时且计算流程将会特别复杂。

罗洁等[1]采用各不相同的用户对不同的项目的评分作为用户项目评分矩阵并以此计算用户的相似度，根据所得出的相似度给所需用户推荐与其兴趣相对一致的用户的其他项目。在此基础上，通过将用户对不同项目的评分行为表示为矩阵，进而计算项目之间的相似度，并根据相似度为用户推荐与其偏好高度匹配的项目[10]。每个用户具有独立的特征向量，操作不受邻近用户偏好的影响，这使得能够向目标用户推荐感兴趣的、新颖的或冷门的项目，同时避免了冷启动和稀疏性问题对算法性能的影响[1]。

1. 基于模型的方法通过建立预测模型来分析用户的偏好，并且通过建立好的这些模型来预测用户可能感兴趣的物品。与基于记忆的协同过滤不同的是，该方法首先训练一个模型，然后使用该模型来预测用户的评分或偏好。基于模型的协同过滤的核心思想是通过数据训练一个模型[11]，这个模型能够针对用户的行为进行建模，从而形成用户与物品的预测评分。它主要利用机器学习、统计学和矩阵分解技术等方法来处理和分析用户和物品之间的关系。基于模型的协同过滤方法具备一个优势，即借助矩阵分解等相关技术，它能够对稀疏数据进行有效处理，可以减少对大量用户评分数据的依赖；通过训练模型，可以在大规模数据集上进行有效的推荐，具有较好的可扩展性；能够从数据中发现用户和物品之间的潜在关系或隐含特征，这对于提高推荐效果非常重要。但与此同时，也有不可忽略的缺点，训练模型通常需要较长的时间，尤其是在大规模数据集上，计算量非常大；相较于基于记忆的协同过滤方式，基于模型的手段一般来说更加错综复杂，模型难以解释性较差，不利于理解和调整；一旦系统内出现新用户或者新物品，鉴于历史数据的匮乏，模型或许难以给出有效的推荐内容（冷启动问题乃是所有推荐系统均面临的共性难题）。

Munson等[13]通过在国际上具备一定影响力的Netflix Prize竞赛的视角，阐述了协同过滤。以比赛早期的关键FunkSVD方法作为开始，在此基础上讨论了概率矩阵分解以及该模型的重要性和可扩展性。最后依据Netflix奖得主的策略，提供了与概率矩阵分解框架的比较，并详细分析了为什么一种方法在研究中被广泛使用，而另一种方法却没有受到采用。总的来说，不同的模型都在各自的角度上有助于帮助理解协同过滤在基于模型的类型中存在何种优劣。此外，他们整体提出了一系列显式反馈CF模型，其中一些也包含了隐式反馈。总的来说，这些模型独立地展示了处理RS数据的重要经验以及值得深思的教训[14]。

Trinh等[15]介绍了一种基于产品的协作筛选方法，该方法利用Apache Spark解决云计算环境中推荐系统的可扩展性问题。使用Spark的分布式计算能力，与传统的单机方法相比，我们的模型将训练时间提高了7.6倍，同时保持了均方根误差（RMSE）0.9的准确性。这些结果表明，并行和分布式技术在为大规模电子商务应用程序开发高效、准确的推荐系统方面的有效性。

### 1.2.2 数据稀疏性和冷启动问题

在本文当中，重点探讨的是优化协同过滤方法时所遭遇的两个关键问题。

数据稀疏性问题：数据稀疏性指的是用户和物品之间的评分矩阵中，大部分评分为空或缺失。用户通常只对一小部分物品进行评分，而对大部分物品意兴阑珊、兴致缺缺，这导致评分矩阵稀疏至极。稀疏的数据使得基于协同过滤的推荐系统不能找到足够的信息来进行有效的预测。造成这一窘境的主要原因是因为绝大部分用户只与少数物品交互或对其进行评分，因此大量的物品因为没有得到用户的评分导致数据矩阵稀疏，部分低频的物品（冷门物品）甚至可能出现没有评分数据的极端情况。这种现象具体的影响主要体现在两个方面：其一是它会导致推荐准确性下降，系统则无法提供有效的个性化推荐。其次在计算相似度时，共同评分用户的数量可能会过低，导致相似度计算难如登天。常见的解决方案一般有：矩阵补全，采用机器学习或深度学习方法，如SVD（奇异值分解）、MF（矩阵分解）等，通过补全稀疏矩阵来推断缺失的数据。增强数据，通过用户行为分析（如点击、浏览历史等）来补充评分矩阵的稀疏性。近邻加权，引入用户或物品的特征信息，结合基于内容的推荐，减轻稀疏性问题。

冷启动问题：冷启动问题指的是在推荐系统刚上线时，由于用户数据或物品数据微乎其微，导致系统很难进行有效推荐。冷启动通常分为以下几种情况：用户冷启动情况：当系统增添新用户之际，因缺少这些用户的过往数据或者评分信息，无法准确地为其推荐物品；物品冷启动：系统收录新物品时，由于缺乏用户对其的评分，系统无法判断该物品的受欢迎程度或者哪些用户可能对这个物品感兴趣；系统冷启动：系统初期整体缺少足够的用户和物品数据，无法进行有效的推荐。这些情况常造成的影响是冷启动会导致新用户或新物品无法获得推荐，甚至影响用户体验和系统的信任度。常见的解决方案一般有：基于内容的推荐，在用户冷启动问题中，使用用户的初始化输入的个人数据（如年龄、性别、兴趣等）和物品的特征信息（如类别、描述等）进行推荐。引导用户评分，通过引导新用户在初期提供更多的评分数据（如通过问卷调查或选择偏好的物品）来解决用户冷启动问题。

## 1.3 内容介绍本文研究内容及章节安排

### 1.3.1 本文研究内容

这项研究以协同过滤算法为核心，依据教育领域中个性化学习资源推荐的现实需求，重点解决数据稀疏性、冷启动及推荐效率等关键问题，构建面向在线学习平台的个性化推荐系统。主要研究工作如下：

数据收集与预处理：对模拟的教育场景中的用户行为数据（课程点击、学习时长、资源评分）和资源属性数据（学科类别、难度等级、知识标签）进行特征提取与结构化处理。

构建用户-资源交互矩阵，分析数据稀疏性特征，提出基于混合策略的矩阵补全方法（如结合内容特征的协同过滤模型）。

协同过滤算法改进与优化：鉴于传统协同过滤算法存在的不足，本研究提出一种将用户画像与资源属性相融合的混合推荐模型。本文着重探究在基于用户的协同过滤里引入学习者认知水平标签这一内容。

个性化推荐系统设计与实现：完成一个具备完整功能模块的个性化学习资源推荐系统原型，支持注册登录与权限管理，学习资源的多维度检索，基于协同过滤的个性化推荐列表与学习路径规划。

实验验证与性能评估：通过准确率（RMSE、MAE）、覆盖率、多样性等多维度指标构建评估体系，在模拟教学的环境中验证用户体验优化与学习效能并且尝试进行改进。

### 1.3.2 章节安排

本文基于协同过滤的个性化学习资源推荐系统设计与实现进行全面探讨，论文共分为四章，具体安排如下：

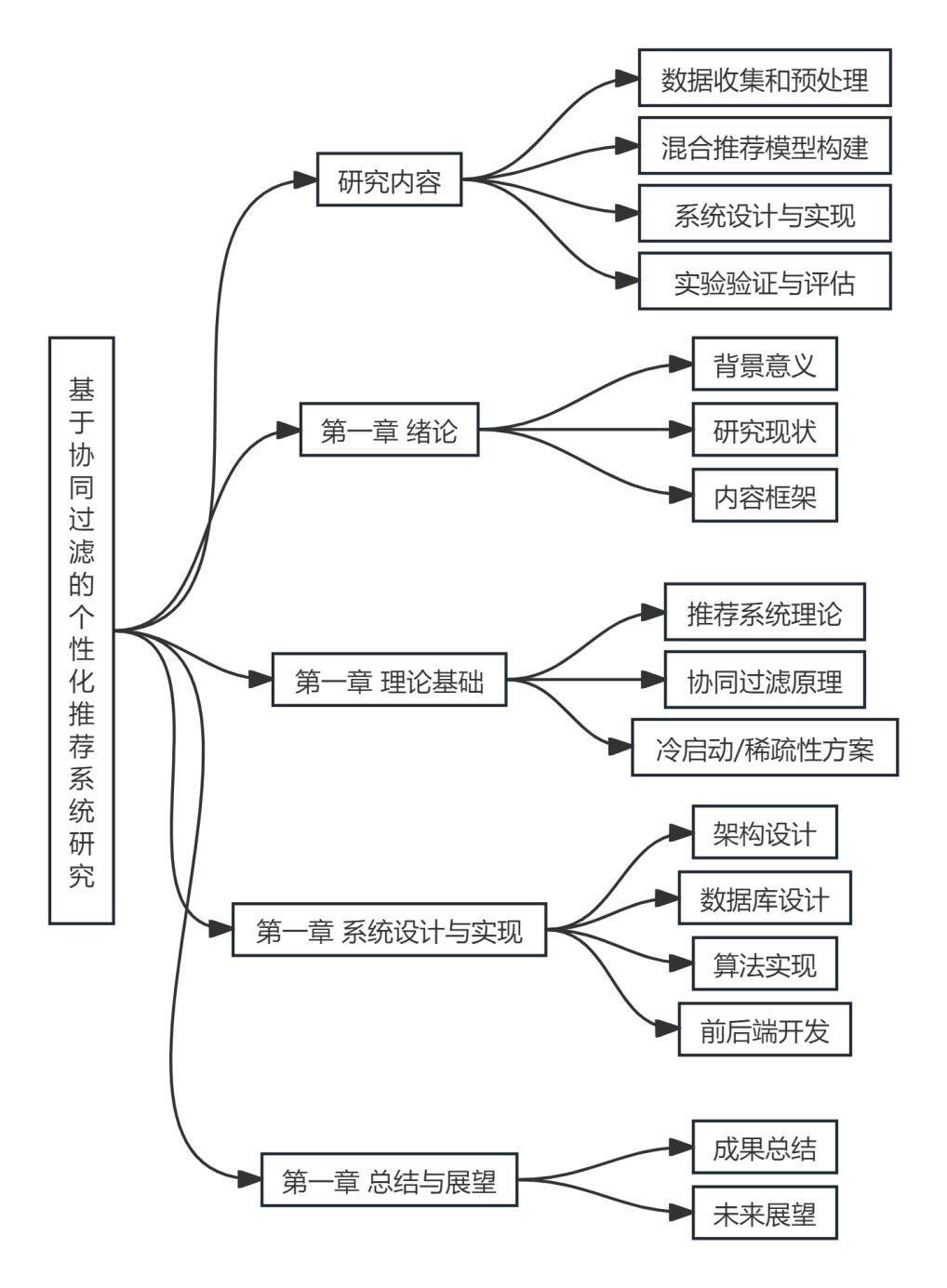
第1章为绪论，对研究背景和意义予以阐释，剖析国内外有关协同过滤算法以及推荐系统的研究现况，提出本文的研究目标与内容，并说明论文整体结构框架。

第2章为相关理论与技术基础，系统介绍协同过滤算法的主要类型与核心原理，包括基于用户、基于物品和基于模型的协同过滤方法；分析推荐系统关键技术（如相似度计算、矩阵分解、深度学习模型）；探讨教育场景中学习资源特征建模与用户画像构建方法。

第3章为说明系统的实现流程，搭建系统的架构以及功能模块，对数据预处理进行详尽说明、前后端搭建和推荐服务等核心环节；展现前端页面的架构、样式和交互效果；说明后端数据、接口对调及实现功能。

第4章为总结与展望，系统总结研究成果，客观分析研究局限，并展望未来工作方向，涵盖联邦学习框架下的隐私保护机制、多模态资源特征融合等拓展领域。

文章整体框架图1-1如下：



1-1 本文的框架图

# 第2章 理论基础

在这一章中，主要介绍本系统将进行设计完成的个性化推荐系统的理论基础，特别是传统推荐算法中的协同过滤算法，并对协同过滤的网络架构进行深入的解析。

## 2.1 个性化推荐系统

### 2.1.1 个性化推荐系统分类

个性化推荐系统根据不同的技术方法和应用场景可以分为几类，每种分类下包含具体的方法和应用实例，具体按照以下情况：

基于内容的推荐（Content-Based Recommendation）：该推荐机制的核心机理是通过物品的具体特征（文本、图像、标签等）与用户历史偏好的特征匹配程度实现推荐。典型的方法涵盖文本特征分析（TF-IDF、主题建模）、多模态特征建模（CNN视觉特征提取、NLP文本处理）以及基于行为日志的用户兴趣建模。常见应用场景有新闻资讯平台的内容分发，音乐流媒体服务的曲目推荐等。

混合推荐系统（Hybrid Recommendation）：其核心思想是结合协同过滤、内容推荐或其他方法，弥补单一方法的缺陷。典型策略有加权融合（协同过滤和内容推荐的得分加权求和，级联推荐（先用内容推荐过滤候选集，再用协同过滤排序），特征交叉（将用户行为数据和内容特征输入同一模型）。

协同过滤推荐（Collaborative Filtering, CF）：这种方法的关键理念，是依据用户群体行为数据展现出的相似特性来实施推荐，并不需要借助物品的具体内容。其关键逻辑在于，如果用户A和用户B对某些物品的偏好相似，那么A可能也会喜欢B喜欢的其他物品；同理，喜欢物品X的用户往往也喜欢与X相似的物品Y。而典型方法则包括用户协同过滤（计算用户之间的相似度，推荐相似用户喜欢的物品），物品协同过滤（计算物品之间的相似度，推荐相似物品），矩阵分解（将用户-物品评分矩阵分解为低维隐向量，解决数据稀疏性问题）[12]。该方式主要的应用场景包括电影推荐（Netflix）、电商推荐（亚马逊“买了又买”）[16]。

### 2.1.2 个性化推荐系统发展

个性化推荐系统的技术演进可分为四个阶段，从早期实验性系统到现代智能化平台：

（1）萌芽期（1990年-2000年），关键事件主要包括：1995年，首个个性化导航系统Web Watcher是由卡耐基梅隆大学提出的，而斯坦福大学则推出了LIRA系统，这标志着推荐技术的学术起点。1997年，Amazon首次尝试基于用户历史行为的书籍推荐，销售额提升35%,验证商业价值。2001年，IBM将推荐功能集成到WebSphere平台，推动企业级应用。

该时期的技术特点是以规则推荐和简单协同过滤为主，依赖人工标注和有限数据集。

（2）成长期（2000年-2010年），在萌芽期的基础上，出现了一定的核心突破，主要体现在协同过滤（CF）逐步成为主流，如Amazon基于项目的CF（Item-Based CF）显著提推荐稳定性。矩阵分解（MF）和隐语义模型（LFM）解决了数据稀疏性问题，Netflix Prize竞赛（2006-2009）一定程度上推动了算法创新。在具体的应用上出现了诸多扩展，例如电商（Amazon）、流媒体（Netflix）、社交（Facebook）等广泛采用推荐系统。

（3）智能化阶段（2010年-2020年），本阶段主要进度体现在技术的融合，首先是深度学习，包括Wide & Deep（Google）、YouTube的深度神经网络（DNN）处理用户序列行为，提升长尾推荐效果。其次是实时推荐主要有Apache Flink、Kafka支持流式计算，实现秒级更新（如新闻推荐）。最后是多模态推荐，重点在于结合文本、图像、语音特征（如抖音视频推荐）。

（4）全场景智能（2020年至今），该阶段的主要内容体现在前沿技术的进一步发展，包括有强化学习（RL），动态优化推荐策略（如阿里巴巴的DEAR模型）。图神经网络（GNN），建模用户-物品复杂关系（如社交推荐）。用户实现隐私保护，联邦学习（Federated Learning）在从网络上收集训练样本，在保护用户数据隐私的同时训练模型。

核心架构与技术模块，现代推荐系统通常采用分层架构，以下为典型结构（主要参考Netflix、YouTube等系统设计）：

（1）离线层（Offline），该层次功能主要包含批量处理历史数据，训练模型。其技术栈主要包括 Hadoop、Spark处理大规模数据，HDFS存储特征。基本示例流程是用户行为日志 → 特征工程 → 模型训练（如矩阵分解、深度模型）。

（2）近线层（Nearline），该层次的主要功能是准实时更新特征和模型（分钟级延迟）。其主要的技术栈包括有Apache Flink流处理，Redis缓存实时特征。通常的应用场景是用户实时点击反馈 → 更新用户画像 → 调整推荐策略。

（3）在线层（Online），该层次的主要功能是实时响应用户的请求（达到<100ms延迟）。其主要的流程包括召回（Recall），从海量候选集中快速筛选千级物品（如协同过滤、Embedding召回），排序（Ranking），融合多特征（用户画像、上下文）进行精排（如CTR模型）。重排（Re-Ranking），主要在于业务规则干预（去重、多样性控制）。

面对的挑战和未来发展趋势预测，结合过去的发展历程，分析解决的困难以及面对的挑战，简单分析未来的发展趋势：

（1）当前面临的主要挑战，常见的冷启动问题，该问题主要难点在于面对新用户/物品时缺乏行为数据，依赖内容特征或迁移学习效果较差。数据存在稀疏性，该问题体现在用户-物品交互矩阵往往稀疏，此时需结合图嵌入或跨域推荐才能达到理想效果。可解释性不足，黑盒模型（如DNN）在部分情况下难以解释推荐逻辑，这回影响到用户信任，产生较差的效果。

（2）未来发展趋势的简单预测，多模态融合，结合文本、图像、视频内容理解如GPT-4生成式推荐）。因果推理，识别用户行为因果关系，避免偏差（如点击诱饵问题）。端到端优化，从召回到排序一体化建模，减少人工干预。

典型成功案例示例：

（1）Amazon，其基于Item-Based CF的“买了又买”，混合内容特征提升长尾推荐。

（2）Netflix，分层架构（离线+近线+在线）支持亿级用户实时推荐。

（3）抖音，深度序列模型+多模态内容理解，优化视频推荐多样性。

现代推荐系统的架构图2-1，系统资源推荐流程图2-2如下：

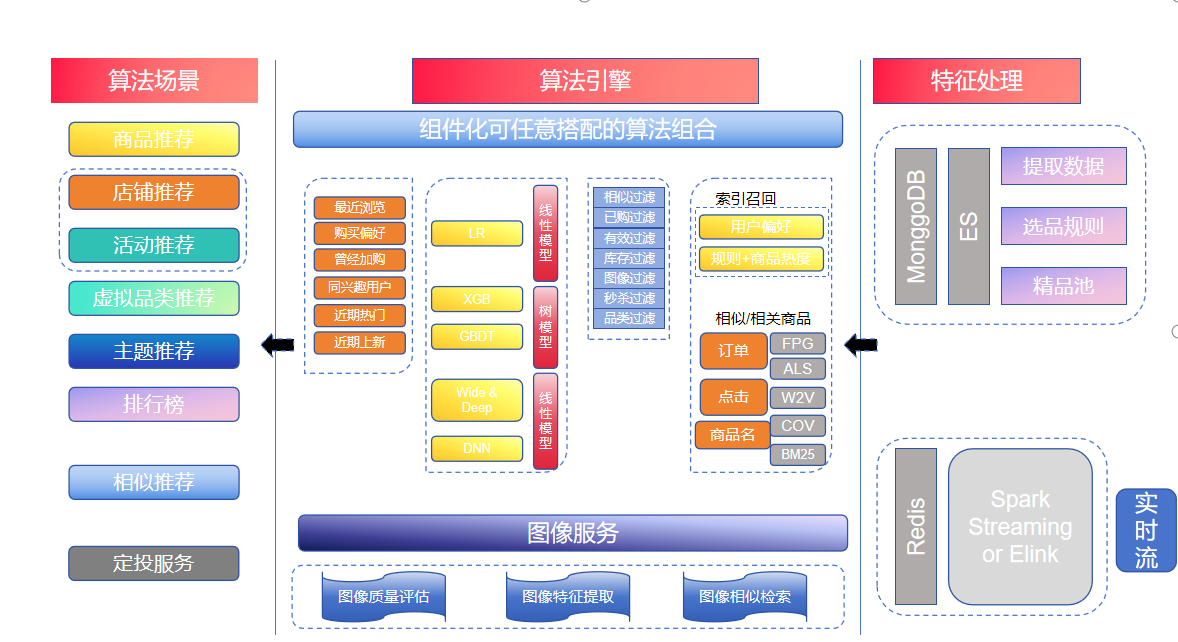


图2-1 现代推荐系统架构图

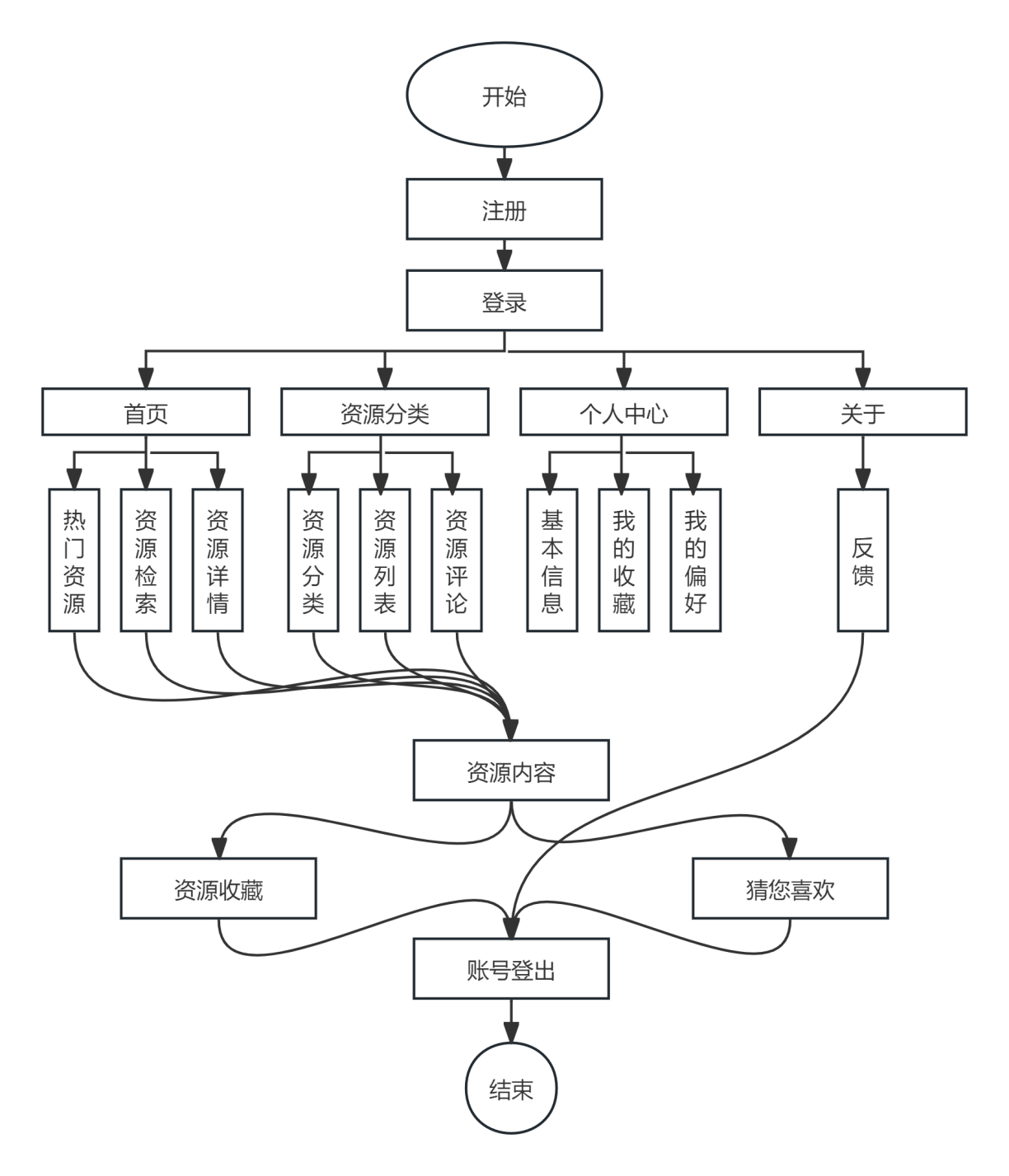


图 2-2 资源推荐系统流程图

## 2.2 协同过滤算法

在推荐系统领域，协同过滤（Collaborative Filtering，CF）属于核心算法之一，由Goldberg等人于1992年首次提出。依据推荐策略的差异，协同过滤大体可分为基于内存的协同过滤（Memory - based CF）与基于模型的协同过滤（Model - based CF）[14]这两大类别。其中基于内存的方法通过相似度计算实现推荐，计算复杂度较高但可解释性强；基于模型的方法通过机器学习建模用户偏好，具有更好的扩展性。

与传统的内容推荐方法相比，协同过滤本质上是通过挖掘用户行为数据的潜在关联完成推荐的决策，相较而言不再依赖物品的显式特征信息，该算法能够有效解决"信息过载"问题。

协同过滤系统架构主要包括数据预处理、相似度计算、预测评分和推荐生成四个核心模块，其整体流程如图 2-3所示。

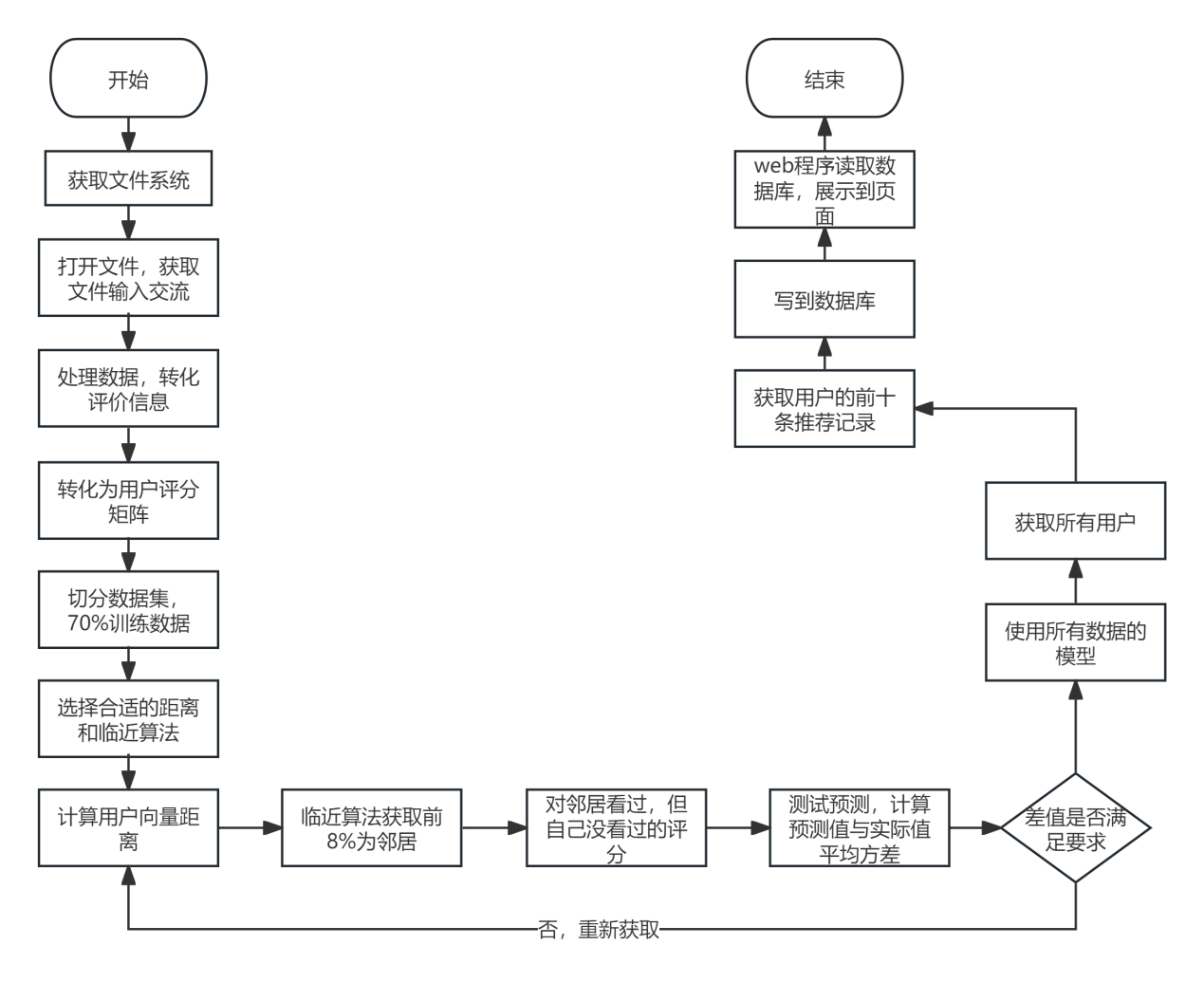


图 2-3 协同过滤算法流程

### 2.2.1 基于用户的协同过滤算法原理

相似度度量是协同过滤算法的核心环节。在协同过滤的过程中，系统需要依赖这些相似度来进行推荐。相似度计算包括用户相似度计算和物品相似度计算。

相似度计算的结果通常用一个介于-1到1之间的值来表示，值越接近1表示相似度越高，值越接近-1表示相似度越低，值为0时表示两者之间没有明显的相似关系。特定相似度计算方法的选择需要适配数据特性与任务需求，本节重点论述几种常见的相似度计算方法及其适用场景。

（1）计算用户相似度，常用的相似度计算方法有皮尔逊相关系数（Pearson Correlation Coefficient）、余弦相似度（Cosine Similarity）。本文皮尔逊相关系数作为例子，它衡量的是两个用户对共同物品评分的线性相关性[17]。设用户和用户对物品集合（即他们共同评分的物品集合）中的物品进行了评分，皮尔逊相关系数的计算公式为：

（2-1）

其中，表示用户对物品的评分，表示用户对所有物品评分的平均值，表示用户对物品的评分，表示用户对所有物品评分的平均值[18]。

通过一系列计算后，能够得到一个用户相似度矩阵，在这个矩阵里，其元素所代表的是任意两个用户间的相似度，依据相似度可以类比用户的爱好，用户矩阵表2-1，用户推荐逻辑简图2-4如下：

表 2-1 用户矩阵表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用户 | P1 | P2 | P3 | P4 | 平均 |
| A | 4 | ? | 3 | 5 | 4 |
| B | ? | 5 | 4 | ? | 4.5 |
| C | 5 | 4 | 2 | ? | 3.67 |
| D | 2 | 4 | ? | 3 | 3 |
| E | 3 | 4 | 5 | ? | 4 |

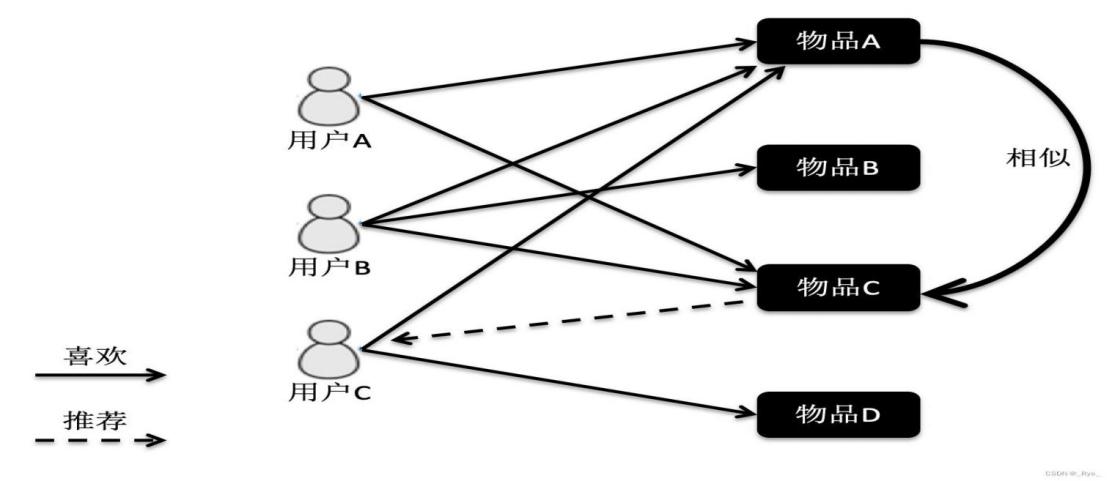


图 2-4 用户推荐逻辑简图

（2）依据用户相似度矩阵来寻觅相似用户，也就是为目标用户找出K个与其相似度较高的用户（K通常是一个预先设定的参数），这些用户被称为目标用户的邻居。

（3）生成推荐清单：按照邻居用户针对物品给出的评分，预估目标用户对尚未评分物品的评分。加权平均法是一种常用的预测手段，也就是依据邻居用户和目标用户的相似度，对邻居用户给出的评分实施加权操作，随后计算加权平均值，把这一数值当作目标用户对该物品的预测评分。预测评分的计算公式为：

（2-2）

目标用户为，物品为，邻居用户集合为。

最后，依据预测得到的评分，对物品加以排序，挑选出评分相对较高的物品，按照用户意愿挑出部分组成推荐列表，最后将内容以前端页面推荐给目标用户。

### 2.2.2 基于物品的协同过滤算法原理

基于物品的协同过滤算法的流程所计算的主体是物品，并非用户。

计算物品相似度：同样可以使用皮尔逊相关系数、余弦相似度等方法计算物品之间的相似度，得到一个物品相似度矩阵[19]。以余弦相似度为例，设物品和物品被用户集合（即同时对这两个物品进行评分的用户集合）中的用户评分，余弦相似度的计算公式为：

（2-3）

寻找相似物品：对于目标用户已经评分的物品，根据物品相似度矩阵找到与之相似的K个物品。

生成推荐列表：借助目标用户对已打分物品的评分，以及这些已评分物品和未评分物品间的相似度，来预估目标用户对未评分物品的得分。设目标用户为，未评分物品为，用户已评分的物品集合为，则预测评分的计算公式为：

（2-4）

最后，根据预测评分生成推荐列表。

### 2.2.3 协同过滤算法的优缺点

协同过滤算法的优点主要包括：

1. 可解释性强：推荐结果可以通过用户或物品之间的相似性进行解释，用户能更容易理解为什么会得到这些推荐。
2. 无需领域知识：无需对物品的特性展开深度认知，仅仅依靠用户的过往行为数据，就能够实施推荐。
3. 能够发现新的兴趣点：通过挖掘用户之间的相似性，可以发现用户可能感兴趣的新物品。

然而，协同过滤算法也存在一些缺点：

1. 数据稀疏性问题：于实际运用场景里，用户针对物品所给出的评分数据，往往呈现出极度稀疏的状态，这导致相似度计算的准确性出现下降，推荐效果收到影响。
2. 冷启动问题：当新用户加入系统或新物品进入系统时，因为系统缺乏足够完备的历史数据，无法准确计算其与其他用户或物品的相似度，所以难以进行有效的推荐。
3. 计算复杂度高：在遭遇大规模的数据集时，计算用户或物品之间的相似度寸步难行，需要大量的计算资源和时间。

## 2.3 冷启动和数据稀缺性问题及解决方案

冷启动问题主要指用户冷启动问题、物品冷启动问题和系统冷启动问题。

1. 用户冷启动：当一个新用户注册进入系统时，系统缺乏该用户的历史行为数据，无法准确了解其兴趣偏好。在这种情况下，系统为用户推荐的效果很难尽如人意。例如，在一个在线学习平台中，新用户刚注册时，系统对该用户感兴趣的课程无从知晓，也就无法为其提供称心如意的课程推荐。
2. 物品冷启动：当新物品加入系统时，缺少用户对其的评分或行为记录，系统没有依据确定该物品与其他物品的相似度，也难以将其推荐给合适的用户。比如，在一个图书推荐系统中，新出版的图书由于没有用户的评价数据，很难被推荐给感兴趣的用户。
3. 系统冷启动：推荐系统在上线的初始阶段，由于缺乏足够的用户和物品数据，整个系统的推荐效果会受到很大影响。

解决思路2-5如下：

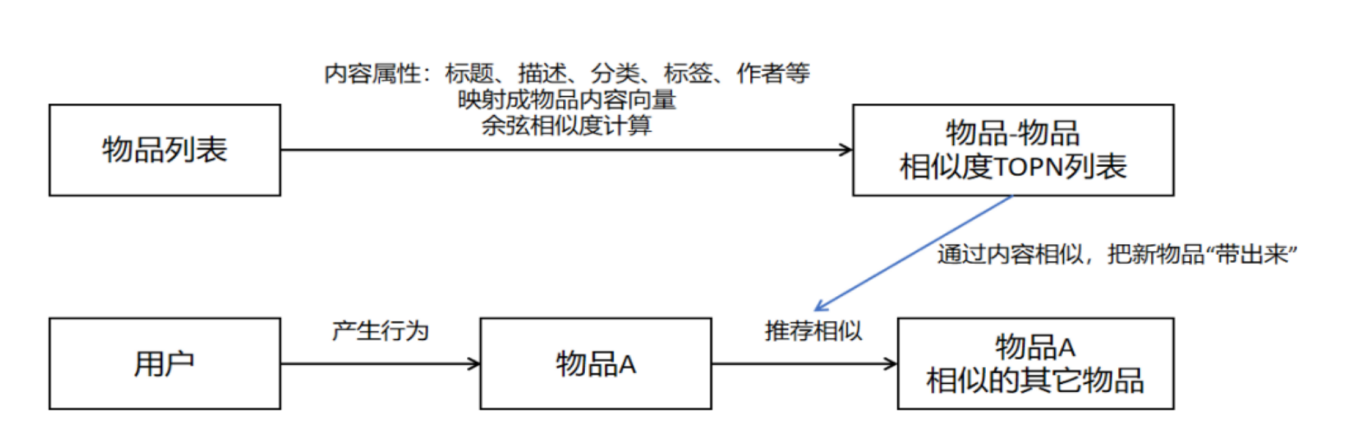


图2-5 冷启动思路

数据稀缺性是指在推荐系统中，用户对物品的评分或行为数据寥寥无几、寥若晨星，导致数据矩阵非常稀疏。最终的结果是相似度计算的准确性下降，因为在计算相似度时，需要有足够的共同评分或行为数据作为基础。

数据稀缺性的原因主要包括：

1. 用户行为多样性：不同用户的行为模式差异很大，有些用户可能只对少数物品进行评分或行为操作，提供的参照数据会很分散。
2. 物品数量增加：随着系统中物品数量的不断增加，用户对每个物品进行评分的概率会降低，数据稀疏性每况愈下。
3. 新用户和新物品的加入：新用户和新物品的加入会进一步加剧数据的稀疏性，主要原因是缺乏足够的历史数据。

本文对冷启动问题的缓和策略是通过用户注册阶段收集的基本信息（年纪、性别、职业等）构建初始推荐模型，典型场景如在线学习平台根据用户职业属性匹配编程类课程资源；对于新增的物品则采用元数据特征提取技术（本文选择设置课程主题、图书分类），结合用户历史偏好实现相似性推荐。

本文为解决数据稀缺性问题，采用以下几种方法：使用一些方法对稀疏数据进行填充，例如均值填充（是指用所有用户对某个物品的平均评分来填充该物品的缺失评分）、基于模型的填充（通过建立一个预演模型，根据用户和物品的其他特征来合理推断并预测补充缺失的评分）。设物品i的平均评分的计算公式为：

（2-5）

其中，表示对物品i进行评分的用户集合，表示用户对物品的评分，表示集合的元素个数。

优化相似度计算精度，可对原始数据矩阵实施降维操作。采用主成分分析或奇异值分解等算法能够一定程度降低特征维度，通过将高维稀疏特征空间映射至低维稠密向量空间，在保留关键信息分布的同时提升运算效率。针对数据稀疏性问题，选择拓展多源数据采集渠道，整合用户浏览轨迹、检索记录及收藏偏好等多维度行为特征，结合跨系统数据共享机制获取扩展的用户画像与物品元数据，以此增强数据集的完备性。

# 第3章 系统设计与实现

本章对系统的总体设计与最终实现效果进行论述，并详细介绍了系统运行环境、系统框架设计逻辑和系统中部分重要模块的分工、实现过程。

## 3.1 系统总体设计

### 3.1.1 系统目标

本系统的构想是在打造一个基于协同过滤的个性化学习资源推荐体系，其关键宗旨在于向用户供应精确且具个性化的学习资源推荐服务。着重探究如何切实解决协同过滤算法于实际应用时遭遇的冷启动难题以及数据匮乏问题，同时兼顾前后端的交互协作，最终实现提升用户学习效率与优化学习体验的成效。具体而言，系统需实现以下目标：

能够快速、准确地收集和分析用户的学习行为数据，包括浏览记录、收藏记录、评分等，从而深入了解用户的学习兴趣和需求。

针对新用户和新学习资源，通过有效的策略解决冷启动问题，确保新用户能够迅速获得有价值的推荐，新资源能够及时被用户发现。

采用合理的数据处理和算法优化方法，缓解数据稀缺性对推荐结果的影响，提高推荐的准确性和多样性。

提供友好、便捷的用户界面，方便用户与系统进行交互，如查看推荐资源、管理个人信息、反馈意见等。

### 3.1.2 系统架构

本文系统采用分层架构，主要4层，分为表现层、业务逻辑层、数据访问层和数据存储层；各层之间分工合作，各司其职，通过接口进行交互；系统的可维护性和可扩展性得到了显著提高。

表现层：承担用户进行交互的媒介，包括前端用户界面和客户端应用程序。前端界面采用响应式设计，支持多种设备访问，为用户提供直观、便捷的操作体验。客户端应用程序则可以提供更丰富的功能和更好的用户体验，如离线学习、推送通知等。

业务逻辑层：是系统的核心部分，负责业务的逻辑处理和算法的调度。该层包含推荐算法处理、学习资源管理、用户管理多个模块。业务逻辑层调用数据访问层的接口获取数据，并将处理结果返回给表现层。

数据访问层：主要负责与数据存储层进行交互，提供数据的增删改查等操作。该层封装了数据库操作的细节，为业务逻辑层提供统一的数据访问接口，提高了系统的可维护性和可扩展性[20]。

数据存储层：用于存储系统的各类数据，包括用户信息、学习资源信息、用户行为数据等。本系统采用关系型数据库MYSQL存储结构化数据，同时使用文件系统存储学习资源文件，确保数据的安全和高效存储[21]。

### 3.1.3 系统功能模块划分

在设计系统主要功能模块的过程中，我着重依据系统的实际需求以及既定的设计目标来展开工作，最后为5个模块，如下：

用户管理模块：此为用户登录用以验证身份的入口，它不仅承担着用户注册、登录、信息更改以及密码找回等任务，还对用户的权限与角色进行管理。借助这个模块，系统能够对不同类型的用户加以区分，依照类型为其提供不同的服务和功能。

学习资源管理模块：是主体数据的承载箱，主要负责学习资源的上传、审核、分类、标注等操作，确保学习资源的质量和规范性。该模块还提供资源的搜索、浏览、下载等功能，方便用户获取所需的学习资源。

推荐算法模块：是系统的核心模块，负责实现协同过滤算法和解决冷启动问题、数据稀缺性问题。该模块通过对用户行为数据的分析和处理，计算用户之间的相似度和资源之间的相似度，从而为用户生成个性化的推荐列表[22]。

数据的处理模块：是所有数据的集中处理中心，对用户的行为数据完成收集、清洗、存储和分析，为推荐算法提供高质量的数据支持。该模块还可以对数据进行挖掘和分析，发现用户的学习规律和趋势，可以为系统的优化和改进提供依据。

反馈与评价模块：支持用户针对推荐结果给出反馈并作出评价，如点赞、差评、评论等。系统收集用户的反馈和评价，而管理端和开发者可以及时调整推荐相关策略，最后提高推荐的准确性和用户满意度。

## 3.2 数据库设计

考虑到系统的数据规模、性能要求和开发成本等因素，我选用MySQL作为数据库管理系统。MySQL属于开源的关系型数据库管理系统，它运用了像索引优化、查询优化等多种优化手段，能够保证数据的安全和完整性[23]；它支持分布式架构和集群技术，可以方便地进行水平和垂直扩展，满足系统不断增长的数据需求。作为开源软件，它可以免费被使用和修改，降低了开发成本。

基于系统的功能要求以及数据架构，设计出了如下主要数据表 3-1：

表 3-1 数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 字段含义 | 主键 | 外键 |
| 用户表 | user\_id | bigint | 用户ID | 是 | 无 |
|  | username | varchar(50) | 用户名 | 否 | 无 |
|  | password | varchar(100) | 用户密码 | 否 | 无 |
|  | email | varchar(100) | 用户邮箱 | 否 | 无 |
|  | role | varchar(20) | 用户角色 | 否 | 无 |
| 学习资源表 | resource\_id | bigint | 资源 | 是 | 无 |
|  | title | varchar(200) | 资源标题 | 否 | 无 |
|  | description | text | 资源描述 | 无 | 无 |
|  | category | varchar(50） | 资源类别 | 否 | 无 |
|  | file\_path | varchar(200) | 文件路径 | 否 | 无 |
| 用户行为表 | behavior\_id | bigint | 行为 | 是 | 无 |
|  | user\_id | bigint | 用户 ID | 否 | 无 |
|  | resource\_id | bigint | 资源 ID | 否 | 是 |
|  | behavior\_type | varchar(20) | 行为类型 | 否 | 无 |
|  | behavior\_time | datetime | 行为时间 | 否 | 无 |
| 收藏表 | storeup\_id | bigint | 收藏 ID | 是 | 无 |
|  | user\_id | bigint | 用户 ID | 是 | 无 |
|  | resource\_id | bigint | 学习资源 | 否 | 是 |
|  | storeup\_time | datetime | 收藏时间 | 否 | 无 |

用户表（users）与学习资源表（learning\_resources）通过用户行为表（user\_b ehavior）建立多对多的关系，即一个用户可以对多个学习资源进行操作，一个学习资源也可以被多个用户操作。

用户表（users）与收藏表（storeup）建立一对多的关系，即一个用户可以有多个收藏记录。学习资源表（learning\_resources）与收藏表（storeup）建立一对多的关系，即一个学习资源可以被多个用户收藏。

## 3.3 协同过滤算法实现

### 3.3.1 数据预处理

数据采集阶段，系统采用前端页面与客户端应用程序进行用户行为数据的捕获，涵盖浏览轨迹、收藏条目及评分信息等维度。技术实现层面通过在界面层植入JavaScript脚本建立事件监听机制，实时捕获用户交互行为（包括点击、滚动及收藏等操作），并将相关操作日志传输至后端服务器。服务器端将接收到的行为数据持久化存储至数据库的用户行为表（user\_behavior）中。在数据预处理环节，针对原始数据可能存在的冗余记录、字段缺失及数值异常等情况，需执行数据清洗流程以提升数据质量。具体步骤包括：基于SQL查询语句识别并剔除用户行为表中重复条目；对于存在字段缺失的情况（如评分值未记录），采用均值插补或中位数填充等方法进行数据补全；针对异常数值（如超出预设范围的评分数据），实施数据过滤或数值修正操作。特征工程主要从用户属性和资源属性两个维度展开，提取协同过滤算法所需的有效特征。前者涉及用户兴趣维度、学习模式及历史轨迹等特征项，后者则包含资源分类标签、难度系数及流行度等参数指标。部分的数据去重以及计算用户的兴趣标签结果如表3-2，表3-3，图3-1：

表 3-2 部分数据预处理前

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| behavior\_id | INT | 行为记录唯一标识 |
| usr\_id | INT | 用户ID |
| resource\_id | INT | 资源ID |
| resource\_id | INT | 资源ID |
| category | VarChar(50) | 资源分类 |

表 3-3 部分数据预处理后

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| behavior\_id | usr\_id | resource\_id | resource\_id | category |
| 1 | 101 | 201 | 201 | Math |
| 2 | 102 | 202 | 202 | English |
| 3 | 103 | 203 | / | Music |

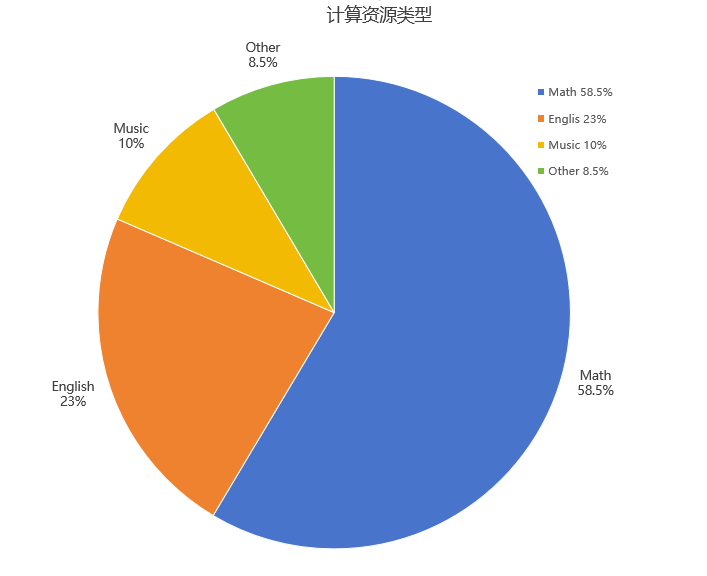


图 3-1 计算用户标签结果

### 3.3.2 相似度计算

相似度计算方法选择：本系统采用余弦相似度作为用户之间和资源之间的相似度计算方法。余弦相似度是一种常用的相似度计算方法，它通过计算两个向量之间的夹角余弦值来衡量它们的相似度，取值范围为[-1,1]，值越接近1表示相似度越高[29]。余弦相似度的计算公式如下：

（2-1）

相似度计算实现：根据余弦相似度的计算公式，实现用户之间和资源之间的相似度计算，用户之间相似度计算流程如图3-2，资源之间相似度计算类似。

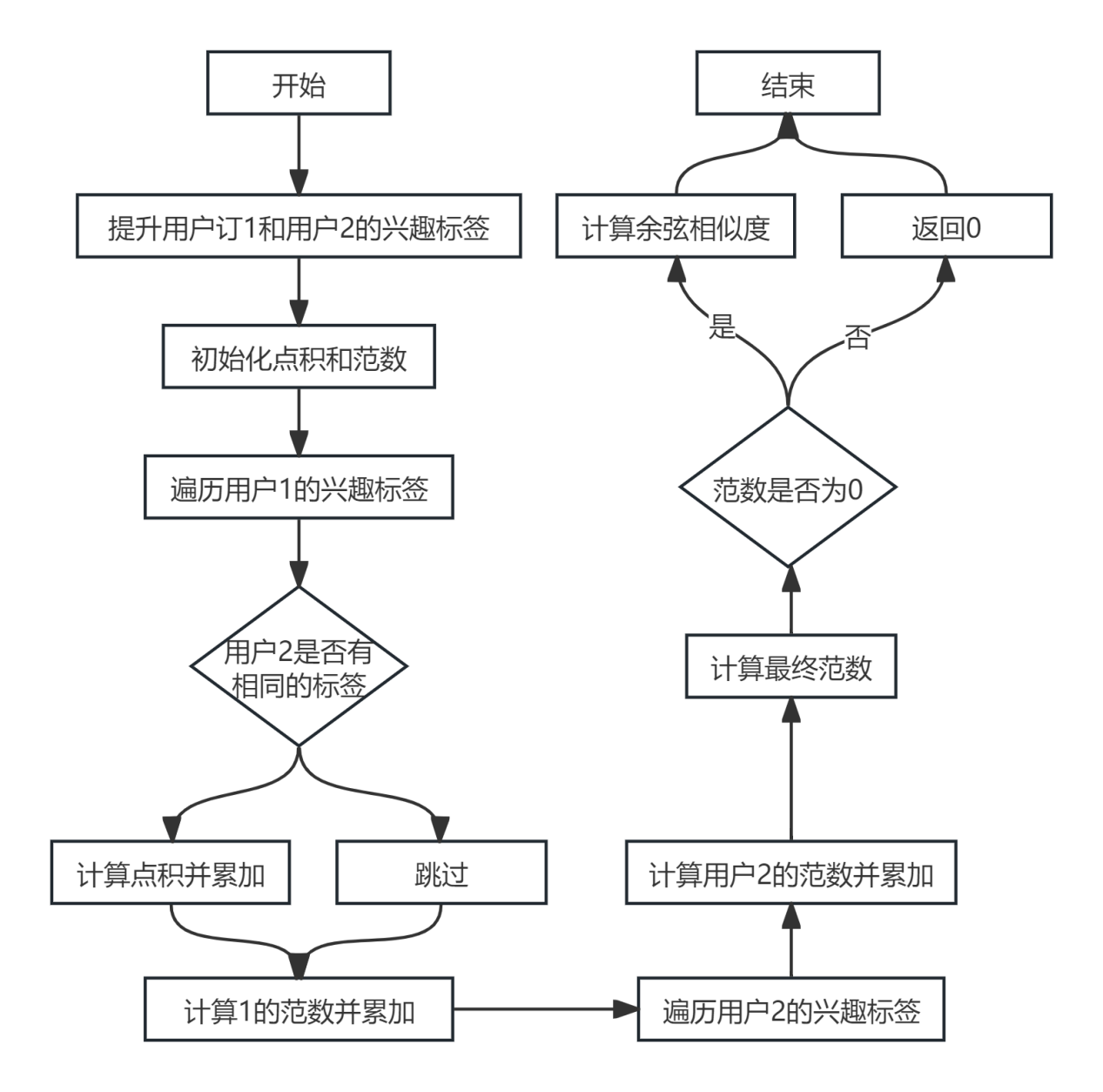


图3-2 相似度计算

### 3.3.3 推荐结果生成

推荐算法流程：本系统采用基于用户的协同过滤算法生成推荐结果[24]。主要流程是找出邻居用户喜欢的学习资源，但目标用户尚未接触过的资源；计算目标用户与其他用户之间的相似度；选择与目标用户相似度较高的K个用户（邻居用户）；根据邻居用户对这些资源的评分和相似度，为目标用户生成推荐列表[25]。

推荐结果筛选和排序：对生成的推荐结果需要进行筛选和排序，提高推荐结果的质量和准确性。筛选的过程大体是根据用户的偏好和需求，筛选出符合条件的学习资源，如资源的类别、难度等级等。而排序的过程主要是根据资源的评分、热度、相似度等因素，对筛选后的学习资源进行排序，将最符合用户需求的资源排在前面。

系统中有5个用户（ID分别为1,2,3,4,5）和6个学习资源（ID分别为101,102,103,104,105,106），目标用户ID为1，K值为2。

表 3-3 目标用户与其他用户相似度

|  |  |
| --- | --- |
| 用户ID | 与目标用户（ID为1）相似度 |
| 2 | 0.8 |
| 3 | 0.6 |
| 4 | 0.2 |
| 5 | 0.4 |

目标用户（ID=1）接触过的资源ID为[101,102]，邻居用户（ID=2）喜欢的资源ID为[101,103,104]，邻居用户（ID=3）喜欢的资源ID为[102,104,105]。

表 3-4 找出目标用户尚未接触过的资源

|  |  |
| --- | --- |
| 推荐资源ID | 得分 |
| 103 | 0.8\*资源得分（用户2对103） |
| 104 | 0.8\*资源得分（用户2对104）+0.6\*资源得分（用户3对104） |
| 105 | 0.6\*资源得分（用户3对105） |

最终，对推荐资源进行排序可得出结果：推荐资源ID 104得分最高，对应其资源名称

## 3.4 系统实现细节

### 3.4.1 前端实现

前端技术开发工具：本系统的前端采用Vue.js作为前端框架。Vue.js是一个JavaScript框架，量级轻、开发时事半功倍。其具有组件化开发、虚拟DOM、响应式数据绑定等优点，能够快速构建出高性能、可维护的前端页面或程序。同时，使用Element UI（一款基于Vue.js的组件库）作为前端组件库，Element UI是，提供了丰富的界面组件，让前端开发变得更加高效。

前端页面设计：系统前端主要包括首页、学习资源列表页、用户个人中心页等页面，各页面的功能和布局如下：

首页：展示系统的欢迎信息、热门学习资源推荐、搜索框等内容，方便用户快速找到所需的学习资源。

学习资源列表页：展示学习资源的列表，包括资源的标题、描述、类别、评分等信息,支持分页、筛选、排序等功能。

用户个人中心页：展示用户的个人信息、收藏记录、学习历史等内容，支持用户修改个人信息、管理收藏记录等操作。

前端与后端的交互：前端页面通过Axios（一个基于Promise的HTTP客户端）库与后端服务器进行交互，通常被用于浏览器和Node.js，具有拦截请求和响应、转换请求和响应数据等功能[26]。

前端页面在用户操作时，通过Axios发送HTTP请求到后端服务器，后端服务器接受并且处理这些请求，返回响应数据，最后前端页面会根据响应数据更新页面内容[27]。前后端交互流程图3-3如下：

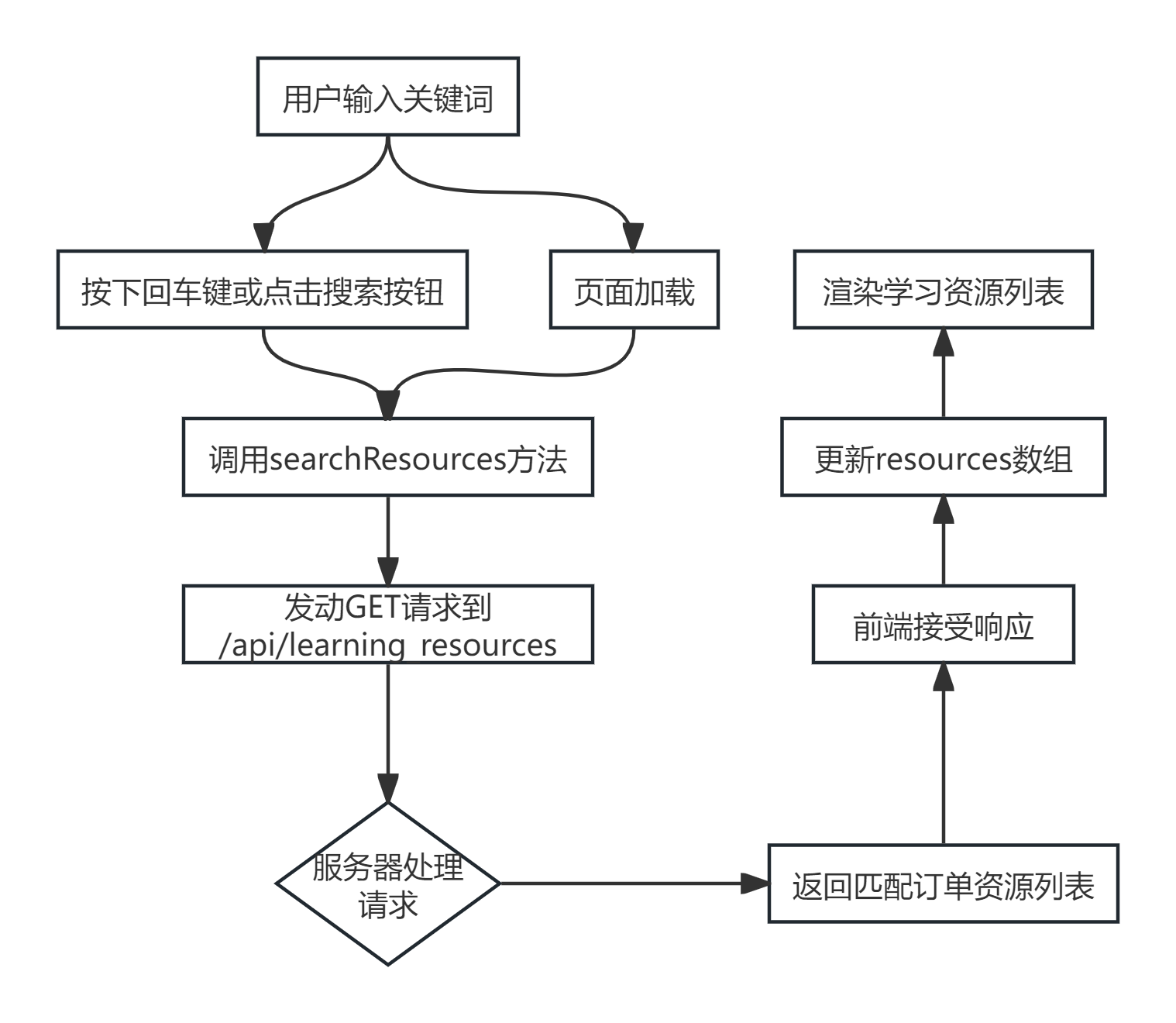


图 3-3 前后端交互

客户端可依照用户对页面的交互收集相关信息。

浏览行为记录：当用户在客户端浏览各种各样的学习资源（课程资源、视频资源、文本资源等）时，前端会记录用户的浏览历史，包括浏览资源类型、浏览时长、浏览时间等信息。这些信息会被实时发送到后端，推荐算法以收到的信息为基础为用户生成个性化的推荐列表。例如，当某个用户频繁浏览视频资源中的编程教程，推荐算法就会根据此行为，在客户端的推荐区域为用户呈现更多编程相关的资源。

搜索行为分析：用户在搜索框输入关键词进行搜索时，前端会将搜索关键词传递给后端。推荐算法会根据搜索关键词分析用户的兴趣点，并为用户推荐相关的资源。前端搜索功能可以分别从学习资源名称，资源作者，发布资源的教师名字这三个检索方向为基准。根据这三个方向的检索，可以准确的搜索到用户需要的资源，当检索词均不存在，则会默认推荐类似的资源。前端检索样式图3-5如下



图 3-4 前端检索组件

评论评分与点赞行为：当用户对某个资源进行评论评分或点赞操作时，前端会将该操作信息发送到后端。推荐算法会认为用户对该类型标签资源感兴趣，并根据该资源的特征和用户的历史行为，为用户推荐更多相似的资源。在客户端的推荐页面，会以排列靠前的方式展示这些基于收藏和点赞行为的推荐资源，如图3-6所示。



图 3-5 资源评价评分

推荐效果呈现：客户端的推荐效果主要会根据用户的个性化推荐结果，以卡片式布局展示推荐资源，每个卡片包含资源的标题、封面、简介等信息，用户可以直接点击卡片进入资源详情页。在资源列表页面，会根据推荐算法的排序结果，将推荐资源排在前列，方便用户快速发现感兴趣的资源。

### 3.4.2 后端实现

后端技术选型：本系统后端采用Spring Boot为后端框架，Spring Boot作为一个基于Spring框架的快速开发框架，很大程度简化了Spring应用的开发过程[28]。它具备自动配置以及嵌入式服务器等特性，助力开发者迅速搭建起具备高性能且易于维护的后端应用程序。同时，使用MyBatis-Plus作为数据访问层框架，MyBatis-Plus是MyBatis的增强工具，简化了MyBatis的开发过程，提供了代码生成、条件构造器、分页插件等功能，达到了提高数据访问层的开发效率和质量的效果。

在后端设计提供了一系列的RESTful API接口用于处理前端页面的请求。主要的接口如下显示：

用户注册接口：POST /api/users/enroll，用于用户注册。

用户登录接口：POST /api/users/login，用于用户登录。

学习资源列表接口：GET /api/learning\_resources，用于获取学习资源列表。

学习资源详情接口：GET /api/learning\_resources/{resource\_id}，用于获取学习资源的详情。

收藏接口：POST /api/storeup，用于用户收藏学习资源。

推荐列表接口：GET /api/recommendations，用于获取用户的推荐列表。

表 3-6 后端数据接口逻辑

|  |  |  |
| --- | --- | --- |
| 步骤 | 操作 | 说明 |
| 1 | 发送请求 | 前端据需求发送GET请求到/api/learning\_resources |
| 2 | 接受请求 | LearningResourceController接收前端请求，根据请求路径和参数调用相应方法 |
| 3 | 业务处理 | learningResourceService.search  Resources(keyword)方法搜索学习资源 |
| 4 | 返回响应 | 业务处理完成后，将结果封装在R对象中返回给前端 |

### 3.4.3 系统页面概览

此处插入系统架构图例如表现层有箭头指向业务逻辑层表示请求传递，客户端-教师页面：学生管理（可以锁定学生使其无法登录）如图3-7，标签增删改查如图3-8，课程资源增删改查如图3-9，教师管理功能如图3-10：

图 3-7 学生管理



图 3-8 标签增删改查



图 3-9 课程资源管理



图 3-10 教师管理

客户端-教师页面：教师在注册时，能够选择职称，包括高级讲师，讲师助教。不同的职称会影响到推荐的优先级，同管理员界面，教师界面同样拥有标签增删改查，课程资源增删改查这两个功能，注册界面图3-11如下：



图 3-11 客户端-教师注册

用户端-学生页面：学生用户在注册时，能够选择自己需要的学习资源类型，即标签（标签则由管理员和教师编辑）。不同的标签会影响到推荐的优先级，注册界面图3-12如下：

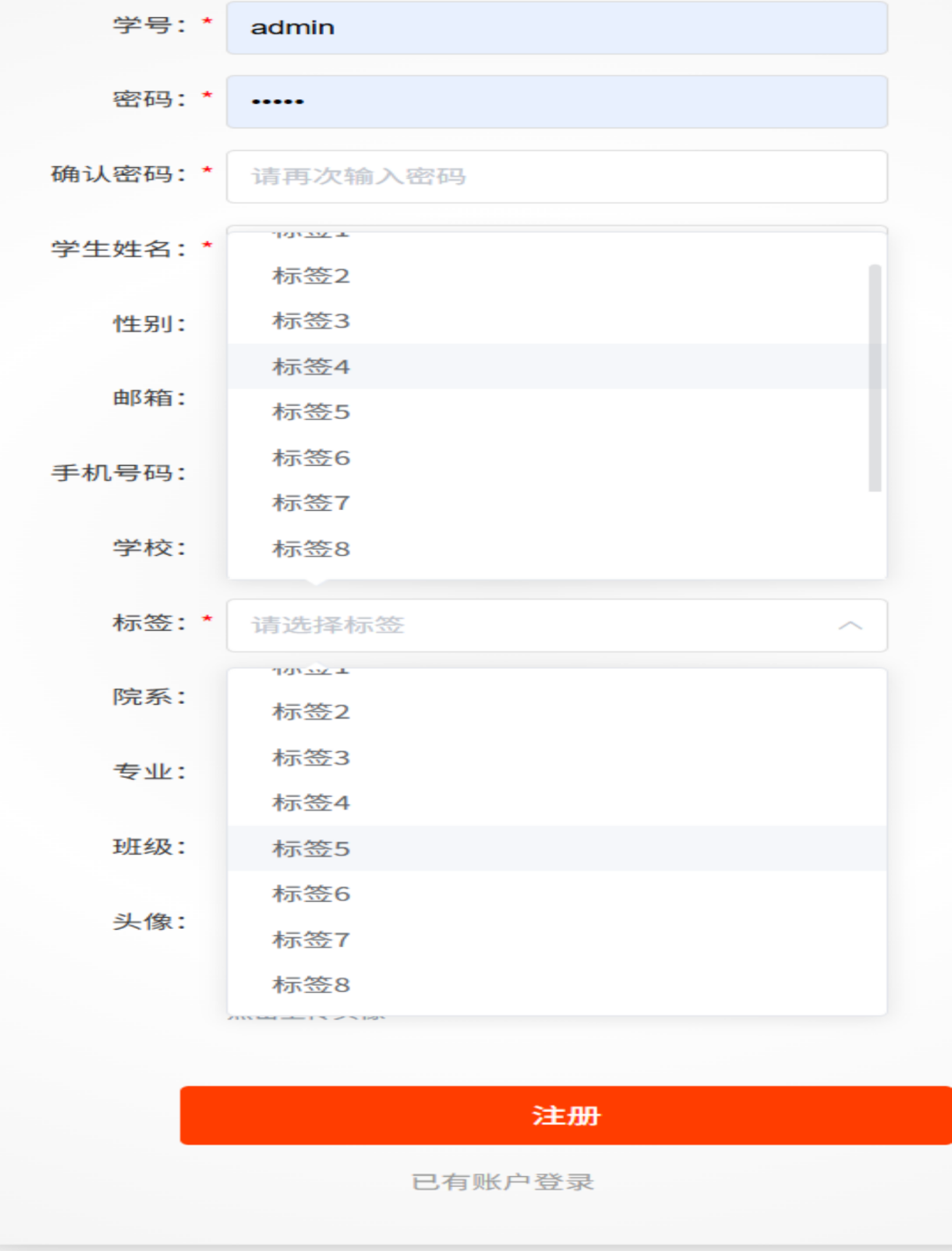


图 3-12 用户端-学生注册

与传统的内容推荐方法相比，协同过滤通过挖掘用户行为数据的潜在关联进行推荐决策，所以不需要依赖物品的显式特征信息。该算法具有发现长尾物品的独特优势，能够有效解决"信息过载"问题。前端页面整体如图3-13所示：

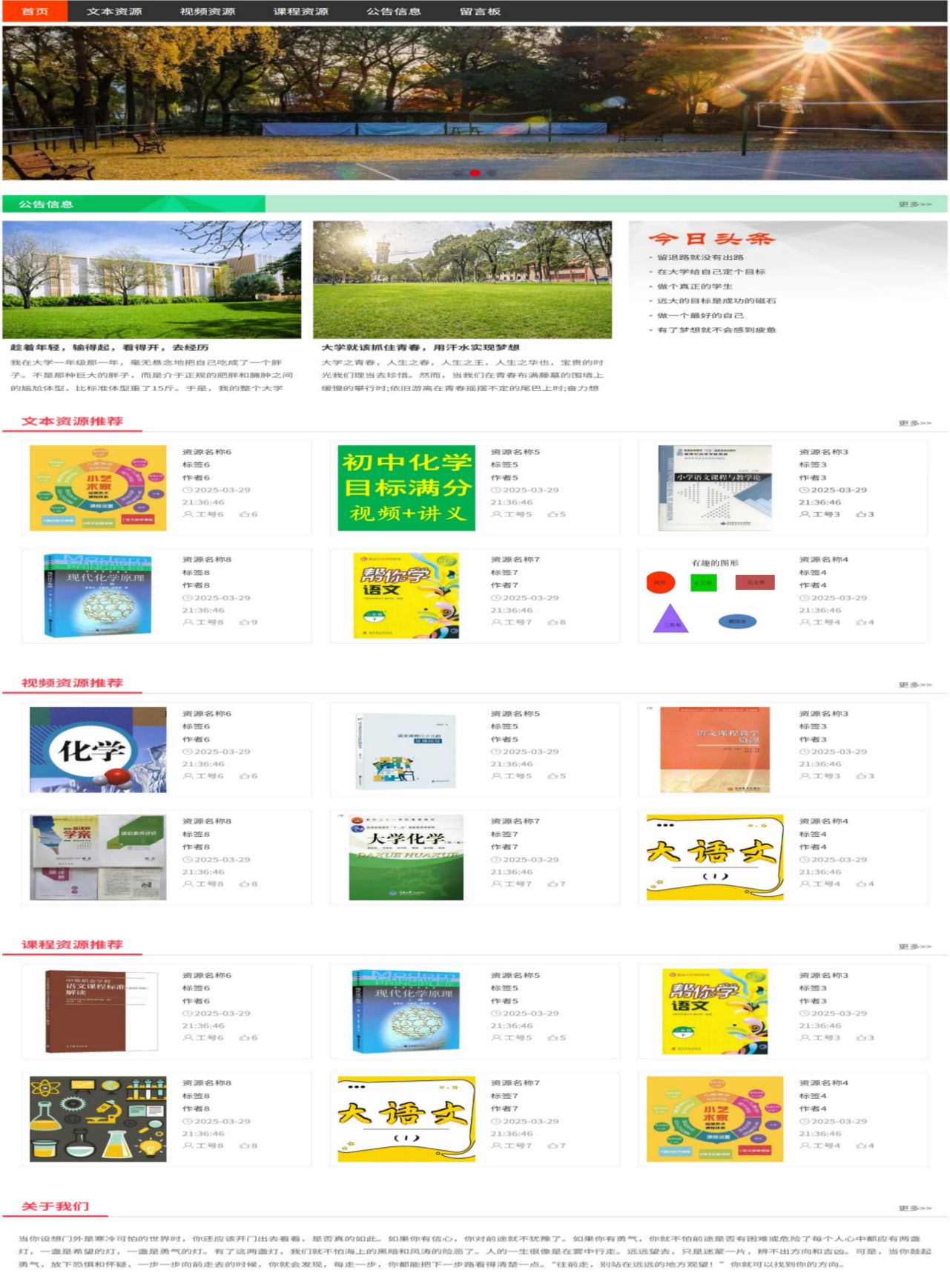


图 3-13 用户端界面

# 第4章 总结与展望

## 4.1 全文总结

本研究围绕教育信息化进程中用户面临的“信息过载”与“资源迷航”难题，以协同过滤算法为核心，进行深入开展个性化学习资源推荐系统的设计与实现工作。

分层架构的个性化学习资源推荐系统的完成过程虽有波折但不辱使命，囊括表现层、业务逻辑层、数据访问层和数据存储层。表现层采用响应式设计与丰富的客户端功能，提升用户交互体验；业务逻辑层集成多种核心模块，实现用户与资源管理、推荐算法处理等功能；数据访问层与存储层选用MySQL数据库与文件系统，保障数据高效存储与访问。通过前端Vue.js结合Element UI、后端Spring Boot搭配MyBatis - Plus的技术选型，完成了系统前后端开发，实现用户的注册、登录、资源管理、个性化推荐功能。

对传统协同过滤算法进行改进，采用余弦相似度计算用户或者资源之间的联系和相似度，引入用户画像与资源属性构建混合推荐模型，并针对数据稀疏性和冷启动问题，运用数据填充、降维、增加数据来源等策略以及基于内容推荐、引导评分、混合推荐等方法加以解决。

## 4.2 未来展望

尽管本研究取得的成果差强人意，但仍存在一些可以改进和拓展的方向。以下是对未来研究的一些展望：

算法优化方面：未来考虑引入联邦学习技术，在保护用户数据隐私的前提下，实现多源数据协同训练珠联璧合，进一步提升模型的推荐效果和泛化能力；探索将强化学习融入推荐策略，增强推荐的精准性和适应性。

在资源处理上，随着多模态学习资源的丰富多彩，后续研究将致力于引入多模态学习资源特征，融合文本、图像、音频等多种信息，更全面地刻画学习资源和用户兴趣，为用户提供更丰富、多元的推荐内容。

在系统性能与用户体验层面，持续优化系统架构，采用分布式计算、缓存技术等手段，降低计算复杂度；同时，深入开展用户调研，根据用户反馈优化界面设计和交互流程，简化系统的使用难度，提升其友好性，以更好地满足用户需求，推动个性化学习资源推荐系统在教育领域的广泛应用和发展。

# 致 谢

在本论文完成之际，我怀着无比感激的心情，向所有给予我帮助和支持的人致以最诚挚的感谢。

我要衷心感谢我的导师李瑞佳老师。在论文的选题、研究过程中，导师给予了我悉心的指导和无私的帮助。导师严谨的治学态度、渊博的学识和敏锐的学术洞察力，让我深受启发。从论文的构思到最终成稿，导师始终耐心地为我答疑解惑，提出了许多宝贵的意见和建议，帮助我克服了一个又一个困难，使我能够顺利完成学业。

我要感谢我的家人。他们始终是我坚实的后盾，无论遇到什么困难，他们总是无条件地支持我、鼓励我。他们的爱是我不断前行的动力，让我在求学的道路上充满信心。

我还要感谢我的同学和朋友们。在论文写作过程中，我们相互交流、相互学习，共同探讨学术问题。我们一起度过了许多难忘的时光，正是这些交流和分享，让我拓宽了思路，丰富了知识。感谢你们在我遇到困难时给予的帮助和支持，让我感受到友谊的温暖和力量。

最后，我要感谢成都理工大学和计算机与网络安全学院为我提供了一个良好的学习环境和丰富的学术资源。在这里，我不仅学习到了专业知识，更培养了独立思考和解决问题的能力。感谢学校和学院的每一位老师，你们的辛勤付出让我受益匪浅。再次感谢所有关心、支持和帮助我的人。

我将永远铭记这段难忘的大学时光，带着你们的期望和祝福，继续努力，追求更高的目标。

# 参考文献

1. 罗洁,王力.2023. 基于用户和项目的协同过滤算法的比较研究[J]. 智能计算机与应用, 44(10): 39-42.
2. 张鹏威. 面向中职学生的移动学习资源推荐方法及应用研究[D]. 浙江: 浙江工业大学，2019.
3. 石倩. 基于用户兴趣偏好的个性化学习资源推荐系统设计与实现[D]. 浙江: 华中师范大学，2018.
4. 陈华. Python在数据分析中的应用[J]. 上海轻工业,2024,(03):108-110.
5. 徐晓彤. 人工智能技术在个性化推荐系统中的应用与效果评估[J]. 信息记录材料,2025.
6. 林星星.基于Spark的商品个性化推荐系统的研究与实现[D].贵州大学,2021.
7. 王睿,姜学军.2023.基于改进协同过滤的图书推荐算法研究[J].信息技术与信息化,(04):149-152.
8. 林伟伟,刘波.分布式计算、云计算与大数据[M].机械工业出版社:201910.1172.
9. 张希翔. 面向语音环境的情感补偿推荐模型及方法研究[D].湖南： 湖南大学,2017.
10. 王喜宾,文俊浩.基于优化支持向量机的个性化推荐研究[M].重庆大学出版社:201704.181.
11. 张宗海. 融合多重注意力机制的深度推荐方法研究与应用[D].江苏： 江苏科技大学,2022.
12. 刘栋.基于深度学习的个性化APP分层推荐研究[D].湖南大学,2020.
13. Munson J ,Cummins B ,Zosso D .An introduction to collaborative filtering through the lens of the Netflix Prize[J].Knowledge and Information Systems,2025,67(4):1-50.
14. 黄传飞,万剑怡,王明文,等.协同过滤中一种项目综合相似度计算方法[J]. 山西大学学报(自然科学版),2015,38(02): 199-205.
15. Trinh T ,Nguyen H V ,Nguyen N , et al.Product collaborative filtering based recommendation systems for large-scale E-commerce[J].International Journal of Information Management Data Insights,2025,5(1):100322-100322.
16. Liu M ,Wang M ,Li B , et al.Collaborative filtering based on GNN with attribute fusion and broad attention.[J].PeerJ. Computer science,2025,11e2706.
17. 吴宾.多样化数据的推荐方法研究[D].郑州大学,2020.
18. 李倩,李诗瑾,徐桂琼.基于谱聚类与多因子融合的协同过滤推荐算法[J].计算机应用研究,2017,34(10):2905-2908.
19. 王婵.一种基于加权因子的混合协同过滤算法[J].电脑知识与技术,2018,14(09):14-16.
20. 夏鸿宇.学生课堂状态分析系统的后端设计及实现[D].电子科技大学,2022.
21. 祁蔚,胡旭鹏,鲁星言,等.SPDM系统在核电数字化设计仿真验证平台的应用研究[J].核动力工程,2024,45(S1):52-57.
22. 张良均,王路,谭立云,等.Python数据分析与挖掘实战[M].机械工业出版社:201512.533.
23. 伍之昂.Linux Shell编程从初学到精通[M].电子工业出版社:2015.
24. 张艺.基于用户偏好和用户聚类的协同过滤算法研究[D].西北大学,2018.
25. 程伟杰.基于用户偏好的协同过滤推荐方法研究[D].哈尔滨工程大学,2016.
26. 叶静.服务支付系统APP的设计与实现[D].首都经济贸易大学,2019.
27. 谈丽君.基于Robot Framework的API接口自动化测试系统的设计与实现[D].华东师范大学,2022.
28. 任志浩.面向智慧园区的物联网中台的设计与实现[D].长江大学,2024.
29. 李寅龙.节点结构-属性融合的社区发展方法研究[D].哈尔滨工程大学,2020.

成都理工大学

**学士学位论文（设计）诚信承诺书**

本人郑重声明所呈交的学士学位论文（设计）是本人在指导教师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得成都理工大学以外的其他教育机构的学位或证书而使用过的材料。与我一同工作的同学对本文研究所做的贡献均已在论文中作了明确的说明并表示谢意。

学生本人（签名）：

年 月 日