基于隐语义模型的文章分析与推荐系统的设计与实现

摘 要

基于隐语义模型的文章分析与推荐系统的设计与实现使用Java语言编写，本系统使用SpringBoot+Vue+MySql实现系统搭建与数据动态更新。在项目SpringBoot+Vue的基础上，同时使用协同过滤算法中的隐语义模型实现不同用户的个性化推荐，项目结构完整且合理。本设计以便于用户收集碎片化信息同时及时掌握近期社会热点词汇为出发点，每个用户既是信息的提供者，又是信息的受益者，形成用户之间互帮互助的良性循环。本系统运用基于B/S结构的系统开发模式，从文章标签和琐事需求出发，文章统计网络热词，标签便于筛选，琐事记录用户自己的个人生活，采用边设计边开发的方式，根据需求及时修正代码和功能，最终完成系统的详细设计。主要实现标签信息管理、琐事收录、短信通知、热搜榜生成、用户信息管理、后台数据库信息管理等，以上功能结构共同形成了本系统。设计过程中使用Vue分析并解决前端问题，后端使用注解开发@PathVariable、@RequestParam、@RequestBody获取参数，对前端axios发送的Get、Post等请求及时响应，完成前后端设计。

关键词：隐语义模型，梯度下降算法，标签信息管理，MySql，SpringBoot，Vue，MyBatis-Plus

Design And Implementation Of ArticleAnalysis AndRecommendation SystemBased On Cryptic Meaning Model

Abstract

The design and implementation of article analysis and recommendation system based on cryptic model is written in Java language, This system uses SpringBoot+Vue+MySql to realize system construction and data dynamic update. Based on the project SpringBoot+Vue. Meanwhile, the cryptic meaning model in the collaborative filtering algorithm was used to realize the personalized recommendation of different users, and the project structure was complete and reasonable. This design makes it easy for users to collect fragmented information and grasp the recent social hot words in time. Each user is both the provider of information and the beneficiary of information, forming a virtuous circle of mutual help among users. This project uses the system development mode based on B/S structure. Starting from the requirements of article labels and trivia, the article counts the hot words on the Internet, the labels are easy to filter, and the trivia records the personal life of users. The method of designing and developing is adopted, and the code and functions are timely modified according to the requirements, and the detailed design of the system is finally completed. The main realization of label information management, trivia collection, user information management,SMS notification, background database information management, hot search list generation, etc.The above functional structure together form a note management system. In the design process,Analyze and solve front-end problems using Vue in the design process.The back-end uses annotations to develop @PathVariable, @RequestParam and @RequestBody to obtain parameters. Timely response to Get, Post and other requests sent by front-end axios to complete the front and back end design.

**Keywords:** Latent Factor Model, Gradient Descent, MySql, SpringBoot, Label Information Management, MyBatis-Plus

目　录

[前　言 1](#_Toc17249)

[第1章 概述 2](#_Toc22951)

[§1.1 系统开发的背景及意义 2](#_Toc18037)

[§1.1.1 系统开发的背景 2](#_Toc23641)

[§1.1.2 系统开发的意义 2](#_Toc1078)

[§1.2 国内外同类设计的概况 3](#_Toc5298)

[§1.2.1 国外发展现状 3](#_Toc15602)

[§1.2.2 国内发展现状 3](#_Toc8517)

[§1.3 系统的主要开发内容 4](#_Toc24199)

[§1.4 系统开发工具与开发技术 4](#_Toc9152)

[§1.4.1 系统开发工具 5](#_Toc27146)

[§1.4.2 系统开发技术 5](#_Toc28287)

[第2章 需求分析 7](#_Toc20375)

[§2.1 系统的可行性分析 7](#_Toc17300)

[§2.1.1 经济可行性 7](#_Toc14782)

[§2.1.2 技术可行性 7](#_Toc22798)

[§2.1.3 社会可行性 8](#_Toc19746)

[§2.2 系统的总体需求分析 8](#_Toc13453)

[§2.3 系统的功能需求分析 9](#_Toc17989)

[第3章 系统设计 10](#_Toc16485)

[§3.1 系统模块介绍 10](#_Toc28318)

[§3.1.1 系统设计目标 10](#_Toc20184)

[§3.1.2 系统模块设计 11](#_Toc27495)

[§3.2 数据库设计 12](#_Toc9271)

[§3.2.1 数据库概念设计 12](#_Toc15297)

[§3.2.2 数据库表设计 13](#_Toc9956)

[第4章 详细设计 18](#_Toc10535)

[§4.1 管理员模块实现 18](#_Toc3212)

[§4.1.1 管理员登录 18](#_Toc17475)

[§4.1.2 后台管理员主页 18](#_Toc22038)

[§4.1.3 用户管理 19](#_Toc6564)

[§4.1.4 标签管理 20](#_Toc14357)

[§4.1.5 Redis管理 21](#_Toc25829)

[§4.1.6 隐语义模型 22](#_Toc15295)

[§4.2 用户模块实现 24](#_Toc32321)

[§4.2.1 网站前台首页 24](#_Toc1212)

[§4.2.2 用户注册登录 25](#_Toc23986)

[§4.2.3 热搜榜页面 26](#_Toc19019)

[§4.2.4 实时论坛页面 27](#_Toc10244)

[§4.2.5 标签和琐事创建页面 28](#_Toc18048)

[§4.2.6 个人信息和用户收藏页面 29](#_Toc7557)

[§4.3 隐语义模型 31](#_Toc13499)

[§4.3.1 关于隐语义模型 31](#_Toc16188)

[§4.3.2 隐语义模型与其他算法比较 31](#_Toc128)

[§4.3.3 隐语义模型计算公式 31](#_Toc26315)

[§4.3.4 模型测试样例 32](#_Toc4181)

[第5章 系统测试 34](#_Toc11879)

[§5.1 系统功能测试结果 34](#_Toc13397)

[§5.1.1 管理员及用户登录 34](#_Toc12682)

[§5.1.2 标签管理 35](#_Toc16025)

[§5.1.3 琐事管理 35](#_Toc13394)

[§5.1.4 用户收藏管理 35](#_Toc15269)

[§5.1.5 热搜榜管理 36](#_Toc28323)

[§5.1.6 Redis管理 36](#_Toc13176)

[§5.1.7 隐语义模型预测 36](#_Toc24962)

[§5.2 系统测试结论 37](#_Toc24565)

[结　论 38](#_Toc4180)

[参考文献 39](#_Toc27585)

[致　谢 41](#_Toc1235)

# 前　言

俗话说：“好记性不如烂笔头”随着互联网技术的不断发展，国人生活的生活方式得到了极大的改善，从以前的写信交流到现在的即时沟通软件，从线下排队买票到现在的线上预约订票，互联网给人们生活带来更好的体验。但是，互联网的发展也使得知识的积累变得丰富起来，面对每天都有大量信息产生的互联网，纸质的写笔记方法已经不适合现在的年轻人，随时随地、方便快捷的笔记网站才是问题的正解。针对大众的这种需求，许许多多的笔记网站就由此孕育而生，笔记网站不仅仅是用户记录生活的方式，同时也是用户了解世界，拓宽视野的方式之一，所以一个具有记笔记和浏览公共笔记的共享网站是十分必要的。

目前国内外对隐语义模型都有研究，且已经出现了一些效果优异的网站或APP个性化推荐服务系统模式应用于文本管理和文章分析，对于创新行的个性化算法研究和实践还在初级阶段，应用于一些简单服务场景。国内基于隐语义模型的文章分析和个性化推荐服务体系的研究也在同步进行，但相比于国外模型各有千秋。目前来看，国内发展潜力巨大。

基于隐语义模型的文章分析与推荐系统的设计，将提供文章管理和个性化推荐系统，目的是更好的服务用户。既能够满足用户随时随地的录入碎片化信息，又能使用户在闲暇之余了解网络热词，每个用户即使提供者又是受益者，共同构建一个良性循环。针对隐语义模型、微服务架构等技术，查阅相关硕博论文以确定相关技术的可实现性，最初根据简单的功能结构，搭建页面框架并编写功能模块，随着设计分析内容的增多，系统模块逐渐丰富，如短信验证、热搜榜模块、token验证。用户可以随时随地的浏览实时资讯、发布文章、登录注册、个人中心编辑、浏览收藏、标签查询，琐事记录等，其中琐事模块是用户私有信息，不对外公开展示。本系统管理员可以对现有用户和文章进行管理等等。

# 概述

## 系统开发的背景及意义

### 系统开发的背景

随着互联网的普及，社会信息化进程和5G技术的推进，第三方服务类技术有了更多的使用和应用场景。从以前到图书馆借阅图书到现在得微信读书，从街口市井聊八卦到抖音短视频，从写信邮寄到动手发即时消息，人们获取消息得途径越来越方便，足不出户便可一览热点新闻，人们获取消息的代价也逐渐变低。因此，如何高效处理日常文本信息，如何将碎片化阅读内容实体化，赋予日常琐碎时间以新的意义，应该是人们重点关注的问题。

另一方面，随着4/5G网络的飞速发展，社交媒体APP已经融入到了人们生活的方方面面，大家每天使用碎片化时间浏览短视频、电影、朋友圈时，一定会接触到很多新事物，获取大量的新知识，如果手机上的便签、记事本记录这些内容，很臃肿且不便于分类。因此，针对记录并管理新事物的需求，各种笔记APP相继推出，为了更方便快捷的实现需求，本系统应运而生。

### 系统开发的意义

通过构建本系统，人们成功地将各个文章分类、精彩内容和用户笔记等信息整合至用户可视化界面。用户可以通过浏览器轻松浏览和检索这些信息。同时，该系统还融入了隐语义模型，能够深入剖析用户发布的文章内容和阅读偏好，为用户提供更为精准的文章推荐。在这个平台上，用户不仅可以发布自己的笔记、心得和文章，还能与其他用户进行互动交流。此外，系统还提供了便捷的标签管理功能，使用户能够轻松整理和检索自己的内容。

值得一提的是，本系统不仅为用户提供了一个展示和分享的平台，还通过深入的数据分析，助力用户更好地了解自己的写作习惯和读者喜好，从而不断优化内容创作。这一系统的推出，不仅促进了文章内容的传播与交流，还提高了用户内容管理的效率，为用户带来了全新的使用体验。同时，也进一步推动了文化创作和分享的热情，为用户创造了一个既实用又有趣的在线内容管理平台。

## 国内外同类设计的概况

### 国外发展现状

隐语义模型的研究在国外学术界一直非常活跃。许多知名的研究机构和大学，如微软研究院，都在这一领域进行了深入研究。这些研究不仅涵盖了隐语义模型的理论基础，还包括了各种优化算法和扩展模型等。例如，研究者们提出了基于矩阵分解的隐语义模型（LFM）等。在商业应用方面，许多国际知名的互联网公司，如谷歌、亚马逊等，都将隐语义模型广泛应用于个性化推荐系统、搜索引擎、广告推送等领域。这些公司不仅在模型应用上取得了显著成效，还不断对模型进行扩展和优化，以适应日益复杂和多样化的用户需求。

国外笔记网站或APP的市场规模在近年来持续扩大，以笔记记录类APP为例，根据GIR (Global Info Research)的调研，2021年全球笔记记录APP的收入大约为592.8百万美元，并预计到2028年将达到1644.7百万美元，年复合增长率（CAGR）为15.7%。人们对笔记记录的需求，推动了Evernote、Notion等国际知名厂商发展，使得国外有关笔记网站或笔记APP呈现出市场规模持续扩大、功能创新与技术应用不断进步、跨界合作日益加强以及用户群体与市场需求多样化的特点。

总的来说，国外隐语义模型在学术研究、商业应用和技术创新等方面都取得了显著的进展。随着技术的不断发展和应用场景的不断拓展，隐语义模型的作用在未来将不断体现。而针对文章管理系统，使用隐语义模型分析用户兴趣，做到个性化推荐并精准推送，是该模型的一项重要应用场景，同时也形成了文章+隐语义的有益结合和良性循环。

### 国内发展现状

隐语义模型在国内学术界的研究也日益活跃。国内多所顶尖大学和研究机构，如清华大学、北京大学以及中国科学院等，都在积极推广隐语义模型并创新出很多相关优化算法。这些研究涉及隐语义模型的基础理论以及在不同领域的应用。例如，研究者们已经成功将隐语义模型应用于自然语言处理、情感分析和个性化推荐等多个方面。在商业领域，国内互联网大厂如阿里巴巴、字节跳动等也在积极将隐语义模型集成到产品和服务中，如电商推荐、实时新闻推送和广告投放等。

近年来，国内笔记类网站或APP市场也呈现出稳步增长的趋势。随着移动互联网的深入普及和5G的广泛使用，国人对高效、便捷的笔记管理服务需求不断提升。相关数据显示，国内笔记类应用的用户量和活跃度都在逐年上升，且市场前景广阔。国内已经涌现出一批优秀的笔记APP，如印象笔记、蓝奏云笔记等，通过不断创新和优化个性化推荐功能，满足了用户多样化的需求，进一步推动了市场的发展。

综上所述，国内在隐语义模型的研究和应用，以及笔记类网站或APP市场的发展上都取得了积极的进步。隐语义模型在精准推送、个性化服务等方面展现出了强大的潜力，而笔记类应用则通过结合这些先进技术，提升了用户体验和服务质量。展望未来，随着技术的不断创新和市场需求的持续增长，隐语义模型和笔记类应用有望在国内迎来更加广阔的发展空间。

## 系统的主要开发内容

本系统是为了建立一个将文章管理和分类、生活琐事、学习新知等功能有机融合的简便轻量化网站管理系统。系统管理员登录后台管理界面后即可进行用户信息管理、文章分类管理、文章信息管理、热搜榜管理等。用户登陆后，即可使用任何浏览器实现浏览实时资讯、发布文章、登录注册、个人中心编辑、浏览收藏、标签查询，琐事记录等。

## 系统开发工具与开发技术

本系统立足于网络信息化背景，致力于提升碎片化时间的使用效率，因此精选了IDEA 2021开发工具和MySql数据库管理工具Sql yog。本系统后端服务器框架使用SpringBoot轻量级框架，网站前端使用Vue框架，同时使用MyBatis-Plus数据库处理技术来完成本系统的开发，最终打造出一款便捷高效、用户体验优异的文章信息管理系统，满足广大用户随时随地学习和记录新知识的需求。

此外，本系统引入了隐语义模型，这一技术能够智能地根据主用户的历史浏览数据（对于新用户，则根据其年龄和偏好等信息）来寻找相似兴趣的用户群体。系统会筛选出那些主用户未曾浏览，但在相似兴趣用户中高频阅读的文章。这些被选中的文章信息将被存入数据库，并精准推送给主用户，从而实现个性化的文章推荐服务。

### 系统开发工具

本系统采用IDEA 2021作为开发工具来构建项目，其是一款简单高效的Java Spring开发工具，在编码过程中，可以使用其代码提示功能加快开发速度，同时还使用了阿里巴巴最新推出的AI开发助手“通义灵码”帮助简化开发过程。此外，当SpringBoot与MySql数据库结合使用时，能够动态加载并智能完成配置，从而避免了大量不必要的手工配置文件编写工作。

### 系统开发技术

1. SpringBoot

SpringBoot是一个便捷的应用开发架构，其以Spring框架为基础，并内置了服务器功能，不再需要本地部署tomcat。该架构拥有自带的独立Web容器和丰富的Starter依赖库，使得开发者能够轻松地自动装配Spring的各个模块，同时SpringBoot全家桶无缝集成其他出色的框架，而无需繁琐地手动配置XML文件，使用yml文件简单配置后即可使用。此外，SpringBoot还支持各种第三方库，从根本上简化了开发过程。对于初学者而言，SpringBoot无疑是一个友好且理想的选择。

1. MyBatis-Plus

MyBatisPlus是在MyBatis的基础上进行了增强，整合了MyBatis的所有api，同时还提供了一套丰富的Java API来帮助用户更高效地开发数据库应用，对部分api进行了进一步的强化，从而显著提升了开发效率。

1. Vue

Vue.js是一款渐进式的JavaScript框架，本系统使用的是Vue2。其为开发者提供了一种便捷的方式来创建用户友好的可视化界面。通过使用Vue.js，无需再直接操作前端页面的DOM元素，也不需要手动编写原生js代码，只需引入JQuery，调用Vue的生命周期函数和JQuery相关api，即可快速实现想要的功能。在使用Vue时，借助Vue提供的数据双向绑定和循环遍历功能，可以轻松地管理和更新界面数据，并通过axios获取数据库信息，最后将数据展示给用户。目前，Vue的生态链正在不断完善，展望未来，其发展趋势仍然非常乐观。

# 需求分析

## 系统的可行性分析

### 经济可行性

本系统软件在配置方面非常便捷，只需在Windows系统环境下安装并配置好IDEA 2021的Java开发环境和MySql数据库环境即可。在硬件需求上，任何一款搭载Windows系统的笔记本电脑都能满足其运行要求。这样的低门槛配置意味着几乎所有企业都能轻松部署和使用该系统，极大地降低了初始投入成本。在深入分析了系统的成本与潜在收益后，发现本系统不仅能为网站管理者带来显著的经济效益，更能大幅提升文章信息的整合效率。该系统通过其强大的管理功能和个性化推荐算法，帮助管理人员实现对文章信息的快速、高效管理，展现了极高的实用价值。

本系统还采用了隐语义模型技术，实现了用户的个性化推荐，进一步提升了系统的智能化水平和用户使用体验。隐语义模型能够深入挖掘文章中的潜在语义信息，为用户提供更加精准的内容推荐和个性化服务。这一技术的应用，不仅增强了用户体验，也使得本系统在信息管理领域具备了更强的竞争力。

### 技术可行性

本系统是基于B/S架构设计的，服务器使用Java代码编写，即使用Java语言进行开发。为了提升开发效率和系统的易用性，本系统选用了SpringBoot这一快速整合的第三方框架技术，其实现了约定大于配置、即配即用的特性。SpringBoot通过提供简化的Maven依赖管理、内置服务器以及自动装配等功能，极大地简化了开发流程。此外，本系统采用的微服务架构代表了未来软件开发的发展趋势，且允许个人开发过程更加专注于细分的工作领域，通过独立的技术栈实现更安全、高效的开发。

在前端部分，本系统选用了热门的Vue框架。Vue的轻巧和易用性，以及其提供的组件化开发思想和开发模式，为项目研发带来了极大的便利。通过与ElementUI的结合，本系统不仅技术可行性高，而且能够为用户提供丰富、有效的界面体验。

### 社会可行性

本系统拥有独特的个性化UI界面设计，用户只需遵循简洁明了的操作提示，即可轻松掌握本系统的使用方法。该系统主要面向信息接受能力较强的青年和中年用户，因此用户友好性极高，无需担心新用户无法快速上手问题。同时，系统为管理员提供了直观、有效、易用的界面设计，只需简单阅读操作指南，管理员便可迅速熟悉系统操作。

笔记管理类应用正逐渐成为行业发展的新趋势，市场前景广阔。综合以上因素，本系统在技术、经济和社会层面均展现出良好的可行性。其未来的发展趋势清晰明确，不仅网站易于操作上手，更有望为用户和管理员带来便捷、高效的笔记管理使用体验。

## 系统的总体需求分析

本系统是通过对用户添加的所有文章信息资源整合从而为每位用户提供学习新知途径的系统，随着文章信息的增多，涉及方面的扩大，用户的服务要求也在不断提高。伴随着计算机网络的普及，人们也习惯通过互联网在短时间内随时随地的获取到文章资讯，然后根据自己的需求搜索不同标签来寻找最佳答案。因此本系统提供了更优更快的文章信息管理服务，以满足用户浏览资讯，学习新知等多方面需求。

该系统中最关键的问题是如何实现文章的个性化推荐。隐语义模型应能够深入分析用户的历史记录和行为，从中提取出用户的兴趣偏好，将文章的特征与用户的兴趣进行精准匹配，推荐与用户兴趣点最符合的文章。网站应实时获取数据库信息，实时更新推荐结果，保持推荐的时效性和准确性。除此之外，本系统在使用隐语义模型实现文章的个性化推荐时，需要满足推荐准确性、系统性能、可扩展性与灵活性以及安全性与隐私保护等多方面的需求。这将有助于提升系统的用户体验和实用价值，为用户提供更加智能、个性化的文章推荐服务。

## 系统的功能需求分析

为了满足本系统的实际业务需求，并实现管理员对文章资源信息的有效管理以及为用户提供便捷的资讯服务，本系统采用了浏览器端与服务器端的体系结构设计。在本系统中，用户角色被明确划分为一般用户和网站管理员两类。根据全面的需求分析，为管理员设定了以下功能权限：管理用户信息、热搜榜内容、文章信息及分类等。这些权限确保管理员能够全面掌控网站内容，提供高效的资讯服务。对于前台设计，本系统主要着眼于为用户提供流畅的信息浏览体验。用户可以根据自己的兴趣，轻松浏览各类信息分类，从而快速获取所需的资讯内容。这样的设计旨在提升用户的使用体验，同时确保网站信息的有效传递。

# 系统设计

## 系统模块介绍

本系统分为后台系统管理员模块和前端用户模块两大模块。两个模块的介绍如下：

前端用户模块包括：用户注册登录、短信验证、天气信息浏览、重置密码、个人信息管理、文章搜索管理、文章管理、文章信息处理、琐事管理、文章热搜榜浏览、实时论坛浏览、个人收藏管理等。

后台管理员模块包括：管理员登录token拦截、热搜榜内容管理、重置密码、管理员个人中心、用户信息管理、文章信息管理、redis数据库同步管理、隐语义模型管理等。点击相应管理页面，即可查看相关管理模块信息，管理员还能根据搜索字段进行模糊查询，根据标题、标签分类等查询等。

### 系统设计目标

本系统主要是为了建设一个便于用户记录信息、获取信息、便于操作的笔记网站，主要的设计目标包括以下五点：

1. 各类标签文章信息的发布及管理

用户可以发布个人编写的文章信息，给文章添加标签分类，标签可以是年份、类别、APP名称等等，可对已发布内容修改或补充，不仅可以最大限度的满足用户对分类的需求，也使用户对已添加内容更快的搜寻或编篡。对于一些喜欢记录生活用户来说，可以添加琐事信息，琐事没有标签限制，但可描述此时此刻的心情且可书写内容总结，进一步提升用户的使用体验，提高服务质量。

1. 注册和登录功能

本系统用户需要通过手机获取验证码的方式进行注册验证，提高了系统的安全性并解决恶意注册的问题。注册后，用户只需输入正确的账户、密码、验证码即可登录本系统。

1. 用户浏览热搜榜

用户登录本系统之后，除了能浏览当地的天气、定位等信息，还可以浏览热搜榜，该榜单记录了当前点击量最高的前N个文章，代表了当前网站所有用户最关注的文章信息。

1. 用户查看和收藏文章、浏览实时论坛

用户可以在网站上查看感兴趣的文章，对于喜欢的文章可点击收藏按钮，并在个人信息的“海纳百川”页面查看收藏内容，不喜欢可取消收藏。针对用户自己添加的文章和琐事将在个人信息的“多彩生活”页面展示。用户可访问实时论坛，实时论坛是按文章录入的时间顺序从新到旧展示的，方便用户掌握最新的文章信息。

1. 用户个人信息的修改

用户首次登录后，可以在个人信息页面完善自己的昵称、姓名、邮箱、出生日期等信息。修改个人信息后，在用户下次登录时将更好的在其他页面显示相关内容，提升用户的使用体验。

1. 管理员后台管理

管理员登录后台页面，可对用户的基本信息做简单修改（不包括用户详细信息）或删除用户，可查阅所有文章内容并删除不良文章内容，可查阅redis内容并管理相关内容，可手动实现redis与数据库的同步操作。

### 系统模块设计

本系统共分为网站后台数据管理，用户前端界面展示，隐语义模型三大模块，其中隐语义模型主要用于用户的个性化推荐服务，优化用户服务体验。系统的相关功能模块结构图如图3-1所示。

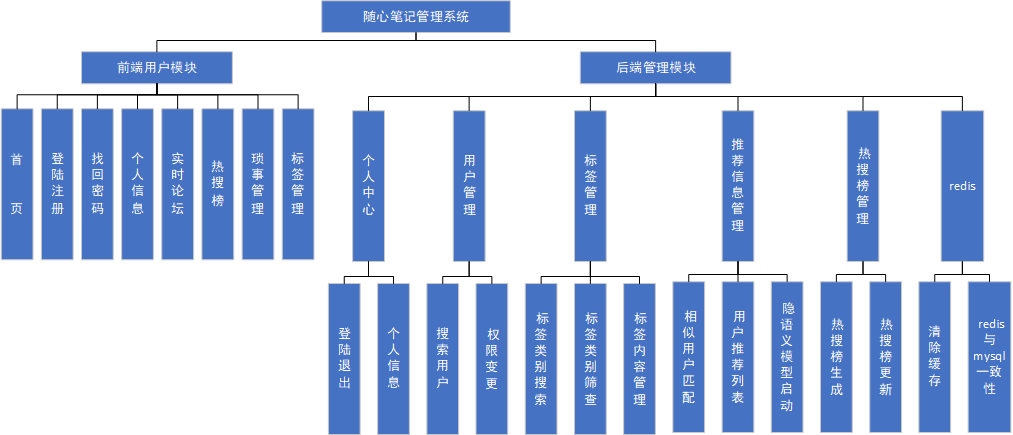


图3-1 本系统功能模块图

## 数据库设计

### 数据库概念设计

数据库的概念设计是数据库设计的核心，是整个项目的数据逻辑结构，为整个数据库系统的构建提供了基础。本系统的顺利运行，正是依赖于精心设计的数据库。没有详尽的数据库设计，本系统的数据便无法流畅地动态展示在浏览器界面上。在数据库设计的过程中，本系统遵循数据库的三大范式，确保设计的合理性和高效性。首先，为了更加直观的理解数据库结构，构建ER关系实体模型图。本系统使用数据库驱动，将系统连接数据库，确保数据的流动与系统的功能相契合。在此基础上，保证数据结构的合理性，开发的思路清晰。

本系统详细设计了数据库所需包含的数据结构关系和数据对象中的数据项。为此，本系统数据库设计了一系列数据表，包括用户表、系统管理员表、用户详细信息表、标签信息表、标签类别关系表、琐事信息表、用户收藏信息表、赞助订单表、用户行为信息表、用户相似度匹配结果表、用户推荐列表、用户历史搜索记录表等。这些表之间通过实体关系紧密相连，共同构成了完整的数据库结构，其E-R图如图3-2所示。

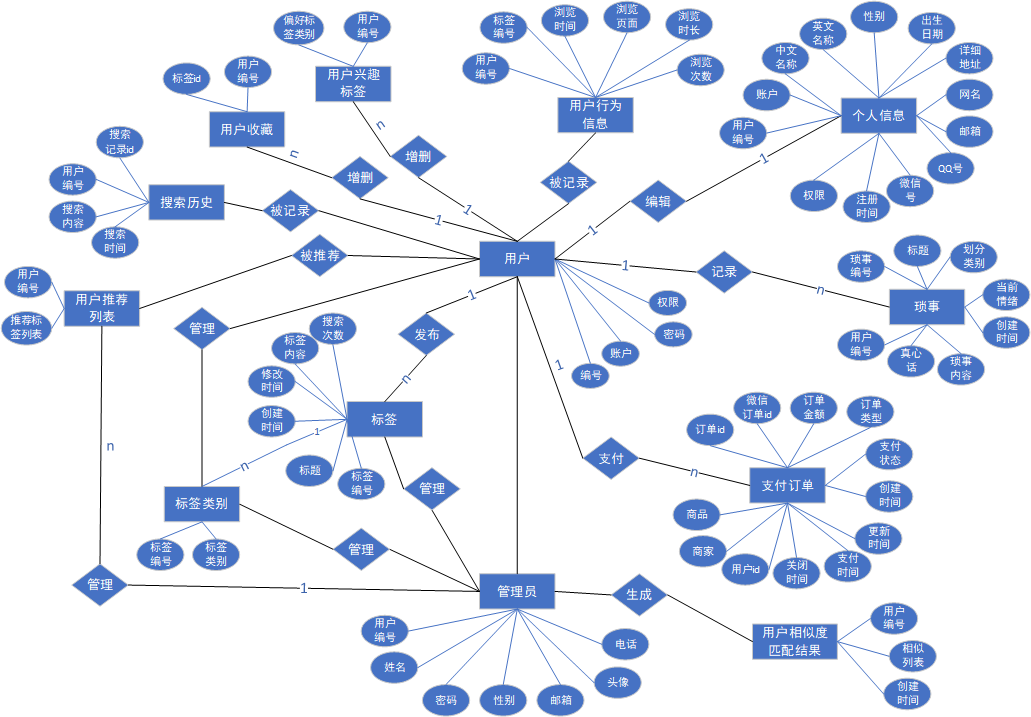


图3-2 系统E-R图

### 数据库表设计

根据以上数据库设计，设计出数据库表，该管理系统数据库主要表设计和数据库主要字段释意如下：

1. 用户表（user）用于存储本系统注册用户的基本信息。如表3-1所示：

表3-1 用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名称 | 含义解释 | 数据类型 | 数据长度 | 属性设置 |
| uid | 用户编号 | bigint |  | 非空,主键,自增 |
| username | 账户 | varchar | 11 | 必填,非空 |
| password | 密码 | varchar | 50 | 必填,非空 |
| power | 权限划分 | varchar | 2 | 非空，默认是B |

1. 用户个人信息表（userself）用于存储本系统已注册用户的的个人详细信息，初次注册后该表默认只有初始值，如表3-2所示：

表3-2 用户个人信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名称 | 含义解释 | 数据类型 | 数据长度 | 属性设置 |
| uid | 用户编号 | bigint |  | 必填,非空,主键,外键 |
| username | 账户 | varchar | 11 | 必填,非空 |
| cname | 中文名 | varchar | 200 |  |
| ename | 英文名 | varchar | 100 |  |
| birthday | 出生日期 | date |  |  |
| address | 详细地址 | varchar | 200 |  |
| petname | 昵称 | varchar | 200 |  |
| email | 邮箱 | varchar | 200 |  |
| qq | QQ号 | varchar | 11 |  |
| wechat | 微信号 | varchar | 200 |  |
| registertime | 注册时间 | datetime |  | 必填,非空 |
| power | 权限 | varchar | 2 | 必填,非空 |

1. 管理员表（admin）用于存储本系统中管理员的个人信息，如姓名、电话、头像等，用户表管理员表分开存储目的是为了提高系统安全性。如表3-3所示：

表3-3 管理员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名称 | 含义解释 | 数据类型 | 数据长度 | 属性设置 |
| admin\_id | 管理员编号 | int | 10 | 必填,非空,主键 |
| admin\_name | 姓名 | varchar | 20 | 必填,非空 |
| admin\_pwd | 密码 | varchar | 64 | 必填,非空 |
| adnin\_sex | 性别 | varchar | 2 |  |
| admin\_email | 邮箱 | varchar | 20 |  |
| admin\_pic | 头像 | varchar | 100 |  |
| admin\_phone | 电话 | varchar | 11 | 必填,非空 |

1. 标签信息表（labelcontent）保存本系统用户提交文章的相关信息，如表3-4所示：

表3-4 标签信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名称 | 含义解释 | 数据类型 | 数据长度 | 属性设置 |
| lcid | 标签编号 | int |  | 必填,非空,主键 |
| lctitle | 标签标题 | varchar | 500 | 必填,非空 |
| lctime | 创建时间 | timestap | 20 | 必填,非空,默认为  当前时间 |
| updatetime | 最新修改  时间 | timestap | 20 | 必填,非空,默认为  当前时间 |
| lccontent | 标签内容 | varchar | 10000 | 必填,非空 |
| searchnumber | 搜索次数 | double |  | 默认值为0 |
| uid | 用户编号 | bigint |  | 必填,非空,外键 |

1. 标签类别信息表（labelcategory）存储文章标签类别的相关信息，如表3-5所示：

表3-5 标签类别关系表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名称 | 含义解释 | 数据类型 | 数据长度 | 属性设置 |
| lcid | 标签编号 | int |  | 必填,非空,外键 |
| category | 标签类别 | varchar | 200 | 必填,非空 |

1. 琐事信息表（triviacontent）用于用户存储个人的日常生活及心情等相关信息，该表仅对用户开放，管理员不可直接访问，如表3-6所示：

表3-6 琐事信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名称 | 含义解释 | 数据类型 | 数据长度 | 属性设置 |
| tcid | 琐事编号 | int |  | 必填,非空,  主键,自增 |
| tctitle | 琐事标题 | varchar | 500 | 必填,非空 |
| usercategory | 用户自主划分分类 | varchar | 200 |  |
| emotion | 当前情绪 | varchar | 200 |  |
| tctime | 创建时间 | timestamp |  | 必填,非空 |
| tccontent | 琐事内容 | varchar | 10000 | 必填,非空 |
| truewords | 真心话/  琐事总结 | varchar | 10000 |  |
| uid | 用户编号 | bigint |  | 必填,非空,外键 |

1. 订单信息表（oreder）用于保存用户赞助本系统的支付订单相关信息，支付方式仅限微信、支付宝支付方式。如表3-7所示：

表3-7 订单信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名称 | 含义解释 | 数据类型 | 数据长度 | 属性设置 |
| orderid | 商家订单号 | varchar | 22 | 必填,非空,主键 |
| transactionid | 微信订单号 | varchar | 90 |  |
| payment | 订单金额 | decimal | 10,2 |  |
| paymenttype | 订单类型 | varchar | 3 |  |
| status | 支付状态 | varchar | 3 |  |
| createtime | 创建时间 | datetime |  |  |
| updatetime | 更新时间 | datetime |  |  |
| paymenttime | 支付时间 | datetime |  |  |
| closetime | 关闭时间 | datetime |  |  |
| uid | 用户ID | bigint |  | 必填,非空,外键 |
| seller | 商家 | varchar | 300 |  |
| goods | 商品 | varchar | 300 |  |

1. 用户收藏信息表（userfavorities），该表保存本系统中用户收藏文章的相关信息，如表3-8所示：

表3-8 用户收藏信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名称 | 含义解释 | 数据类型 | 数据长度 | 属性设置 |
| uid | 用户编号 | bigint |  | 必填,非空,外键 |
| ufid | 标签编号 | int |  | 必填,非空,外键 |

1. 用户行为信息表（browsingHistory），服务器会自动存储用户的浏览历史记录，便于后续对用户进行个性化推荐，如表3-9所示：

表3-9 用户行为信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名称 | 含义解释 | 数据类型 | 数据长度 | 属性设置 |
| uid | 用户id | bigint |  | 必填,非空,主键 |
| lcid | 标签id | int |  | 必填,非空,外键 |
| createtime | 浏览时间 | timestamp |  | 非空 |
| page | 浏览页面 | varchar | 100 |  |
| duration | 浏览时长 | bigint |  |  |
| num | 浏览次数 | int |  |  |

1. 用户相似度匹配结果表（userSimilarity），使用隐语义模型对用户行为信息进行分析，将相似用户信息存入该表中，如表3-10所示：

表3-10 用户相似度匹配结果表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名称 | 含义解释 | 数据类型 | 数据长度 | 属性设置 |
| uid | 用户id | bigint |  | 必填,非空,主键 |
| userlist | 相似用户列表 | varchar | 200 |  |

1. 用户推荐列表（recommend），用户相似度分析后，将相似用户浏览而被分析用户未浏览的标签推荐信息存入该表中，如表3-11所示：

表3-11 用户推荐列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名称 | 含义解释 | 数据类型 | 数据长度 | 属性设置 |
| uid | 用户id | bigint |  | 必填,非空,主键 |
| recommendlist | 推荐列表 | varchar | 300 | 必填,非空 |

1. 用户历史搜索记录表（searchHistory），统计用户输入的搜索内容，作为隐语义模型分析的一部分，如表3-12所示：

表3-12 用户历史搜索记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名称 | 含义解释 | 数据类型 | 数据长度 | 属性设置 |
| uid | 用户id | bigint |  | 必填,非空,主键 |
| recommendlist | 推荐列表 | varchar | 300 | 必填,非空 |
| createtime | 搜索时间 | timestamp |  | 必填，非空 |

# 详细设计

## 管理员模块实现

### 管理员登录

管理员首先进入到管理员登录页面，在输入正确的用户名和密码后向服务器发送登录请求，通过服务器验证后，获取到唯一的token密钥返回给管理员页面，实现管理员登陆。登陆后跳转至管理员主页。如图4-1所示。



图4-1 管理员登录界面

### 后台管理员主页

管理员登陆本系统后台后，进入本系统系统后台导航页面，该页面左侧是用户管理等功能模块菜单栏，管理员通过点击不同菜单栏切换到不同功能模块页面。如图4-2所示：

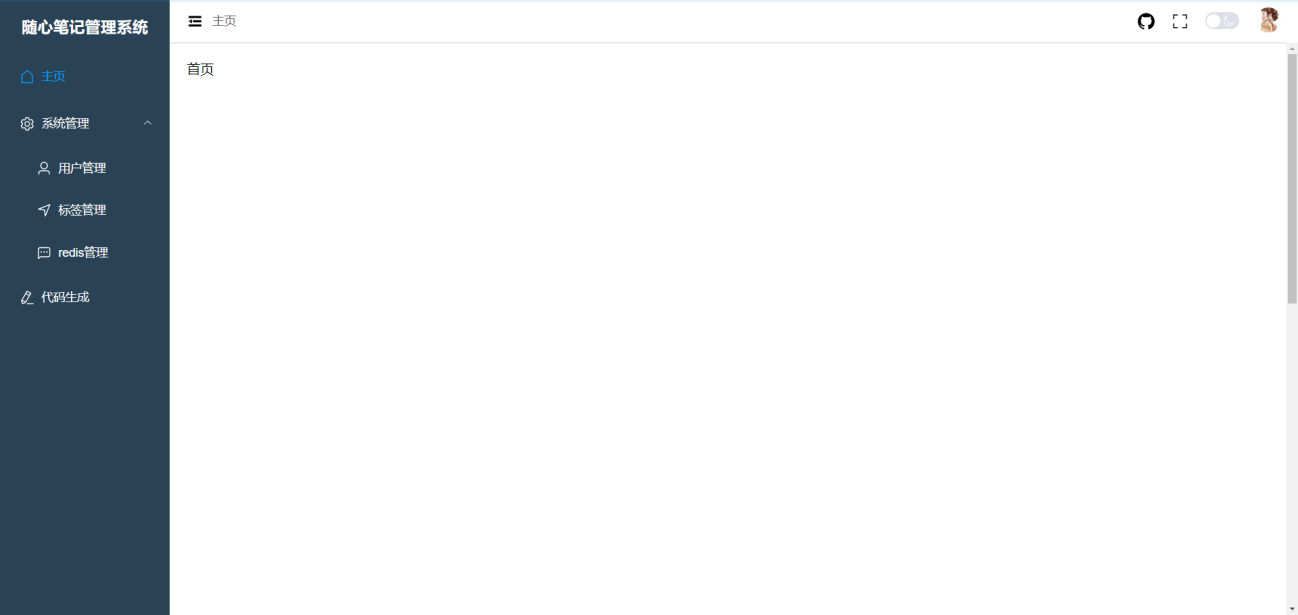


图4-2 管理系统后台首页

### 用户管理

点击管理员左侧的用户管理，页面将切换到用户管理界面，该界面展示了已经在本系统注册用户的基本信息，如昵称，账户，用户id等，管理员可对用户信息进行增删改查操作，也可修改用户权限。如图4-3、图4-4、图4-5所示：

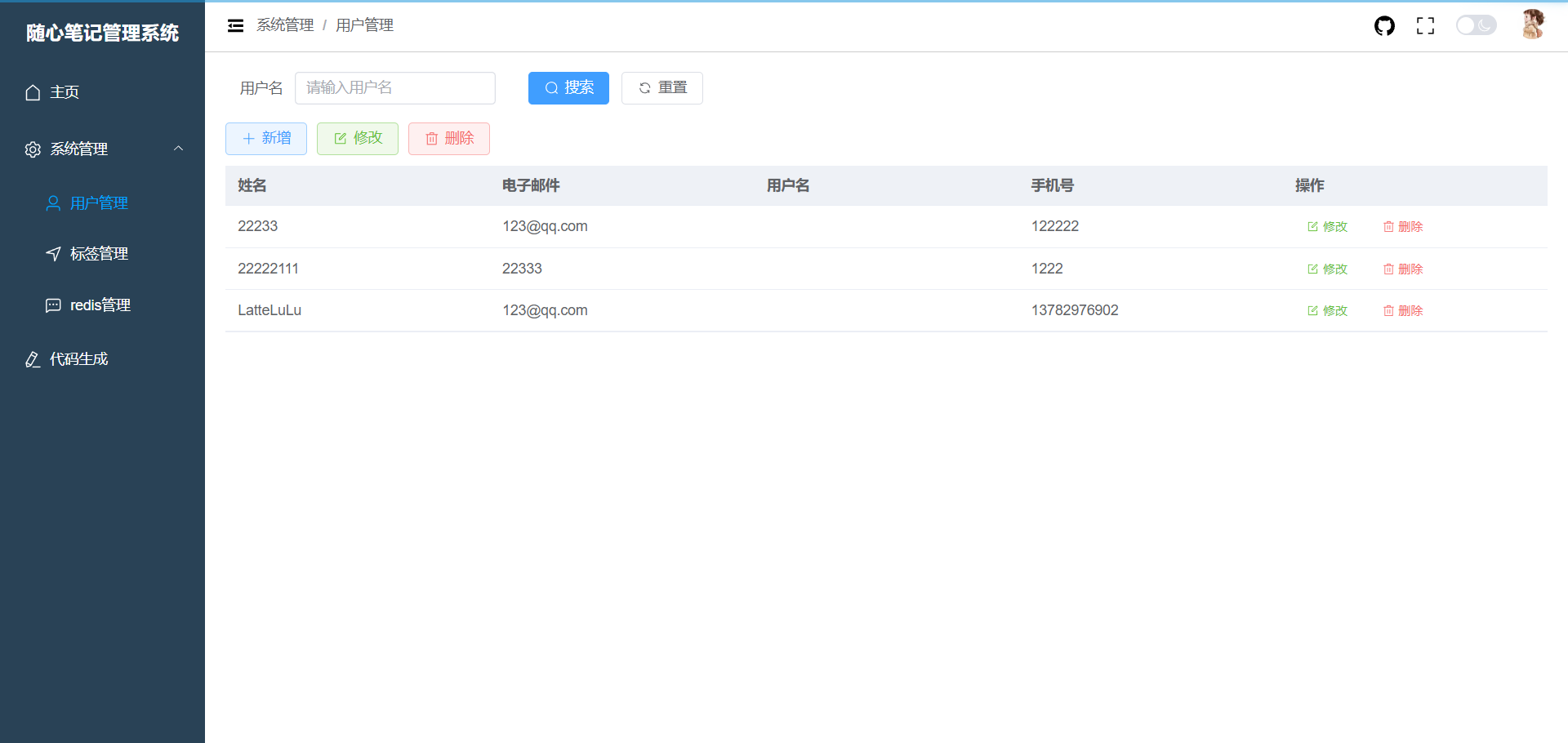


图4-3 后台用户管理界面



图4-4 修改用户信息或权限



图4-5 根据用户名查询

### 标签管理

管理员点击左侧菜单项的标签管理进入标签管理页面，在该页面中，管理员可查阅所有标签信息，对于一些错误添加的标签文章进行删除。如图4-6、图4-7所示：

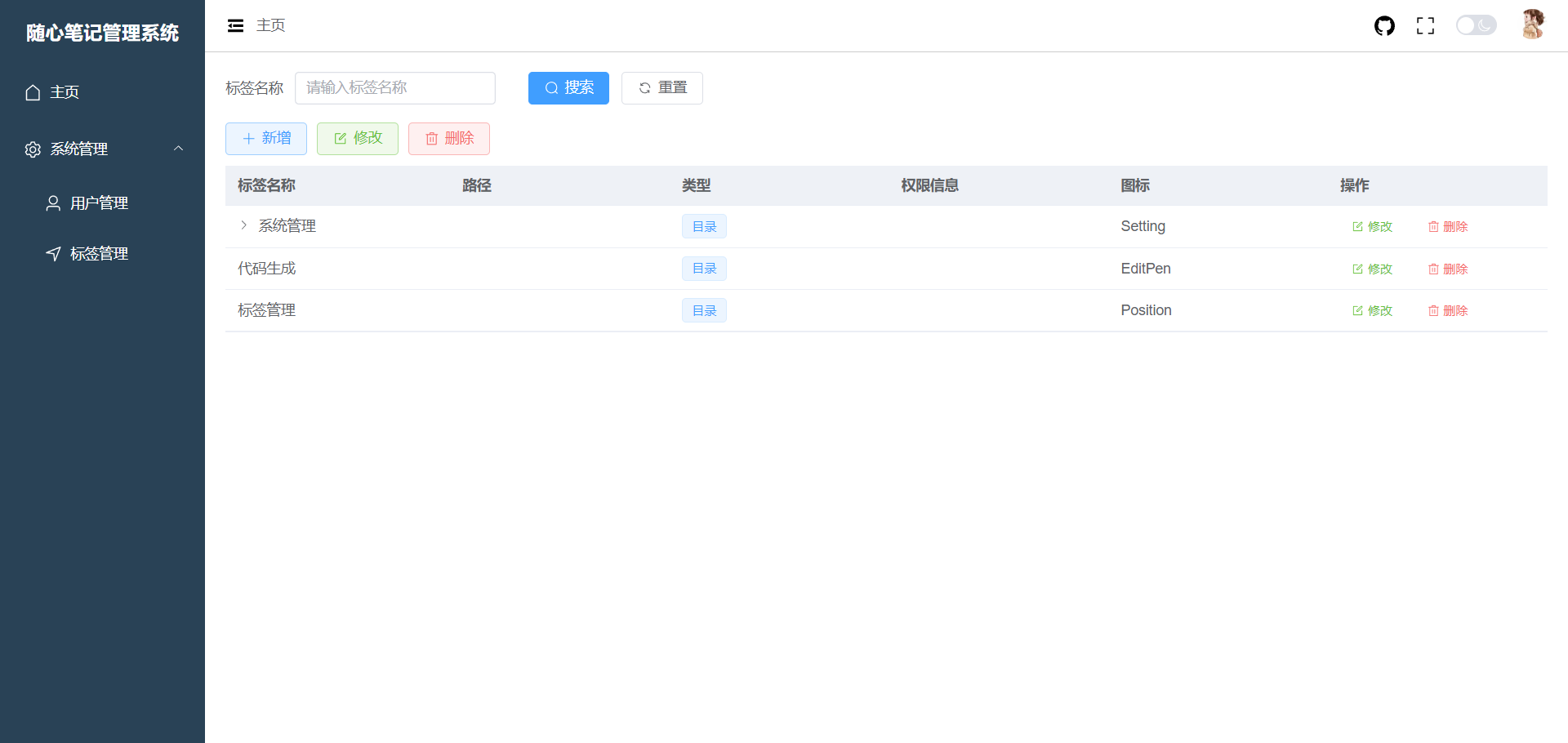


图4-6 标签管理界面

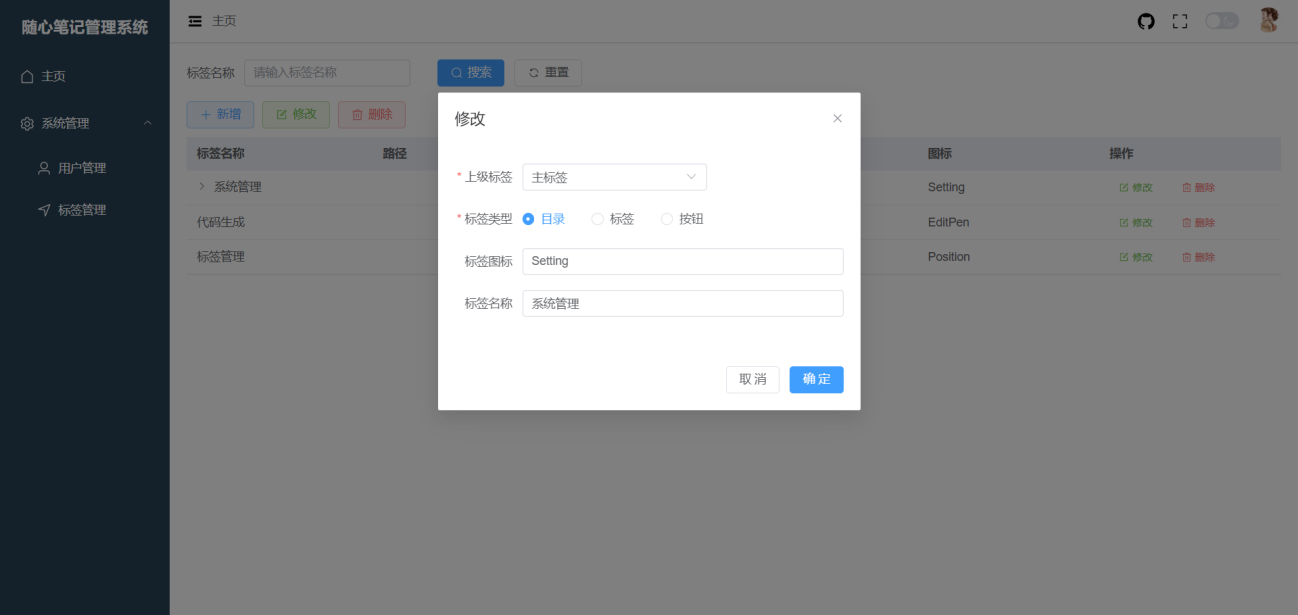


图4-7 修改标签信息界面

### Redis管理

管理员点击左侧菜单栏的的Redis管理进入redis操作界面，在该页面中设置有三个按钮：重启redis、清空redis1缓存，redis与MySql数据统一。管理员点击相应按钮，服务器执行相关操作。如图4-8、图4-9所示。

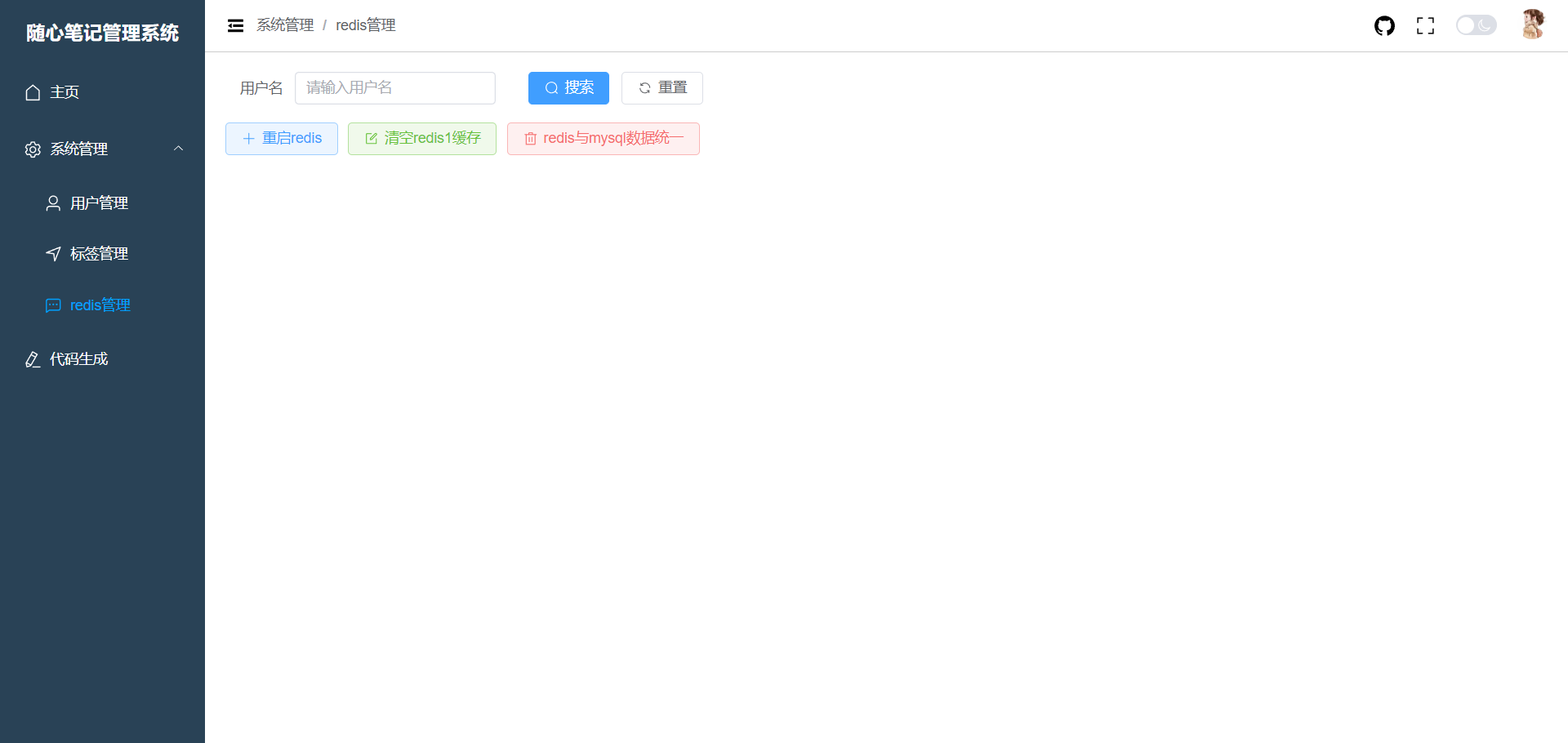


图4-8 redis管理页面



图4-9 redis与MySql数据一致性功能部分代码

### 隐语义模型

管理员切换到pyCharm软件中，打开lfm.py文件，点击Run启动隐语义模型代码，代码会自动读取MySql数据库中的用户离线数据，作为训练集，通过矩阵分解+梯度下降方法，分析用户数据并建立用户之间的相似度匹配，最后根据用户之间的相似度对每个用户进行个性化预测，预测结果即用户推荐列表存入MySql中，在下次用户请求服务器响应时，向用户展示预测后的个性化信息。如图4-10、图4-11所示。

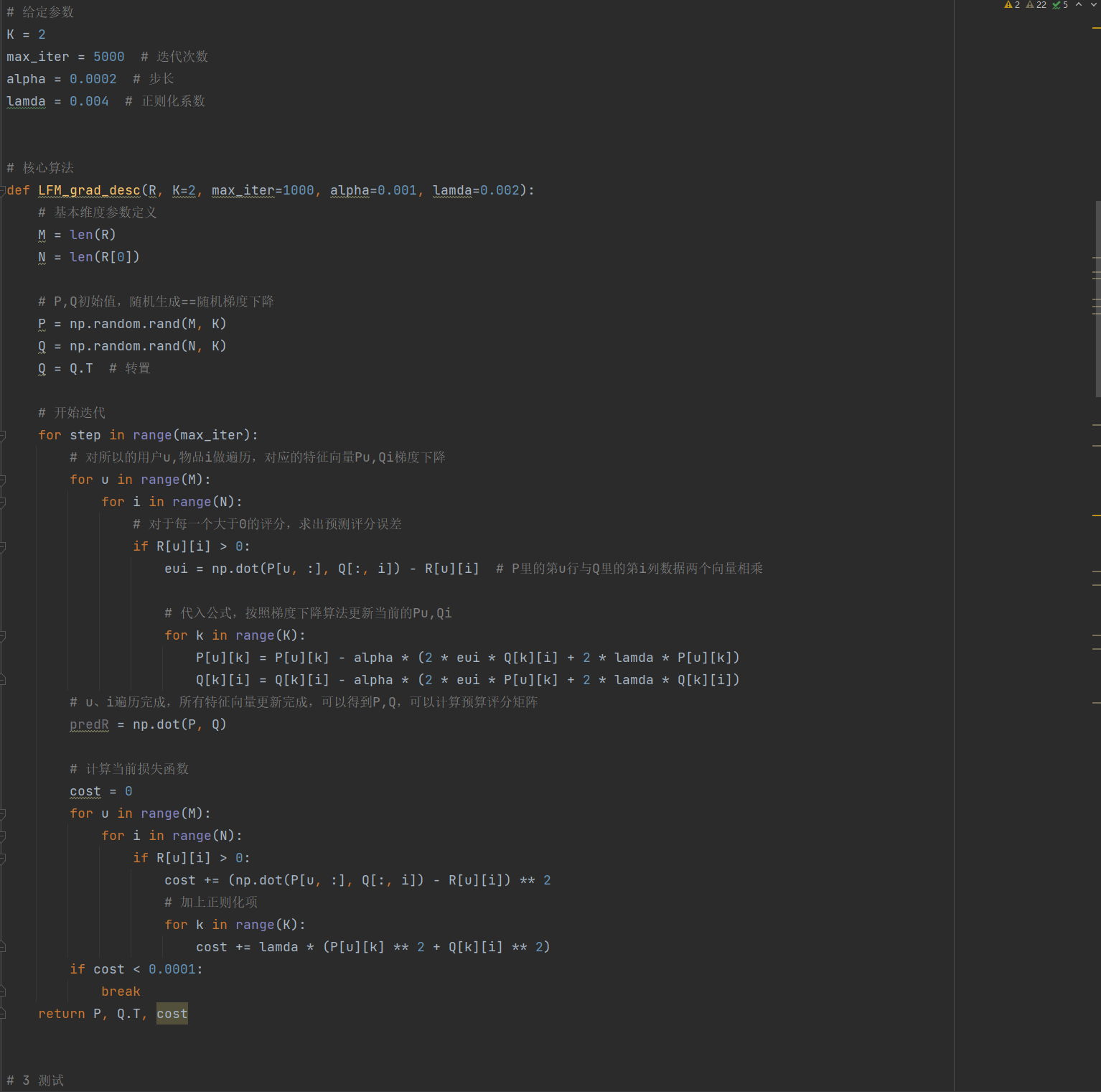


图4-10 隐语义模型代码

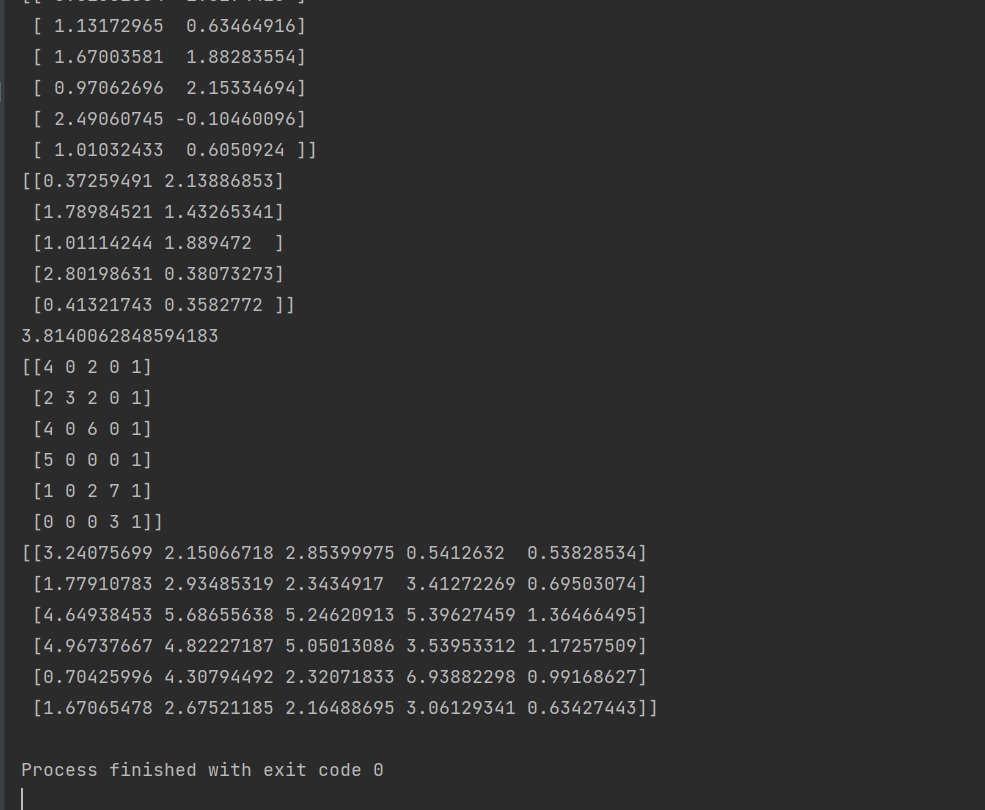


图4-11 测试用例结果

## 用户模块实现

### 网站前台首页

用户初次访问本系统时，即可看到本系统的首页。其中包含了用户模块的相关功能导航栏，当地天气展示，主机所在地区的百度地图信息等。如图4-12所示：



图4-12 用户前台首页

### 用户注册登录

用户只有注册后才能登录本系统，注册时需输入手机号、密码等相关信息，注册采用手机号+短信验证码的方式，安全可靠。用户登陆后，会返回到首页，接下来用户即可浏览或使用用户模块相关功能。如图4-13、图4-14所示。



图4-13 用户前台注册页面



图4-14 用户前台登录页面

### 热搜榜页面

用户点击导航栏的热搜榜后，会跳转到热搜榜页面，该页面有搜索框，支持标签类别搜索和标签标题内容搜索，搜索框下面是热门的搜索内容，页面中间是当前搜索次数最高的5个标签文章，点击GO按钮可跳转到对应的标签文章内容页面。如图4-15所示：



图4-15 热搜榜页面

### 实时论坛页面

用户点击导航栏的论坛按钮后，会跳转到论坛页面，该页面按找标签文章的创建时间降序展示，会显示最新添加的标签文章内容，用户也可点击文章标题跳转到对应的标签文章内容页面进行查看。如图4-16所示：



图4-16 实时论坛页面

### 标签和琐事创建页面

用户在网络其他地方浏览完感兴趣的信息后，可在标签创建页面添加相关内容的标签，在记录信息的同时，可以给标签标记类别，便于个人查阅和其他用户搜索并浏览，这些标签创建后可以展示在个人信息的多彩生活页面，同时可被其他用户搜素并浏览。

用户在琐事创建页面可以记录自己的个人生活，内容记录生活中的琐事，最底下可以标注当前的心情以及对于这些事想发表的一些看法。有助于用户养成记录生活和写日记的习惯，提高用户生活幸福感。如图4-17，图4-18所示：



图4-17 标签创建页面



图4-18 琐事创建页面

### 个人信息和用户收藏页面

用户在注册后可在个人信息页面完善自己的详细信息，以便下次登录更好的展示用户的个性化内容。

用户可以在个人信息页面的海纳百川页面查看用户个人收藏的标签，用户可在实时论坛和热搜榜页面查阅标签时点击收藏星星按钮添加标签至个人收藏中，再次点击可取消收藏。如图4-19，图4-20所示：



图4-19 用户个人信息页面



图4-20 用户标签收藏页面

## 隐语义模型

### 关于隐语义模型

协同过滤算法有很多，隐语义模型（LFM）便是其中之一。LFM（latent factor model）隐语义模型包括许多著名的模型和方法。其实该算法最早应用于文本挖掘领域，用于找到文本的隐含语义。相关的算法有LSI、Topic Model等等。显性反馈和隐性反馈是推荐系统的用户行为。LFM更擅长在显性反馈数据上解决评分预测问题，并且在优化后能够达到了很好的精度。

隐语义模型揭示了用户之间的隐藏特征,个人是无法对这些隐藏特征进行精确分类的，所以隐藏特征也无法直接用语言解释描述的，而这便是隐语义模型的精妙之处。

隐语义模型通过矩阵分解进行降维分析，隐藏特征存在于用户和物品之间，即分解矩阵中。

### 隐语义模型与其他算法比较

隐语义与ItemCF或UserCF的不同。对于UserCF，可以先计算和目标用户兴趣相似的用户，根据相似用户喜欢的的物品，选择目标用户可能喜欢的物品给予推荐。而ItemCF，则以目标用户喜欢的物品为线索，根据该物品特点来筛选相似物品，最后将筛选出来的相似物品推荐给用户。对于隐语义模型来说，先读取用户的历史行为信息，形成用户-物品矩阵，进行矩阵分解，梯度算法求出最优值后，筛选出目标用户最可能喜欢的N个物品，再推荐给用户，LFM就是用来实现这种方法。

从分类角度看，在对某一事物的分类过程中，很容易想到人工对物品进行分类，但是人工分类是一种很主观的事情，不同的人可以得到不同的分类。而且对于物品分类的粒度很难控制，究竟需要把物品细分到个程度，不同专业领域的人对同一事物也有不同的分类标准，想要对物品进行小粒度细分无疑是非常困难的事情。而解决这个问题，就需要隐语义模型。

### 隐语义模型计算公式

根据隐语义模型的特点，计算用户u对物品i兴趣的公式如图4-21所示：

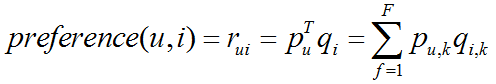


图4-21 隐语义模型用户对物品兴趣公式

其中Puk度量了用户u的兴趣和第k个隐类的关系，而Qik度量了第k个隐类和物品i之间的关系。该公式的简单逻辑思路是：数据集中包括用户喜欢的物品和不喜欢的物品，根据这个数据集来计算p和q。如果已有的数据集没有负样本，则从每个用户没有过行为的物品中采样出与正样本数量基本相同的负样本作为该用户的负样本，以上操作完成后，即可进行隐语义模型训练。

### 模型测试样例

下面给出公式，对于正样本，首先规定r=1，负样本r=0，这样可以得到如下损失函数来找到最合适的参数p和参数q。在下图中的损失函数里边，有两组参数puk和qik，采用随机梯度下降法，需要对损失函数分别求偏导数，如图4-22，4-23：

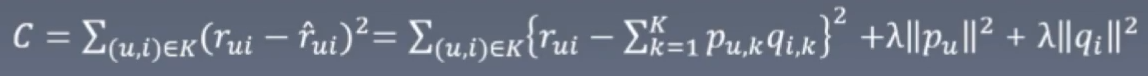


图4-22 损失函数

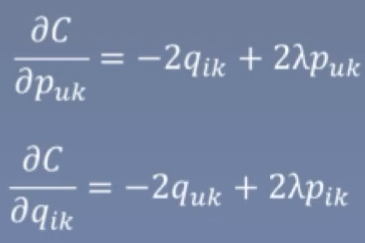


图4-23 梯度下降算法求偏导

然后，根据随机梯度下降法，使得曲线函数沿最快下降方向进行。因此可以得到如图4-24递推公式：

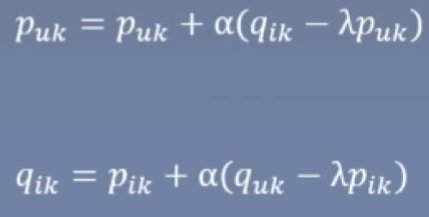


图4-24 梯度下降递推公式

其中α是学习速率，该值的选取需要通过反复试验获得。后面的λ是为了防止过拟合的正则化项。

# 系统测试

## 系统功能测试结果

为了更加全面地验证每个模块的功能实现，本系统采用将使用黑盒测试检验各项功能是否可以达到预计效果。在测试用例的设计方面，本系统采用了不同情况划分的方式，以确保测试的覆盖性和效率。通过这种方法，能够有效地识别出各个功能模块潜在的问题，并及时修复，从而确保系统的稳定性和代码的健壮性。

### 管理员及用户登录

管理员进入本系统后台登录页面，这里将对登录模块功能进行测试。如表5-1、表5-2、表5-3所示：

表5-1 登录描述

|  |  |  |
| --- | --- | --- |
| 功能名称 | 功能描述 | 测试方法 |
| 用户和后台管理员登录 | 验证登录信息 | 黑盒测试 |

表5-2 登录不同情况划分

|  |  |  |
| --- | --- | --- |
| 测试序号 | 登录情况 | 内容设置 |
| 1 | 用户名正确 | 正确用户名 |
| 2 | 密码输入正确 | 正确密码 |
| 3 | 用户名有误 | 错误用户名 |
| 4 | 密码输入有误 | 错误密码 |

表5-3 测试用例设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试数据 | 17837435290，123456 | 17838435290，  11111111 | 15845686978，123456 | 18359745628，958412316 |
| 预期结果 | 登录成功 | 用户名或密码错误 | 用户名不存在 | 用户名不存在 |
| 覆盖范围 | 1、2 | 1、4 | 2、3 | 3、4 |
| 是否通过 | 🗸 | 🗸 | 🗸 | 🗸 |

### 标签管理

管理员对已有标签管理功能测试如表5-4所示：

表5-4 标签功能测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试步骤 | 测试内容 | 测试结果 | 功能简介 |
| 1 | 输入姓名、密码登录 | 🗸 | 是否能查看到已有的标签内容和标签类别，进行标签的添加，修改，删除 |
| 2 | 进入后台导航页 | 🗸 |
| 3 | 进入标签管理页 | 🗸 |
| 4 | 增删改查标签信息 | 🗸 |

### 琐事管理

用户自主编写日常生活，管理功能测试如表5-5所示：

表5-5 琐事管理功能测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试步骤 | 测试内容 | 测试结果 | 功能简介 |
| 1 | 输入姓名、密码登录 | 🗸 | 正常查看用户列表，进行用户修改添加，删除用户 |
| 2 | 进入后台导航页 | 🗸 |
| 3 | 进入用户管理页 | 🗸 |
| 4 | 增删改查用户信息 | 🗸 |

### 用户收藏管理

用户点击收藏按钮，将对应文章添加到用户的收藏列表测试如表5-6所示：

表5-6 用户收藏功能测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试步骤 | 测试内容 | 测试结果 | 功能简介 |
| 1 | 输入姓名、密码登录 | 🗸 | 标签收藏  取消收藏  查看收藏 |
| 2 | 进入个人信息页面 | 🗸 |
| 3 | 进入海纳百川页面 | 🗸 |
| 4 | 新增、浏览、删除收藏标签 | 🗸 |

### 热搜榜管理

除了服务器定时自动更新热搜榜，管理员可以手动更新热搜榜内容，这里时热搜榜手动更新功能测试如表5-7所示：

表5-7 热搜榜功能测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试步骤 | 测试内容 | 测试结果 | 功能简介 |
| 1 | 输入姓名、密码登录 | 🗸 | 清除已有热搜榜，  重新生成新的热搜榜内容 |
| 2 | 进入热搜榜管理页面 | 🗸 |
| 3 | 点击更新热搜榜按钮 | 🗸 |
| 4 | 查看新生成的热搜榜 | 🗸 |

### Redis管理

为了确保信息一致性，redis会定时将redis缓存与MySql进行同步，这里测试同步功能是否能正常运行功能测试如表5-8所示：

表5-8 redis功能测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试步骤 | 测试内容 | 测试结果 | 功能简介 |
| 1 | 输入姓名、密码登录 | 🗸 | 是否使redis缓存与MySql保持一致 |
| 2 | 进入redis功能页面 | 🗸 |
| 3 | 点击redis与MySql  同步按钮 | 🗸 |
| 4 | 同步redis与MySql数据 | 🗸 |

### 隐语义模型预测

管理员启动隐语义模型代码，python读取MySql离线数据，同时启动隐语义模型对离线数据进行分析并生成预测结果，结果存储在MySql数据库中，这里是对隐语义模型的功能测试，如表5-9所示：

表5-9 隐语义模型预测功能测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试步骤 | 测试内容 | 测试结果 | 功能简介 |
| 1 | 管理员打开lfm.py代码 | 🗸 | 是否能正常启动隐语义模型，  并生成预测结果 |
| 2 | 点击Run后启动改代码 | 🗸 |
| 3 | 查看控制台运行结果 | 🗸 |
| 4 | 查看MySql更新后数据库 | 🗸 |

## 系统测试结论

在所有黑盒测试完成后，管理系统的前后台所有功能已全部得到验证，并且达到预期效果，测试成功。

# 结　论

本系统使用SpringBoot轻量级架构编写服务器端，将redis作为MySql与服务器之间的缓存数据库，前端使用Vue搭配MyBatis-Plus、Echarts、Element-UI、Axios等技术构建用户界面，总体上实现了用户对自己编写的标签、琐事的管理，对热搜榜内容和实时论坛内容的浏览，对个人信息和个人标签收藏的统筹，管理员对用户模块、标签模块、redis相关功能的管理，对隐语义模型的启动，生成预测结果，并达到了预期的效果。但仍存在一些小问题，比如：页面不够简洁美观，对redis中间件的使用不熟练，代码不够简练，在使用MyBatis-Plus时，数据库操作接口方法使用不熟练，使用较多原生MyBatis编写。在隐语义模型算法逻辑的设计不够合理，推荐效果需要提高。

本次毕业设计过程中遇到的问题将会是以后系统改进之处，在之后的开发过程中，不仅要扩大隐语义模型的训练集，还要优化推荐算法，加入更多的影响因子，提高个性化推荐服务效果。

参考文献

1. 范伊红,黄彩霞.基于HTML5的网页设计及应用.北京:电子工业出版社,2014
2. 邵阳阳,徐子良,姜玉波,等.基于LFM算法的SpringCloud分布式购物系统[J].聊城大学学报(自然科学版),2023,36(06):27-34.
3. 周熙然. 基于隐语义模型的推荐算法研究与应用[D].西安科技大学,2022.
4. 樊艳清,纪佳琪.基于深度学习的隐语义协同过滤推荐模型研究[J].现代计算机,2023,29(18):18-23.
5. 徐吉,李小波,陈华辉,等.基于协同过滤和隐语义模型的混合推荐算法[J].计算机技术与发展,2020,30(02):52-57.
6. 汪佩,梁立,甘健侯.基于用户兴趣变化的隐语义协同过滤算法[J].云南师范大学学报(自然科学版),2017,37(04):39-43.
7. 王德贤,何先波,贺春林,等.结合L1和L2正则化约束的隐语义预测模型研究[J].计算机工程与应用,2019,55(19):121-127.
8. 胡堰,彭启民,胡晓惠.一种基于隐语义概率模型的个性化Web服务推荐方法[J].计算机研究与发展,2014,51(08):1781-1793.
9. 巨星海,周刚.一种基于用户偏好分析和论坛相似度计算的改进LFM推荐算法[J].信息工程大学学报,2021,22(04):433-437+449.
10. 孔欢. 基于隐语义模型推荐算法研究[D]. 江苏:江苏科技大学,2020.
11. 王蓉,李晗,周国海,等. 基于SpringCloud框架的医疗信息共享平台设计与实现[J]. 中国医学装备,2022,19(5):133-137.
12. 史加荣,王丹,尚凡华,等. 随机梯度下降算法研究进展[J]. 自动化学报,2021,47(9):2103-2119.
13. 刘旋. 基于谱聚类和LFM的选课推荐算法设计[J]. 现代信息科技,2020,4(1):14-16.
14. 孙海峰,甘明鑫,刘鑫,等. 国外电影推荐系统网站研究与评述[J]. 计算机应用,2013,33(z2):119-124.
15. 陈天昊,帅建梅,朱明. 一种基于协作过滤的电影推荐方法[J]. 计算机工程,2014(1):55-58,62.
16. 巢伟. A Movie Recommendation System Based on Hybrid Recommendation Algorithm[D]. 湖北:华中师范大学,2020.
17. 李豫承. 推荐系统小批量内负样本采样和利用研究[D]. 安徽:中国科学技术大学,2023.
18. 孙洪盼. 基于SpringBoot和Vue的友为交流社区的设计与实现[D].重庆大学,2022.
19. 王明浩. 基于SpringCloud的社区电商系统的设计与实现[D]. 湖北:华中科技大学,2021.
20. 宋云奎,吴文鹏,赵磊,等.基于Redis的分布式数据存储方法[J].计算机产品与流通,2020(08):106.

致　谢

时光荏苒，大学四年已经到来了尽头，即将登上社会的大船。回首这段宝贵的时光，每一次课堂的听讲、每一次实验项目的探索，都成为了我心中难以忘怀的美好回忆。那些调试成功后的喜悦与自豪，以及遇到Bug时的迷茫与挣扎，都构成了我成长的足迹。值此毕业论文完成之际，我真心感谢黄彩霞老师。她细致入微的指导、宝贵的建议，对我的论文完成起到了举足轻重的作用。黄老师严谨的治学态度、认真负责的工作作风，始终激励着我不断前行，不敢有丝毫的懈怠。正是在她的悉心督促下，我得以顺利完成毕业设计，对此我深感感激。每当我看到黄老师那几缕白发，心中便涌起一股敬意与感激。那是她为我们学业和未来规划所付出的辛劳与心血的见证。每次与黄老师交流，我都能感受到她不仅是我们的良师，更是我们的益友。她总是耐心倾听、认真解答，让我感受到了师生之间的深厚情谊。黄老师的治学态度和工作作风，将成为我今后工作和生活中的宝贵财富。我会将她的精神继承发扬下去，努力在自己的领域里追求卓越，不辜负老师的期望与付出。

此外，我还要衷心感谢软件学院的各位老师们。从毕业设计选题确定，到导师的分配，再到毕业设计成果的验收，以及毕业论文的提交与答辩安排，这半年来老师们付出了大量的心血和辛劳。作为河科大的学子，作为软件学院的一员，我深感荣幸与自豪。老师们的专业指导与无私奉献，让我在学术道路上不断前行，收获了宝贵的知识与经验。在此，我向各位老师表示最诚挚的谢意！在毕业设计的过程中，我也曾遇到了一些困难和挑战。然而，幸运的是，我得到了刘如冉同学的热情帮助和支持。她不仅与我共同探讨问题，还提供了宝贵的建议和意见，使我能够按时提交毕业设计项目。在此，我要向她表示衷心的感谢！

回顾这段独自进行毕业设计的大四时光，我感慨良多。那些为了一个目标而努力的日子，将成为我人生中难以忘怀的记忆。考研的历程，与毕业设计的并行战斗，都让我更加坚定了自己的信念和追求。不求完美，只求更好，这种不断进取的精神将伴随我走向未来。最后，我要感谢一下自己。二十多岁的年纪，正是为梦想而努力奋斗的时光。无论自信还是自以为是，这都是我成长的印记。我相信，在这个年纪，我们有勇气去尝试、去挑战，即使失败了，那也是青春的印记；如果成功了，那便是成长的见证。毕业之际，我要对自己说：人生就像射箭，梦想就像箭靶，学习只是过程，有目标每日拉弓才有意义。

英文资料翻译

英文原文：

ASurveyofCollaborative Filtering Techniques

1. Introduction

This paper introduces the characteristics and challenges of collaborative filtering, memory-based CF, Model-based CF, Hybrid CF, and evaluation indicators.

CF challenges: highly discrete data, exponential growth in the number of users and items (explosive growth of the user-item matrix), need to complete recommendations in a very short time, synonym processing, data noise, and more

Memory-base CF: Calculates the similarity between users or items based on user ratings and makes predictions and recommendations.

Model-base CF: Model-based CF typically uses machine learning algorithms to build predictive models, such as matrix decomposition, neural networks, decision trees, etc. These algorithms can predict how new users will rate new items by learning patterns and relationships in the rating data. Model-based approaches typically have better predictive performance and scalability than memory-based approaches, but require more computational resources and time to train the model.

In addition to collaborative filtering, content-based recommendation methods are also very important. This paper briefly introduces the similarities and differences between collaborative filter and content-base recommender

Collaborative filtering: Based on historical user behavior data, it will analyze the user's rating or behavior data on the item, and then find similar users, will recommend to their favorite items.

Content filtering based recommendation system: It is based on the characteristics of the item itself to recommend, it will analyze the attributes and characteristics of the item, and then recommend items similar to the historical behavior of the user to them.

Both of these methods have their own advantages and disadvantages in the recommendation system. Collaborative filtering can make recommendations without item data, but it requires sufficient user rating data. Content-based recommendation systems can make recommendations when there is not enough user rating data, but need accurate item metadata. Therefore, in practical applications, the two methods are usually used in combination to achieve better results.

2. Characteristics and Challenges of Collaborative Filtering

The recommendation system provides users with fast and accurate content, which attracts users to use the platform and translates into benefits such as the purchase rate of goods. The proposed recommendation algorithm has the following characteristics and challenges:

2.1 Data sparsity

There is usually a large amount of content and users in the system, so the user-item matrix is very large and very sparse. Sparse data matrices can lead to problems like finding users or items that are similar enough to recommend:

Cold start problem: In some literatures, it is also called new user problem or new item problem. When a new user or new item content is added to the system, it is impossible to find similar data due to insufficient data on the new content, so it is impossible to recommend the new content.

Low coverage problem: reduced coverage When the user's rating data is very small and the number of recommended content is very large, the reduced coverage rate problem will occur. The recommendation system may not be able to generate recommendations for these users.

Neighbor passing problem: In the recommendation system, neighbor passing refers to when there is no common rating between two users, even if they have similar tastes, they cannot be identified as similar users. This can reduce the effectiveness of recommendation systems that generate recommendations based on predictions from comparing user pairs.

Neighbor passing problems are often caused by a lack of data in sparse databases. To solve this problem, more sophisticated algorithms can be used to deal with it, such as item-based recommendation systems, which use the attributes and characteristics of items to make recommendations, rather than relying solely on comparisons between users.

solution

Reduce the latitude of the matrix: remove unimportant user or item information by mathematical means such as singular value decomposition and principal component analysis. Discarded information may result in the loss of information related to the recommendation, reducing the quality of the recommendation

Hybrid CF-based cold start solution: The content-enhanced CF algorithm uses external content information to help solve cold start problems. Or by combining the cooperation information and the content information, the recommended method of cold start is proposed. Or divide the recommended content into different groups and predict the recommended content based on the Gaussian distribution of user ratings.

Model-based CF:TAN-ELR is a classifier based on Bayesian networks and logistic regression to improve the accuracy and efficiency of the Model through tree enhancement and extended logistic regression.

Collaborative filtering algorithm based on matrix decomposition: the user-item score matrix is decomposed into user feature matrix and item feature matrix, wherein the user feature matrix contains the hidden feature vector of the user, and the item feature matrix contains the hidden feature vector of the item. By calculating the inner product of the user feature vector and the item feature vector, we can get the predicted value of the user's score on the item. Common ones are SVD, NMF, PMF, MMMF. MMMF maximizes the marginal spacing between different classes so that the classifier has stronger generalization.

Multiple interpolation CF method and interpolation enhanced CF algorithm use interpolation method to fill in the missing data, so as to improve the performance of recommendation system.

2.2 Scalability

When the number of existing users and projects grows dramatically, traditional CF algorithms will suffer from severe scalability problems. For example, with tens of millions of customers (M) and millions of different directory entries (N), a CF algorithm with O (n) complexity is too large. In addition, the recommendation system needs to be extremely real-time and provide recommendation data to all users, regardless of their behavioral data history, which requires high scalability of the recommendation system.

Model-based Person correlation CF: Item-based person correlation CF: does not calculate the similarity between all item pairs, but only calculates the similarity between item pairs evaluated by users. The simple Bayes CF algorithm makes prediction based on the observed user rating behavior. Clustering CF algorithm: Solves scalability problems by seeking user recommendations in smaller and highly similar clusters rather than across the entire database;

2.3 Synonymy

Synonyms refer to many identical or very similar items with different names or entries. The recommendation system should find associations between word meanings and be able to recommend similar content. This section deals with NLP.

3. Memory-Based Collaborative Filtering Techniques

Memory-based CF algorithms use the whole or an example of a user-item database to generate predictions. Each user is part of a group of people with similar interests. By calculating a new user's (or active user's) neighbors, a prediction of his or her preferences for a new project can be generated.

The neighborhood-based CF algorithm is a collaborative filtering algorithm based on neighborhood, which is used for personalized recommendation in recommendation system. In this algorithm, the similarity between the user and the item is determined by calculating the distance or similarity between them. Then, based on the user's previous ratings and ratings of similar items, recommend the most relevant items for the user. This algorithm is often used for large data sets because it does not need to calculate the similarity between all users and items, but only a small group of neighbors that are most similar to the target user or item. Therefore, it can improve the efficiency and accuracy of the recommendation system.

In practical applications, techniques such as cosine similarity or Pearson correlation coefficient are often used to calculate the similarity between users and items. In addition, the KNN (K-Nearest Neighbor) algorithm can also be used to determine the Nearest Neighbor that is most similar to the target user or item.

3.1 Similarity Computation

3.1.1 Correlation Based Similarity

Based on Pearson correlation:

The Pearson correlation coefficient is a statistic that measures the strength of the linear relationship between two variables. It is used to measure the correlation between two continuous variables, with values ranging between -1 and 1, where 0 means no linear correlation, positive values mean positive correlation, and negative values mean negative correlation. The Pearson correlation coefficient can be used to measure the linear relationship between two variables, but it does not indicate whether there are other types of relationships between them. The Pearson correlation coefficient is commonly used to explore the relationship between two variables, for example in data mining, machine learning, and statistical analysis. It can help researchers determine if there is a correlation between two variables and help select the most relevant variables when building predictive models. However, it should be noted that Pearson correlation coefficient can only measure linear relationships, and is not applicable to nonlinear relationships.

The expansion algorithms of Pearson correlation coefficient include: constrained Pearson correlation and Spearman rank correlation

The latter two: Both are calculated by comparing the rank order between two variables rather than directly comparing their numerical values. Used to measure the monotonic relationship between two variables, i.e. whether they vary together, but cannot say whether there are other types of relationships between them. They are often used to explore the relationship between two variables without the need to assume that the data is normally distributed or linearly correlated.

3.1.2. Vector Cosine-Based Similarity.

Adjusted cosine similarity: The vector cosine similarity calculates the Angle between two items, while the adjusted cosine similarity takes into account the use of different scoring scales between different users. Since different users may use different scoring scales, the vector cosine similarity cannot account for this difference. To solve this problem, the adjusted cosine similarity can be used. The method subtracts the average rating of each user from that user's rating, and then calculates the cosine similarity between items. The adjusted cosine similarity has the same formula as the Pearson correlation coefficient, but it standardizes users' ratings to correct for differences in rating scales.

3.2 Prediction and Recommendation Computation

In the neighborhood method, the prediction process mainly consists of two steps: selecting neighbors and calculating prediction scores. With these two steps, you can generate personalized recommendation results for the current user.

3.2.1 Weighted Sum of Others' Ratings 3.2.1 Weighted sum of others' ratings

You first need to find the group of users that are most similar to the current user, called neighbors. Then, each neighbor's predicted score for items not rated by the current user is calculated.

Specifically, for each neighbor, the similarity between its rated items and the items not rated by the current user can be used to calculate its influence weight on the items not rated by the current user. The influence weight of each neighbor is then multiplied by its rating of the item unrated by the current user to obtain the weighted score of the item unrated by the current user for each neighbor. Finally, the weighted scores of all neighbors are added together to obtain the current user's predicted score for the unrated item. Compared with simple average or weighted average method, this method reflects the differences among neighbors more accurately. At the same time, because the similarity between neighbors is used to calculate the weight, the differences between neighbors can be better handled.

3.2.2. Simple Weighted Average. Simple weighted average

For item-based predictions, user u rates i as follows:

Where r{u,n},n∈N is user n's rating of other items n, and w{i,n} is the similarity of items i and n.

3.3. Top-N Recommendations

3.3.1 User-Based Top-N Recommendation Algorithms

User-based Top-N recommendation algorithm is a technique that analyzes user behavior data to recommend N items that are most likely to interest a particular user. This algorithm is widely used in applications such as e-commerce, social networking, and music videos, and its research and practice have made significant progress with the development of deep learning techniques.

Basic principle: User-based Top-N recommendation algorithm is mainly based on Collaborative Filtering (CF) technology. Collaborative filtering algorithms can be divided into two categories: neighborhood-based and model-based. Neighborhood-based collaborative filtering algorithms work by finding other users with similar interests to the target user, and then recommending items that these users like to the target user. The model-based collaborative filtering algorithm builds a global user-item rating matrix and uses matrix decomposition techniques (such as singular value decomposition, SVD) to predict the user's rating of unknown items.

3.3.2. Item-Based Top-N Recommendation Algorithms

The Top-N recommendation algorithm based on items firstly calculates k most similar items for each item according to the similarity between items. Then the candidate set C of recommended items is determined by taking the union of k most similar items and removing each item in the user's purchased item set U; Finally, the similarity between each item in set C and the item in set U is calculated. The items in the sorted C set are arranged in descending order of similarity, which is the recommended Top-N list based on the items. One problem with this approach is that when the joint distribution of a group of items differs from the distribution of individual items in that group, the above scheme may produce sub-optimal recommendation results. To solve this problem, Deshpande and Karypis developed a higher-level item-based Top-N recommendation algorithm that uses all item combinations up to a specific size when determining the set of items to recommend to the user.

中文译文：

协同过滤技术综述

1. 介绍

从协同过滤的特点和挑战、基于Memory的CF、基于Model的CF、混合CF（Hybrid CF）、评价指标对CF展开介绍。

CF的挑战：高度离散的数据、指数级增长的用户和物品数量（用户-物品矩阵的爆炸性增长）、需要在极短的时间内完成推荐、同义词处理、数据噪声等等

Memory-base CF：根据用户的评分计算用户或物品之间的相似度并做出预测和推荐，实现简单。

Model-base CF：基于模型的CF通常使用机器学习算法来构建预测模型，例如矩阵分解、神经网络、决策树等。这些算法可以通过学习评分数据中的模式和关系来预测新用户对新物品的评分。与基于记忆的方法相比，基于模型的方法通常具有更好的预测性能和可扩展性，但需要更多的计算资源和时间来训练模型。

除了协同过滤外，基于内容过滤的推荐方法也非常重要。简单介绍了collaborative filter和content-base recommender的异同

协同过滤：是基于用户历史行为数据的，它会分析用户对物品的评分或行为数据，然后找到与之相似的用户，将推荐给他们喜欢的物品。

基于内容过滤的推荐系统：是基于物品本身的特征进行推荐，它会分析物品的属性和特征，然后将与用户历史行为相似的物品推荐给他们。

这两种方法在推荐系统中都有自己的优势和不足。协同过滤可以在没有物品数据的情况下进行推荐，但需要足够的用户评分数据。而基于内容的推荐系统可以在没有足够用户评分数据时进行推荐，但需要准确的物品元数据。因此，在实际应用中，这两种方法通常会结合起来使用，以达到更好的效果。

2. 协同过滤的特点与挑战

推荐系统为用户提供快速、精确的内容，从而吸引用户使用这个平台，转化为商品的购买率等效益。构建的推荐算法有以下特点和挑战：

2.1 数据稀疏

系统内通常有海量的内容和用户，因此会造成user-item矩阵极大且非常稀疏。稀疏的数据矩阵会导致很难找到足够相似的用户或物品进行推荐等问题：

冷启动问题：在有的文献中也被称为 new user problem or new item problem，当有新的用户或者新的物品内容加入系统后，由于新内容没有足够的数据，无法找到相似的数据，从而无法对新内容进行推荐。

低覆盖率问题：reduced coverage 当用户的评分数据非常少，而推荐内容的数量非常多时，就会出现reduced coverage rate的问题，推荐系统可能无法为这些用户生成推荐。

邻居传递问题：在推荐系统中，邻居传递是指当两个用户之间没有共同评分时，即使他们具有相似的品味，也无法被识别为相似用户。这可能会降低基于比较用户对的预测来生成推荐的推荐系统的有效性。

邻居传递问题通常是由于稀疏数据库中缺乏数据而引起的。为了解决这个问题，可以使用更复杂的算法来处理，例如基于物品的推荐系统，该系统使用物品的属性和特征来进行推荐，而不是仅仅依赖于用户之间的比较。

解决手段

降低矩阵纬度： 通过奇异值分解、主成分分析等数学手段去掉不重要的用户或物品信息。舍弃的信息可能会导致与推荐相关的信息也丢失了，降低了推荐质量

基于Hybrid CF的冷启动解决方案： 内容增强CF算法利用外部内容信息，有利于解决冷启动问。或通过结合协作信息和内容信息，提出冷启动的推荐方法。或将推荐内容分成不同的组，基于用户评分的高斯分布对用户推荐内容进行预测。

Model-based CF:TAN-ELR是一种基于贝叶斯网络和逻辑回归的分类器，通过树增强和扩展逻辑回归来提高模型的精度和效率。

基于矩阵分解的协同过滤算法： 将用户-物品评分矩阵分解成用户特征矩阵和物品特征矩阵，其中用户特征矩阵包含了用户的隐含特征向量，物品特征矩阵包含了物品的隐含特征向量。通过计算用户特征向量和物品特征向量的内积，就可以得到用户对物品的评分预测值。 常见的如 SVD、NMF、PMF、MMMF。MMMF最大化不同分类的边际间隔使得分类器具有更强的泛化性。

多重插补CF方法和插补增强CF算法则是利用插补方法来填补缺失数据，从而提高推荐系统的性能。

2.2 扩展

当现有用户和项目的数量急剧增长时，传统的CF算法将遭受严重的可伸缩性问题。例如，拥有数千万个客户(M)和数百万个不同的目录项(N)，一个复杂度为O (n)的CF算法太大了。此外，推荐系统要有极高的实时性，并向所有用户提供推荐数据，无论他们的行为数据历史如何，这要求推荐系统的高可伸缩性。

Model-based CF的相关工作提供了一些有效的方案：item-based Person correlation CF：不计算所有项目对之间的相似度，而是仅计算用户共同评价的项目对之间的相似度;简单的贝叶斯CF算法通过基于观察到的用户评分行为进行预测；聚类CF算法：通过在更小和高度相似的集群中而不是在整个数据库中寻求用户的推荐来解决可伸缩性问题；

2.3 同义词

同义词指的是许多相同或非常相似的项目具有不同的名称或条目。推荐系统应该发现词意之间的关联，从而能够推荐类似的内容。这块涉及NLP相关内容。

3. 基于内存的协同过滤技术

基于内存的CF算法使用用户-物品数据库的整个或一个示例来生成预测。每个用户都是一群有相似兴趣的人中的一部分。通过计算一个新用户（或活跃用户）的邻居，可以产生对他或她的新项目的偏好的预测。

The neighborhood-based CF algorithm是一种基于邻域的协同过滤算法，用于推荐系统中的个性化推荐。在这种算法中，用户和物品之间的相似度是通过计算它们之间的距离或相似性来确定的。然后，根据用户之前的评分和相似物品的评分，为用户推荐最相关的物品。这种算法通常用于处理大型数据集，因为它不需要计算所有用户和物品之间的相似度，而只需要计算与目标用户或物品最相似的一小组邻居。因此，它可以提高推荐系统的效率和准确性。

在实际应用中，通常使用基于余弦相似度或皮尔逊相关系数等技术来计算用户和物品之间的相似度。此外，还可以使用KNN（K-Nearest Neighbor）算法来确定与目标用户或物品最相似的邻居。

3.1 相似性计算

3.1.1 Correlation-Based 相似

基于Pearson correlation：

Pearson相关系数（Pearson correlation coefficient）是一种衡量两个变量之间线性关系强度的统计量。它用于衡量两个连续变量之间的相关性，取值范围在-1和1之间，其中0表示没有线性相关性，正值表示正相关性，负值表示负相关性。Pearson相关系数可以用来衡量两个变量之间的线性关系，但并不能说明它们之间是否存在其他类型的关系。Pearson相关系数通常用于探索两个变量之间的关系，例如在数据挖掘、机器学习和统计分析中。它可以帮助研究人员确定两个变量之间是否存在相关性，并在建立预测模型时帮助选择最相关的变量。但需要注意的是，Pearson相关系数只能衡量线性关系，对于非线性关系则不适用。

Pearson相关系数的扩展算法有：constrained Pearson correlation、Spearman rank correlation

后两者：都是通过比较两个变量之间的等级顺序来计算的，而不是直接比较它们的数值。用来衡量两个变量之间的单调关系，即它们是否在一起变化，但不能说明它们之间是否存在其他类型的关系。它们通常用于探索两个变量之间的关系，不需要假设数据呈正态分布或线性相关。

3.1.2. 基于余弦向量的相似度

Adjusted cosine similarity：向量余弦相似度计算两个物品之间的夹角，而调整后的余弦相似度则考虑了不同用户之间使用不同评分尺度的问题。由于不同用户可能使用不同的评分尺度，因此向量余弦相似度无法考虑这种差异。为了解决这个问题，可以使用调整后的余弦相似度。该方法从每个用户的评分中减去该用户的平均评分，然后计算物品之间的余弦相似度。调整后的余弦相似度与Pearson相关系数具有相同的公式，但是它对用户的评分进行了标准化，以纠正评分尺度的差异。

3.2 预测与推荐计算

在邻域方法中，预测过程主要包括两个步骤：选择邻居和计算预测评分。通过这两个步骤，可以为当前用户生成个性化的推荐结果。

3.2.1 加权邻居评分和

首先需要找到与当前用户最相似的一组用户，这些用户被称为邻居。然后，计算每个邻居对当前用户未评价物品的预测评分。

具体而言，对于每个邻居，可以使用其评价过的物品与当前用户未评价物品之间的相似度来计算其对当前用户未评价物品的影响权重。然后将每个邻居的影响权重与其对当前用户未评价物品的评分相乘，得到每个邻居对当前用户未评价物品的加权评分。最后将所有邻居的加权评分相加，得到当前用户对未评价物品的预测评分。这种方法相比于简单平均或加权平均方法，更加准确地反映了邻居之间的差异性。同时，由于使用了邻居之间的相似度来计算权重，因此可以更好地处理邻居之间的差异性。

3.2.2. 简单加权平均

对于item-based的预测，用户u对i的评分如下：

其中r{u,n},n∈N为用户n对其他物品n的评级，w{i,n}为物品i 和n的相似度。

3.3. Top-N Recommendations

3.3.1 基于用户的Top-N推荐算法

用户基础的Top-N推荐算法是一种通过分析用户行为数据来为特定用户推荐最有可能引起其兴趣的N个物品的技术。这种算法在电子商务、社交网络和音乐视频等应用中得到了广泛应用，并且随着深度学习技术的发展，其研究和实践也取得了显著进展。

基本原理:用户基础的Top-N推荐算法主要基于协同过滤（Collaborative Filtering, CF）技术。协同过滤算法可以分为两大类：基于邻域的（neighborhood-based）和基于模型的（model-based）。基于邻域的协同过滤算法通过找到与目标用户兴趣相似的其他用户，然后将这些用户喜欢的物品推荐给目标用户。而基于模型的协同过滤算法则通过建立一个全局的用户-物品评分矩阵，利用矩阵分解技术（如奇异值分解，SVD）来预测用户对未知物品的评分。

3.3.2. 基于项目的Top-N推荐算法

基于物品的Top-N推荐算法首先根据物品之间的相似度计算出每个物品的k个最相似物品；然后通过取k个最相似物品的并集并移除用户已经购买的物品集合U中的每个物品，来确定推荐物品的候选集合C；最后计算集合C中每个物品与集合U中物品之间的相似度。排序后得到的C集合中的物品，按照相似度降序排列，即为推荐的基于物品的Top-N列表。这种方法的一个问题是，当一组物品的联合分布与该组中各个单独物品的分布不同时，上述方案可能会产生次优的推荐结果。为了解决这个问题，Deshpande和Karypis 开发了更高阶的基于物品的Top-N推荐算法，当确定要向用户推荐的项目集时，使用所有项组合，直到特定大小为止。

3.4. 基于内存算法的扩展

3.4.1. Default Voting

在许多协作过滤算法中，仅从用户都评过分的物品的交集中计算成对相似度[5, 27]。当评价数太少以至于无法生成相似度值时，这种方法将不可靠。此外，只关注交集相似度会忽略反映用户整个评分历史中的全局评分行为。

相关研究通过假设一些默认的投票值来填补缺失的评分可以提高CF预测性能。Herlocker等人通过减少具有少于50个共同项目的用户的权重来解决小交集问题。Chee等人使用团体（或小组）的平均值作为默认投票，以扩展每个用户的评分历史。Breese等人对未观察到的评分使用中性或略为负面的偏好，然后在生成的评分数据上计算用户之间的相似度。

3.4.2. 逆用户频率 (IUF)

用于调整用户评分矩阵中的权重，以提高推荐性能。IUF的基本思想是，普遍出现的物品对于区分用户之间的偏好不如罕见的物品重要。因此，IUF通过对用户评分矩阵中的每个物品进行加权来调整物品的重要性，其中权重f\_j=log(n/n\_j)与使用该物品的用户数量成反比，n\_jn为对j物品进行评分的用户总数。这意味着罕见的物品将具有更高的权重，而普遍出现的物品将具有较低的权重。

3.4.3. Case Amplification

Case Amplification通过对权重进行变换来增强相似度高的用户之间的关系，并减弱相似度低的用户之间的关系。这可以通过将权重进行指数变换来实现，其中较高的权重将得到更大的增强，而较低的权重将得到更大的惩罚。

其中ρ是Case Amplification的放大系数，ρ≥1，典型的ρ值为2.5 。Case Amplification可以减少数据中的噪声。它倾向于支持高权重，因为小值的幂次方变得可以忽略。例如，如果权重很高，wi,j = 0.9，则它仍然很高（ 0.9^{2.5} ≈ 0.8）；如果权重很低，例如w{i,j} = 0.1，则它将是可以忽略的（ 0.1^{2.5} ≈ 0.003）。

3.4.4. 赋值增强的CF算法 (IBCF)

Imputation-Boosted CF Algorithm是一种基于物品的协同过滤推荐算法，它使用缺失值填充来提高预测性能。在这个算法中，预测用户对某个物品的评分是通过分析其他用户对类似物品的评分来实现的。与传统的协同过滤算法不同的是，赋值增强的CF算法使用缺失值填充来处理评分矩阵中的缺失值，以提高预测精度。

IBCF-NBM，结合了使用朴素贝叶斯的IBCF和使用均值填补的IBCF。这种混合算法被设计用于不同密度的数据集。对于密集的数据集，使用朴素贝叶斯的IBCF可以更好地处理数据。对于稀疏的数据集，使用均值填补的IBCF可以更好地处理数据。通过将两种算法组合起来，可以提高推荐算法的准确性。