

北方工业大学

硕士 学位 论文



基于深度强化学习的课程思政资源推荐 管理平台开发

学 生 姓 名 康守冲
学 号 2020316210142
学科(专业学位) 计算机技术
研 究 方 向 数据处理技术与软件服务
导 师 杜春涛
校 外 导 师 韩学东

2023 年 6 月 8 日

**Development of Curriculum Ideology and Politics
Resource Recommendation and Management
Platform Based on Deep Reinforcement Learning**

By

Kang Shouchong

**A Dissertation Submitted to
North China University of Technology
In partial fulfillment of the requirement
For the professional degree of
Master of Engineering**

North China University of Technology

June, 2023

基于深度强化学习的课程思政资源推荐管理平台开发

摘要

课程思政是指将思政元素融入到其他课程教学中，让思政课以外的课程也能起到思政教学的作用。强化学习是一种交互式学习方法，相比于静态推荐模型，强化学习的交互更新特性可以更好地解决推荐领域中的用户兴趣偏移问题。

本文针对当前课程思政教学中缺少思政案例问题，通过优化深度强化学习推荐算法、开发课程思政案例资源管理平台，并将优化后的算法应用于该平台，实现了课程思政案例的动态推荐，主要工作如下：

(1) 将用户与推荐系统的交互建模为马尔可夫决策过程，在经典强化学习算法的基础上，结合推荐任务场景，通过模型熵的方法优化算法的损失函数，提出了一种针对推荐多样性优化的深度强化学习推荐算法，提高了推荐结果的多样性；

(2) 在上述算法的基础上，针对智能体的策略更新流程，设计奖励预测模型优化智能体的交互更新过程，提出了一种针对样本效率优化的深度强化学习推荐算法，加快了模型的训练速度，提高了强化学习的样本效率；

(3) 针对课程思政的发展现状，分析教师用户对于课程思政资源的需求，使用容器、缓存、反向代理等开发技术设计并实现了课程思政资源管理平台，为教师用户提供课程思政资源的检索、浏览等功能；

(4) 基于本文提出的优化算法，针对平台中的课程思政资源数据集，开发了课程思政案例推荐系统，并与平台进行整合，实现了为教师用户提供课程思政案例推荐服务。最后对平台的功能实现和性能表现进行了相关测试，结果表明该平台可提供良好的用户体验并具备高可用性。

关键词：课程思政，强化学习，推荐系统，平台开发

Development of Curriculum Ideology and Politics Resource Recommendation and Management Platform Based on Deep Reinforcement Learning

Abstract

Curriculum ideology and politics refer to the integration of ideological and political content into the teaching of other courses, enabling non-ideological and political courses to also serve as vehicles for ideological and political education. Reinforcement learning is an interactive learning method. When compared with static recommendation models, the interactive update feature of reinforcement learning better addresses the issue of user interest shift in the field of recommendations.

Addressing the dearth of ideological and political cases in the current curriculum's ideological and political teaching, this paper optimizes the deep reinforcement learning recommendation algorithm. It develops a curriculum ideological and political case resource management platform and applies the optimized algorithm to enable dynamic recommendation of curriculum ideological and political cases. The main contributions of this work are as follows:

(1) The interaction between users and the recommendation system is modeled as a Markov decision-making process. Building upon the classical reinforcement learning algorithm Actor-Critic, we propose a deep reinforcement learning recommendation algorithm optimized for recommendation diversity. We achieve this by optimizing the loss function of the algorithm through the incorporation of model entropy, thereby enhancing the diversity of recommendation results.

(2) Building upon the aforementioned algorithm, we design a reward prediction model to optimize the agent's interactive update process during the strategy update. By doing so, we propose a deep reinforcement learning recommendation algorithm optimized for sample efficiency. This leads to accelerated training speed of the model and improved sample efficiency in reinforcement learning.

(3) Based on an analysis of the current state of curriculum ideological and political resources and the needs of teachers and users, we design and develop a

curriculum ideological and political resource management platform using technologies such as containers, caches, and reverse proxy. This platform facilitates the retrieval and browsing of curriculum ideological and political resources.

(4) Leveraging the optimization algorithm proposed in this paper, we develop a recommendation system based on deep reinforcement learning for the curriculum ideological and political resource dataset within the platform. By integrating the recommendation system with the platform, we provide teachers with course ideological and political case recommendations. Finally, we test the functionality and performance of the platform, and the results demonstrate its ability to provide an excellent user experience and high availability.

Key words: Curriculum Ideology and Politics, Reinforcement learning, Recommendation System, Platform development

目 录

第一章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 研究现状.....	1
1.3 研究内容.....	2
1.4 论文组织结构.....	3
第二章 深度强化学习推荐算法研究与改进	4
2.1 相关理论.....	4
2.1.1 循环神经网络.....	4
2.1.2 马尔科夫决策过程.....	5
2.1.3 演员-评论家算法	7
2.1.4 常见的推荐算法.....	7
2.1.5 基于强化学习的推荐算法.....	8
2.1.6 主流的评测指标.....	9
2.2 问题陈述.....	10
2.3 多样性优化的推荐算法.....	11
2.3.1 状态表示模型设计	11
2.3.2 决策模型设计.....	13
2.3.3 实验设置.....	16
2.3.4 模型对比实验.....	17
2.3.5 消融实验.....	19
2.4 样本效率优化的推荐算法	19
2.4.1 奖励预测模型设计	20
2.4.2 交互流程优化.....	21
2.4.3 实验设置.....	24
2.4.4 模型对比实验.....	26
2.4.5 消融实验.....	27
2.5 算法移植与部署.....	28
2.5.1 泛化能力分析.....	28
2.5.2 模型预训练.....	29
2.5.3 交互更新.....	30
2.6 本章小结.....	31
第三章 平台设计	32
3.1 需求分析.....	32
3.2 平台架构设计.....	34

3.3 功能模块设计	36
3.4 数据库设计	39
3.5 本章小结	43
第四章 平台实现	44
4.1 相关开发技术	44
4.2 开发环境搭建	49
4.3 平台架构实现	52
4.4 基本功能实现	54
4.5 基于优化算法的课程思政案例推荐	61
4.5.1 数据集与数据清洗	61
4.5.2 推荐系统集成	62
4.5.3 课程思政案例推荐实现	62
4.6 本章小结	64
第五章 平台测试	65
5.1 测试环境	65
5.2 功能测试	65
5.3 性能测试	70
5.4 测试结果分析	73
5.5 本章小结	73
第六章 总结与展望	74
6.1 总结	74
6.2 展望	74
参考文献	76

第一章 绪论

1.1 研究背景及意义

2016 年，习近平总书记在全国高校思想政治工作会议上发表重要讲话，指明了高校各类课程和思想政治理论课必须同向同行、形成协同效应^[1]。2017 年，教育部发布《高校思政工作质量提升实施纲要》，指出高校课程思政协同创新提出是对现实问题的积极探索与回应^[2,3]。2020 年，教育部印发《高等学校课程思政建设指导纲要》，作为全面推进高校课程思政建设的指导性文件，为高校课程思政研究指明了方向^[4]。以“高校课程思政”为主题的相关研究比去年的相关研究在数量上增长明显，成为年度研究热点。

在这种背景下，高校教师都在积极开展课程思政教学，但课程思政教学主要存在以下问题：（1）课程思政资源数量少且分散，许多教师在开展课程思政教学时难以找到合适案例；（2）课程思政资源质量难保证，对思政教学不熟悉的教师难以辨别网络中课程思政资源的质量，可能导致课程知识点与思政元素不能很好融合，影响教学效果；（3）课程思政教学各自为政，教师关于课程思政教学的交流讨论范围具有局限性，难以做到集思广益。

随着网络技术的发展和计算机软硬件成本的降低，信息技术在教育领域发挥着越来越重要的作用。网课平台、教务平台等信息技术的应用大大提高了工作效率，为教师省下了宝贵的时间。

本文针对当前课程思政存在的实际问题，设计并开发了课程思政资源推荐管理平台，主要实现以下功能：（1）为教师提供课程思政相关资源的浏览、发布、点赞和评价功能；（2）为教师提供课程思政教学交流平台，教师可与其他教师进行交流研讨；（3）为教师提供课程思政资源的推荐服务，协助教师找到更合适的资源；（4）为平台管理者提供基本的管理、审核等功能；（5）平台应具备良好的可维护性和扩展性，以便后续的功能变动和增加。

1.2 研究现状

2014 年，上海市率先开展课程思政改革试点工作，社会反响热烈。2016 年，在全国高校思想政治工作会议上，习近平总书记强调要坚持把立德树人作为中心环节，把思想政治工作贯穿教育教学全过程。2018 年，课程思政在全国推广，

学界也掀起了课程思政研究热潮。近年来，随着课程思政建设的不断深入，课程思政研究领域的论文数量逐年上升^[5]。

对于课程思政平台建设方面，李羽佳针对上海市课程思政教学改革中网络思想政治教育平台的建设情况，梳理了平台建设面临的主要困境，就深化课程思政教学改革发展和搭建学生成长成才需要的网络思政教育平台提出实施路径^[6]。孙倩文等人针对教员的需求，使用 WordPress 技术设计并开发了面向青年教员的课程思政资源平台，加强了青年教员课程思政资料获取能力，推动了课程思政成果交流和共建共享^[7]；胡晓丽等人针对教师和学生的需求，使用 Java 语言在微服务架构的基础上编写了前后端分离的计算机应用基础课程思政教学平台，将课程的学习、讨论和研究融为一体，建立了学生和老师联系的同时实现了教师对课程思政内容和教授形式的设计和管理^[8]。

对于课程思政资源推荐方面，刘伟等人设计并研发了一套智能课程思政教学素材管理系统，采用基于物品的协同过滤算法实现了对课程思政素材的智能推荐^[9]；赵营颖等人采用用户偏好度计算方法，结合大数据技术设计并开发了一套课程思政教学资源推荐系统，实现了对课程思政教学资源的推荐^[10]；洪雅敏设计了基于卷积神经网络的偏好课程思政预测模型，实现了在线课程思政资源智能推荐^[11]；何胜等人采用画像建模的方法，分别构建用户画像和资源画像，实现了个性化课程思政资源推荐^[12]。

综上所述，当前针对课程思政资源平台开发的研究缺乏健壮性，同时尚未有学者将深度强化学习应用到课程思政资源的推荐任务中。为了增强平台的健壮性和稳定性，本课题将采用成熟稳定的 Spring 框架为基础框架，结合反向代理、负载均衡、缓存和数据库技术等共同完成平台的开发工作，并针对课程思政资源数据，基于深度强化学习技术设计开发课程思政资源推荐系统，实现平台的课程思政资源推荐。

1.3 研究内容

首先在经典强化学习算法的基础上，分析平台用户与推荐系统的交互过程并将其建模为马尔可夫决策过程，设计了一种针对推荐多样性优化的强化学习推荐算法，该算法通过改进损失函数，对智能体重复推荐单一物品的行为进行惩罚，提高了推荐结果的多样性。

然后针对强化学习的样本效率问题，设计了一种针对样本效率改进的推荐算法，使用真实的交互经验构建环境模型，提高了样本效率和训练速度。经实验证

明，两种算法的性能表现均优于改进前的算法，并阐述了该推荐算法在课程思政数据集上实现推荐任务的可行性。

针对当前课程思政的发展存在的问题，分析教师用户的需求，本文设计并开发了一个课程思政资源推荐管理平台，并基于提出的优化算法开发了适用于课程思政资源数据集的推荐系统，实现了课程思政资源的共建、共享和推荐。平台的资源平台和推荐系统分别采用Java和Python语言，基于Spring和TensorFlow进行开发，二者相互协作，共同实现平台的推荐功能。最后对平台的功能实现和性能表现进行了相关测试，结果表明平台具备良好的用户体验和健壮性。

1.4 论文组织结构

论文组织结构如下：

第一章，绪论。介绍课程思政的研究背景，总结国内学者对于课程思政平台建设的研究，介绍现有课程思政平台建设和思政推荐现状，以及论文的主要研究内容。

第二章，深度强化学习推荐算法研究与改进。首先介绍了算法研究的相关理论基础，然后对本文提出的针对多样性优化的推荐算法和针对样本效率优化的推荐算法进行了详细的叙述，通过实验证明了算法改进的有效性，最后对算法的移植和部署工作进行了阐述。

第三章，平台设计。从需求分析、平台架构、功能模块和数据库对平台设计进行了细化，明确平台的开发路线。

第四章，平台实现。介绍了相关开发技术和环境，对平台的架构、功能模块和基于优化算法的课程思政案例推荐进行实现。

第五章，平台测试。设计测试用例，分别对平台的功能实现与性能表现进行了测试，测试结果表明该平台具备良好的用户体验和优秀的健壮性。

第六章，总结与展望。对本文所做工作进行总结，分析平台的优势与不足，并对未来的研究进行展望。

第二章 深度强化学习推荐算法研究与改进

2.1 相关理论

2.1.1 循环神经网络

循环神经网络（RNN， Recurrent Neural Network）是一种特殊的神经网络，该网络内部存在连接回路，具有时间序列的记忆能力，可以用于序列数据的处理。循环神经网络的前身是由 Hopfield^[13]提出的 Hopfield 网络模型，但后来被 Jordan^[14]和 Elman^[15]提出的循环神经网络框架所取代，其被认为是当前广泛使用的循环神经网络的基础版本。

循环神经网络与传统神经网络的区别在于，它在神经元之间添加了一条循环连接，使得网络可以在时间维度上进行运算，从而具有记忆能力。在循环神经网络中，每个时间步的输入和上一个时间步的输出都会被传递到下一个时间步，这样就形成了一个循环的结构。在每个时间步，神经网络会根据当前的输入和上一个时间步的输出来计算当前时间步的输出。这种结构使得循环神经网络可以对不同长度的序列数据进行处理，从而具有广泛的应用价值。

图 2-1 展示了循环神经网络的网络结构，通过隐藏层的自连接回路，使得当前时刻可以获取前一时刻的网络状态，并且下一时刻也可以获取当前时刻的网络状态^[16]。

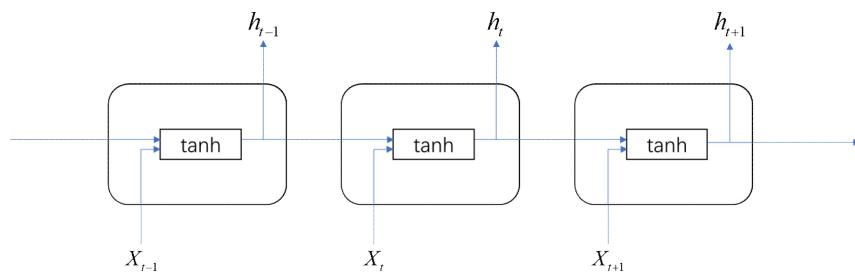


图 2-1 循环神经网络结构

如图所示，隐藏层单元在时间步骤 t 接收两个输入：前一时间步骤的隐藏层单元值 h_{t-1} 和当前的输入数据 x_t 。通过隐藏层单元 h_t 的计算，输出值将被传递到下一个时间步骤，作为下一步骤的隐藏层单元的输入。循环神经网络一般通过基

于时间的反向传播算法^[17] (BPTT, Back Propagation Through Time) 对网络中的参数进行更新，实现网络模型的训练，循环神经网络的反向传播如下所示。

$$\begin{aligned} h_t &= \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \\ o_{t+1} &= W_{hy}h_t + b_y \\ y_t &= \text{softmax}(o_t) \end{aligned} \quad (2-1)$$

其中 W_{xh} 为输入单元到隐藏单元的权重矩阵， W_{hh} 为隐藏单元之间的连接权重矩阵， W_{hy} 为隐藏单元到输出单元的连接权重矩阵， b_y 和 b_h 为偏置向量。由于计算过程中参数共享的特点，循环神经网络理论上可以处理任意长度的序列数据，并且可以根据任务要求调整输出序列数据的长度。

在交互式推荐任务中随着推荐交互的进行，用户与推荐系统的交互记录将持续增长，即在 t 时刻模型的输入数据为 $t-1$ 时刻之前的所有交互数据，这意味着模型需要支持输入任意长度的交互记录。循环神经网络具备可处理任意长度序列数据的特性，可以很好地解决强化学习在推荐任务中的数据输入问题。

2.1.2 马尔科夫决策过程

强化学习 (Reinforcement Learning) 的概念源于心理学对于人类学习过程的研究，人类学习过程是与环境互动的过程，通过与环境的交互并根据观察到的反馈来调整我们的行为，以一种不断试错的方式学习。强化学习算法借鉴了这一思想，通过构建智能体并设定交互的环境，实现以试错的方式进行学习更新。根据智能体对环境学习方式的不同可以分为有模型的强化学习和无模型的强化学习，相应的，对于有模型的强化学习，解决方式通常采用动态规划，而对于无模型的强化学习则采用随机采样法，通过多次试探从经验中获得平均值。

马尔科夫决策过程^[18] (Markov Decision Process) 是强化学习最基本的模型，强化学习一般应用在具有马尔科夫性质的序列过程中。马尔科夫决策过程是一个精心设计的模型，它体现了智能体与环境之间交互的本质，并且在此基础上可以进行严谨的数学运算和逻辑推理。一个完整的马尔可夫决策过程需要包含以下五个元素：

- (1) 状态空间 (State Space) S：是包含当前环境所有状态的集合，根据环境的不同可以是离散的，也可以是连续的。
- (2) 动作空间 (Action Space) A：是包含智能体在当前环境中所有可执行动作的集合。
- (3) 奖励 (Reward) R：即智能体获得的反馈，通常是智能体在当前状态 S

下做出动作 A 所收获的正反馈或负反馈。

(4) 转移概率 (Transition Probability) P: 表示当智能体对于环境状态 S 做出动作 A 后, 环境状态由当前状态 S 转移至下一状态 S' 的概率。

(5) 折扣因子 (Discount Factor) $\gamma \in (0,1]$: 用于平衡智能体所面对的延迟奖励与即时奖励, 数值越接近 0, 智能体就会越倾向于获得即时奖励。



图 2-2 强化学习交互流程

强化学习中智能体与环境的交互流程如图 2-2 所示, 将交互的过程记录下来就形成了一个典型的数据流, 也称为强化学习交互经验, 如公式(2-2)所示。

$$S_1, A_1, R_1, S_2, A_2, R_2 \dots \quad (2-2)$$

强化学习智能体的目标是将获取的累计奖励 (或回报) 最大化, 累计奖励的表示如公式(2-3)所示。

$$G_t = E_{\pi}[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots] \quad (2-3)$$

其中 π 表示当前智能体的动作策略 $\pi: S \times A \rightarrow [0,1]$, t 为交互时刻。由累计奖励可以得到状态价值, 其表示了智能体从当前时刻开始按照动作策略 π 进行交互所能获得的累计奖励。状态价值的表示如公式(2-4)所示。

$$V_{\pi}(s) = E_{\pi}[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots | S_t = s] \quad (2-4)$$

当问题可以被分解为多个交互相合, 而每个回合的交互可以通过有限的步数结束时, 强化学习算法通常被设计为找到使得初始状态的状态价值最大化的最优动作策略 π^* , 随后使得智能体的动作遵循该策略。

本文将用户与推荐系统的交互建模为马尔科夫决策过程, 从而实现将强化学习应用到推荐任务, 智能体和环境分别对应推荐系统和用户, 奖励 R 为用户可能给予的反馈值范围, 动作空间 A 为当前推荐系统中候选的推荐物品, 状态空间 S 即根据交互记录计算可以得到的环境状态表示的范围。

2.1.3 演员-评论家算法

演员-评论家算法也称为 Actor-Critic^[19]算法，是强化学习领域中的重要算法之一，其算法流程如图 2-3 所示，其结合了以值为基础和以动作概率为基础的两种强化学习方法。

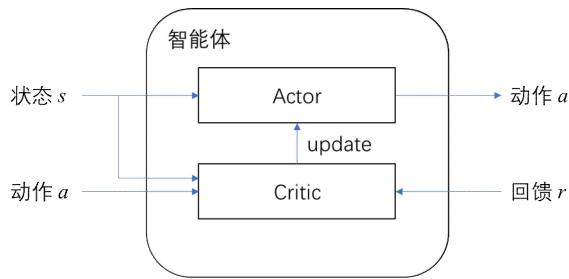


图 2-3 Actor-Critic 算法流程

Actor-Critic 算法分为 Actor 智能体和 Critic 反馈器两部分，可对值函数和策略网络进行逼近，其中策略网络由智能体通过策略梯度估计方法进行梯度下降学习得来，而值函数由反馈器采用时间差分（TD, Temporal Difference）来实现。图 2-3 给出了算法的执行流程：对于状态 s ，智能体根据当前的策略网络选择动作 a ，状态 s 收到动作 a 后，转移至状态 $s+1$ ，并产生一个回馈值 r ；状态 s 和回馈值 r 将输入到反馈器，并经过值函数输出预估的反馈，即智能体获得来自反馈器的反馈，表示智能体在当前状态下采取动作获得的奖惩，并采用时间差分方法对值函数和策略网络进行更新学习，根据反馈调整策略网络的动作选取策略。

由于 Actor-Critic 方法结合了策略学习和值函数学习的优点，通过引入 Critic 网络来估计行动的价值，Actor 网络可以更好地指导策略更新。这种结合可以提高推荐算法的稳定性和效率，同时兼顾探索和利用的平衡，从而更好地适应不断变化的用户偏好和环境，因此本文选用 Actor-Critic 方法作为推荐决策的更新方式。

2.1.4 常见的推荐算法

推荐系统是一种信息过滤技术，通过向用户提供个性化的内容来解决信息过载问题^[20]，目前推荐系统已成为诸多应用领域的关键技术，也是学术界关注的热点^[21]。按照技术原理，常见的推荐算法可以分为两类，即协同过滤推荐算法^[22]和基于内容的推荐算法^[23]。

协同过滤推荐算法是推荐系统应用最为广泛的算法之一。它分为基于内存的

协同过滤和基于模型的协同过滤两类，其中基于内存的协同过滤又分为基于用户的协同过滤和基于物品的协同过滤。最早出现的是基于用户的协同过滤推荐算法，它计算与目标用户兴趣相似的用户集合，然后为目标用户推荐该相似用户集合中用户可能喜欢但未接触过的物品。而目前业界使用最多的是基于物品的协同过滤推荐算法，该算法通过分析记录的用户行为计算物品之间的相似度，进而为目标用户推荐与用户喜欢的物品相似的物品。基于模型的协同过滤法则主要通过机器学习和数据挖掘模型，利用分类、回归、矩阵分解等算法提取用户和物品的隐含模式进行推荐^[24]。基于模型的协同过滤相比于基于内存的协同过滤具有更完善的理论支撑，并且内存占用少，推荐精度高，但难以进行推荐解释。

基于内容的推荐算法是一种利用物品内容特征进行推荐的算法，相较于协同过滤推荐算法，其具有更强的物品独立性和可解释性。不需要其他用户的行为数据，基于内容的推荐算法能够提供个性化的推荐结果。但是，该算法也存在一些缺点，如对于物品内容特征较弱或者无法抽取的物品，难以进行推荐；同时，该算法也容易面临“过度推荐”问题，即仅向用户推荐与其历史喜好相同的物品，而忽略了用户可能的兴趣变化和探索新事物的需求^[25]。

2.1.5 基于强化学习的推荐算法

近年来基于强化学习开发的应用越来越多，强化学习在推荐算法中的应用被越来越多的研究者重视起来。在推荐任务中，推荐系统对应着智能体，平台内容对应智能体可以做出的动作，平台用户对应着智能体交互的环境，推荐系统不仅可以根据在线用户反馈学习，还可以从离线历史数据进行学习，构建出一个最优推荐策略，并遵循这个推荐策略完成推荐。将强化学习应用到推荐领域中，可以使得推荐算法相比于传统推荐算法具备更高的推荐性能，并且得益于强化学习的动态更新方法，其能够根据用户的反馈实时调整自身的推荐策略，从而可以解决用户的兴趣偏移问题，基于强化学习的推荐流程如图 2-4 所示。

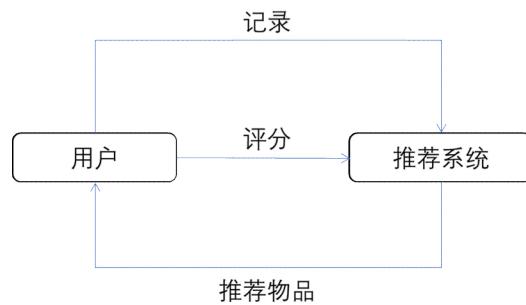


图 2-4 强化学习推荐流程

有许多学者对基于强化学习的推荐算法进行了研究, Li 等人^[26]将新闻文章的推荐建模为上下文老虎机, 提高了点击率; Chen 等人^[27]使用强化策略梯度方法设计了一种基于强化学习的 Top-K 推荐, 提高了用户长期满意度; Zheng 等人^[28]将深度学习和强化学习相结合, 设计了一种深度强化学习推荐系统, 并表现出了更优的性能。

2.1.6 主流的评测指标

根据推荐任务的不同, 平均绝对误差 MAE 和均方根误差 RMSE 常用于评估预测用户对物品的评分的准确性^[29], 而准确率 Precision 和召回率 Recall 则用于评估推荐系统生成的推荐列表的性能^[30]。此外, 还有一些其他的评价指标, 如 F1-score、NDCG 等, 也被广泛应用于推荐系统的评估。

平均绝对误差 MAE 用于度量推荐算法的预测评分与真实评分之间的差异, 这个值越小, 表示模型预测的准确性越高, 其定义如公式(2-5)所示。

$$E_{MAE} = \frac{1}{n} \sum_{i \in U, j \in I} |p_{ij} - r_{ij}| \quad (2-5)$$

其中 p_{ij} 为用户 i 对项目 j 的估计评分, r_{ij} 为用户 i 对项目 j 的真实评分, 集合 U 和 I 分别为测试集中的用户集合和项目集合, n 为 p_{ij} 的个数。

均方根误差 RMSE 是 Netflix 推荐竞赛所采用的一种广泛应用的评价准则, RMSE 的值越小, 表示算法的准确度越高, 其定义如公式(2-6)所示。

$$E_{RMSE} = \sqrt{\frac{1}{n} \sum_{i \in U, j \in I} (p_{ij} - r_{ij})^2} \quad (2-6)$$

准确率 Precision 用于表示推荐列表中有多少是用户实际喜欢的物品, 其定义如公式(2-7)所示。

$$precision = \frac{1}{n} \sum_{i=1}^n \frac{|L_i \wedge R_i|}{|L_i|} \quad (2-7)$$

其中 L_i 为推荐算法为用户 i 产生的推荐列表, R_i 表示测试集中用户 i 喜欢的全部项目。

Recall 指标在推荐系统中用于衡量推荐列表中所包含的用户喜爱物品数量的比例, 它的定义如公式(2-8)所示, 即推荐列表中用户喜欢的物品数除以用户实际喜欢的全部物品数。因此, Recall 值越高表示推荐系统能够更好地覆盖用户的喜好, 提供更多符合用户兴趣的推荐物品。

$$recall = \frac{1}{n} \sum_{i=1}^n \frac{|L_i \wedge R_i|}{|R_i|} \quad (2-8)$$

强化学习的一个典型评价标准是计算每回合交互的平均累计奖励（Average Reward），它代表着当前智能体策略的好坏，由于强化学习算法的目标是最大化累计奖励，因此通过平均累计奖励曲线的增减，可以直观地看到当前智能体策略的性能表现，平均累计奖励的定义如公式(2-9)所示。

$$AverageReward = \frac{\sum_{t=1}^T R_t}{T} \quad (2-9)$$

2.2 问题陈述

信息时代由于信息量远超过个人的信息处理和利用能力，用户无法准确挑选和运用有效信息^[31]。随着本平台的运行和用户数量的增加，平台中的资源数量将持续增长，可以预见的是在不远的将来平台将面临信息过载问题，因此，开发应用于本平台的推荐系统是非常有必要的。推荐系统的加入不仅可以解决信息过载问题，更可以分析用户喜好或需求，为用户提供更有价值的项目推荐，增强用户对于平台的粘性，有利于用户数量和平台规模的增长。

随着近年来关于强化学习受到越来越多的关注，有很多学者尝试将强化学习应用到推荐算法中^[32-36]，但尚未有学者将强化学习推荐技术应用于课程思政资源中。基于强化学习设计的推荐是一个决策过程，推荐系统向用户推荐项目，用户给出该项目的评分，收到评分后推荐系统将调整自身的推荐策略，再次向用户提供推荐并重复该过程，最终目标是使得推荐系统学习到最佳推荐策略。将强化学习应用到推荐任务中，其优势主要有两点：（1）推荐系统能够利用来自用户的交互式反馈来改进自身的策略，相比于传统的推荐算法可以更好地贴合用户兴趣的动态变化；（2）强化学习的学习策略使得推荐系统不仅可以考虑当下的即时奖励，还可以将长期累积奖励为目标进行推荐。

本课题根据平台资源类型与特征，将用户与平台的交互过程建模为马尔可夫决策过程，结合深度学习领域中的循环神经网络，将推荐问题转变为序列决策问题，以经典的强化学习算法为基础，使用一步 Actor-Critic 作为强化学习方法，分别从推荐多样性和推荐准确率两个方面对算法进行了改进：通过对 Actor-Critic 算法的损失函数增加约束项，改善了推荐算法的推荐结果多样性；通过使用真实的交互记录对智能体设计经验回顾，改善了推荐算法的准确性。经实验评估，本

章提出的针对推荐多样性优化的算法和针对样本效率优化的算法，均优于改进前的推荐算法。

2.3 多样性优化的推荐算法

本小节将介绍一种针对推荐多样性优化的深度强化学习推荐算法，该算法以经典的 Actor-Critic 算法为基础，结合推荐任务场景，设计并实现了状态表示模型和决策模型，通过对损失函数进行优化改进，提高了推荐结果的多样性。

将强化学习技术运用于推荐，核心的是针对推荐任务设计状态表示模型和决策模型。状态表示模型以用户与推荐系统的交互记录作为输入，为智能体提供当前环境的状态表示；决策模型则是智能体决策的关键，智能体会根据当前环境的状态表示，使用决策模型输出下一个推荐动作。

2.3.1 状态表示模型设计

状态表示模型在强化学习中对智能体有着重要的作用，对于本文的推荐任务，状态表示模型可以为智能体提供维度压缩、上下文建模和泛化能力，从交互数据中提取关键信息，将其转化为可以由强化学习智能体处理的向量表示。

(1) 维度压缩：在推荐任务中由于动作空间维数为可推荐的物品总数，使得推荐动作的向量具有很高的维度，状态表示模型可以对其进行降维，将高维数据转化为低维的向量表示，从而减少模型的复杂性和计算开销，同时提高推荐算法的效率和可扩展性。

(2) 上下文建模：在交互式推荐中，智能体接收的交互序列数据具有很强的时序性，状态表示模型可以利用序列数据的时序性建模用户与智能体交互的上下文信息，并通过循环神经网络的隐藏层参数共享融入先前的状态信息，捕捉到用户浏览偏好的变化趋势，使得智能体能更好地理解用户当前的偏好需求。

(3) 泛化能力：状态表示模型可直接从原始交互数据中提取隐含特征，使得推荐算法具备对于不同数据集的泛化能力，帮助智能体发现用户行为中潜在的模式和规律，从而更好地适应新的用户和物品，实现个性化的推荐。

由于用户与智能体的交互序列中包含了智能体的推荐决策和用户对推荐结果的反馈值，为方便后续执行嵌入操作，需要首先对数据进行预处理，将智能体的动作数据与用户的反馈数据分隔开，因此本文的状态表示模型需要数据处理层对数据实现预处理；为了将推荐动作和用户反馈奖励映射到低维度的向量空间中，将其表示为具有语义信息的向量，需要分别对推荐动作和用户反馈奖励设计

嵌入层和编码层；随着交互式推荐的进行，智能体与用户的交互记录长度将持续增长，即模型需要能够支持输入任意长度的交互序列数据并将其转换为固定长度的向量表示，因此需要引入循环神经网络，为了尽可能降低模型复杂度，本文选用了标准循环神经网络作为状态表示模型的底层。

综上所述，本文设计的状态表示模型由数据预处理层、动作嵌入层、奖励编码层和循环神经网络构成，模型结构如图 2-5 所示。

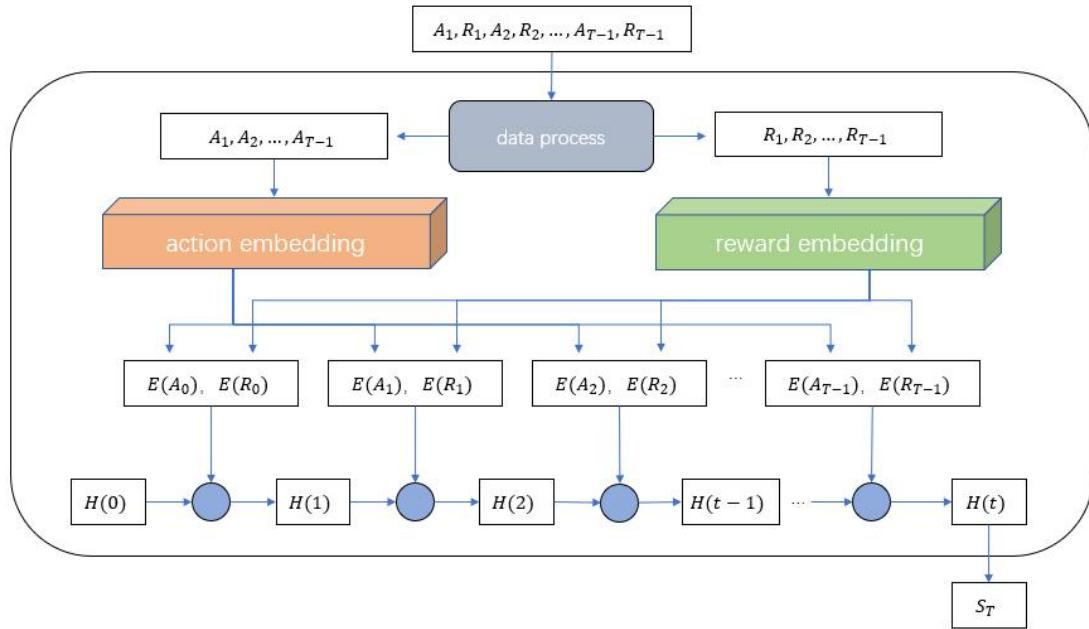


图 2-5 状态表示模型结构

该状态表示模型将交互序列 $A_0, R_0, A_1, R_1, \dots, A_{t-1}, R_{t-1}$ 作为输入，并以一个固定长度的向量 S_t 作为输出，作用是根据 $t-1$ 时刻之前用户与推荐系统的交互记录，计算得到 t 时刻环境的状态表示向量 S_t 。

输入的交互序列会先经过数据预处理，将其中的动作记录和奖励记录分隔开，形成动作序列 A_0, A_1, \dots, A_{t-1} 和奖励序列 R_0, R_1, \dots, R_{t-1} ，随后动作序列和奖励序列将分别输入到动作嵌入层和奖励编码层，动作嵌入层和奖励编码层分别用于将动作和奖励信息转化为向量表示。动作嵌入层将智能体在交互中选择的推荐动作转换为向量表示，以捕捉动作的语义和相关性。奖励嵌入层将交互过程中用户做出的反馈奖励转换为向量表示，以反映奖励的重要性和价值。

动作嵌入层将每一个推荐动作映射为嵌入空间的一个 k 维向量，其实现原理如公式(2-10)所示。其中 O_{A_i} 是长度为 num_items 的一维向量，其内部 A_i^h 的位置上的元素为 1，其余元素均为 0。

$$E(A_i) = E_{k \times num_items} \cdot O_{A_i}, i \in [0, num_items] \quad (2-10)$$

奖励编码层选用 one-hot 编码的方式将奖励数值映射为长度统一的 l 维向量，其实现原理如公式(2-11)所示。其中 l 为编码后的维度，也是输出向量的长度， $\text{floor}(x)$ 的作用是将 x 向下取整，其结果为不大于 x 的最大整数， $(a, b]$ 为奖励 R 的取值范围，对于 $\text{onehot}(i, l)$ 输出的 l 维向量，其中 i^{th} 的位置值为 1，其余位置为 0。

$$E(R_i) = \text{onehot}\left(l - \text{floor}\left(\frac{l * (b - r)}{b - a}\right), l\right), i \in [0, num_rewards] \quad (2-11)$$

交互序列经过预处理、动作嵌入以及奖励编码后传入循环神经网络，用于建模序列数据的时序关系。循环神经网络以动作嵌入向量和奖励嵌入向量作为输入，并通过参数共享更新隐藏层节点，捕捉序列数据中的时间依赖性。本文状态表示模型中使用的循环神经网络包含 t 个隐藏层神经元节点，对应了在 $t-1$ 时间步输入到状态表示模型的 t 个交互数据，其计算结构如公式(2-12)所示。

$$\begin{aligned} h_1 &= \tanh(W[h_0, E_{A_0}, E_{R_0}] + b) \\ h_2 &= \tanh(W[h_1, E_{A_1}, E_{R_1}] + b) \\ &\dots \\ S_t &= h_t = \tanh(W[h_{t-1}, E_{A_{t-1}}, E_{R_{t-1}}] + b) \end{aligned} \quad (2-12)$$

其中 t 为交互序列的时间步， $\tanh(x)$ 为双曲正切激活函数，可将输入的数值映射为 -1 至 1 之间， W 为隐藏层节点权重， b 为偏置项。隐藏层的每个计算节点都同时接收来自动作嵌入层、奖励编码层和前一节点的数据输入，以第 t 个隐藏层节点为例，其输入分别为动作向量 E_A 、奖励向量 E_R 和来自前一个隐藏层节点的共享参数 h_{t-1} ，最终循环神经网络输出当前环境状态的向量表示，包含了交互记录中的状态信息和上下文。

通过该状态表示模型，利用智能体与用户的交互记录，智能体可以学习到一个有效的状态表示，以便更好地理解用户行为模型和环境状态，给出个性化推荐决策。该模型不依赖于预先定义物品内容的特征工程，而是通过学习从原始交互数据中提取隐含特征，提高了推荐算法的灵活性和泛化能力。

2.3.2 决策模型设计

决策模型是智能体做出动作决策的关键，在本文的推荐任务中，决策模型主要有推荐动作选择、策略学习、平衡探索与利用和个性化适应的作用，其根据当前的环境状态和学习到的推荐策略，决定向用户推荐哪些物品或内容。

(1) 推荐动作选择：决策模型的主要作用是在动作空间中选择推荐动作，实现向用户推荐物品或内容。决策模型根据当前的环境状态和习得策略，对可选的推荐动作进行评估和排序，然后选择具有最高预期奖励的推荐动作进行推荐。

(2) 策略学习：决策模型通过智能体与用户的交互记录，使用如 Q-Learning、Policy Gradient 等强化学习算法来学习并优化推荐策略，本文采用 Actor-Critic 算法实现策略学习，以实现最大化累积奖励。

(3) 平衡探索与利用：探索与利用问题指的是在智能体学习过程中如何平衡探索新的行为和利用已知的行为获得最佳奖励的问题。在推荐任务中，探索指尝试新的推荐动作，以发现未知但可能更好的推荐策略；利用是指基于已知的策略选择已经被验证有效的推荐动作。决策模型需要根据当前的状态和学习的经验来决定何时进行探索和何时进行利用，以取得最佳的推荐效果。

(4) 个性化适应：决策模型的学习能力使得推荐系统能够对不同的用户进行适应学习，从反馈中学习用户的偏好，根据用户的兴趣和需求，自适应地调整推荐策略，提供更符合用户期望的推荐结果，实现用户个性化的推荐。

为了在实现个性化推荐的同时，使得推荐结果具备一定的随机扰动，本文设计的决策模型并不直接生成具体的推荐物品，而是计算得到当前动作空间中每一个动作其对应的推荐采样概率。随后基于决策模型输出的采样概率对动作空间中可选择的推荐动作进行随机采样，得到长度为 k 的候选物品列表，最后经过排序、筛选得到最终的推荐物品，决策模型结构如图 2-6 所示。

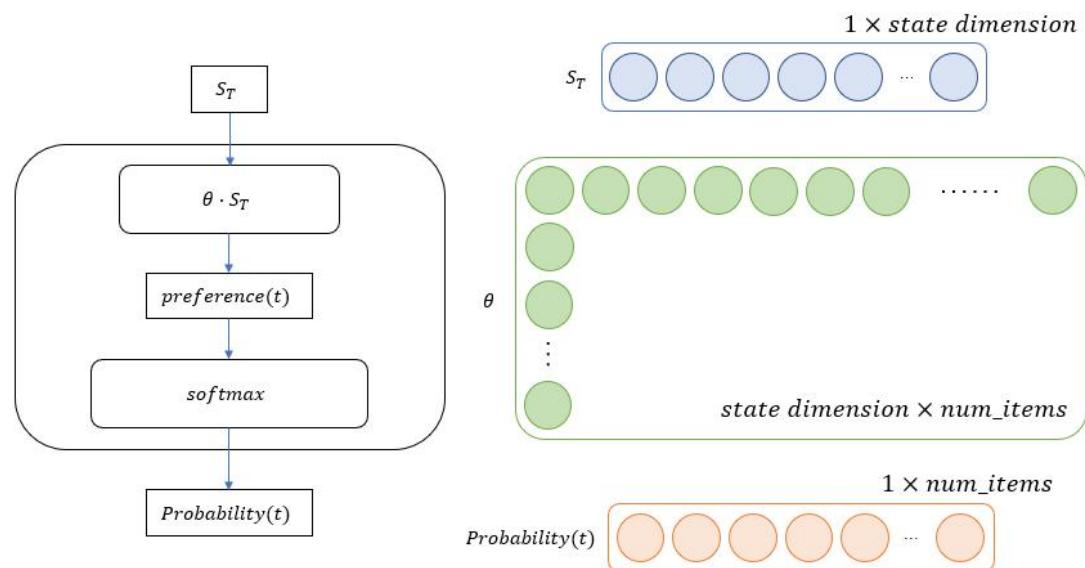


图 2-6 决策模型结构图

决策模型通过与用户的交互获得奖励反馈，学习并优化推荐策略，智能体的

推荐动作提供参考，在 t 时刻下，交互序列经过状态表示模型得到当前的环境状态 S_t ，之后将经由两个全连接层计算得到执行推荐动作对应的概率。

第一个全连接层中包含与可推荐物品数量相等的节点数，可根据输入的 S_t 计算得到候选推荐物品的偏好向量 h_t ，实现原理如公式(2-13)所示。

$$h_t = \Theta_{num_items \times d} \cdot S_t \quad (2-13)$$

其中 Θ 为权重矩阵， d 为环境状态表示向量 S_t 的长度， h_t 中的元素表示在当前状态 S_t 下，智能体选择该物品的偏好值。

第二个全连接层为 soft-max 激活函数，负责将先前得到的物品偏好向量 h_t 映射为动作概率 P_t ，其实现原理如公式(2-14)所示。

$$P_t = \frac{e^{h_i}}{\sum_{i=1}^{num_items} e^{h_i}}, i \in [1, num_items] \quad (2-14)$$

Actor-Critic 是强化学习领域中一种重要的学习方法，其将基于值的方法和基于策略的方法相结合，以期实现最大化累积奖励，相比于 REINFORCE^[37]方法的回合更新机制，Actor-Critic 可以执行单步更新，提高了学习效率，其损失函数与参数更新步骤如公式(2-15)所示。

$$\begin{aligned} \theta_{t+1} &\leftarrow \theta_t + \alpha (R_t + \gamma \hat{v}_w(S_{t+1}) - \hat{v}_w(S_t)) \nabla_\theta \ln \pi_\theta(S_t, A_t) \\ w_{t+1} &\leftarrow w_t + \beta (R_t + \gamma \hat{v}_w(S_{t+1}) - \hat{v}_w(S_t)) \nabla_w \hat{v}_w(S_t) \end{aligned} \quad (2-15)$$

其中 π 为当前智能体的策略网络， \hat{v} 为当前 S_t 环境状态下预测的状态价值， R_t 为交互序列中智能体获得的奖励。

在机器学习中，通常引入交叉熵的概念来对损失函数进行优化调整，其表示了模型预测结果的不确定性。在概率预测任务中，熵值与模型预测的准确度呈负相关，熵值越高，模型预测结果的不确定性就越高，因此通常需要尽可能地降低模型的熵值来实现更高的准确性。

在强化学习中，智能体通常需要在即时奖励与未来奖励之间做出选择，即时奖励的权重越高，智能体的动作策略就趋向保守。对于本文的推荐任务，本人希望使得智能体的推荐结果尽可能地覆盖到更多具备潜在价值的推荐物品，而不是只推荐用户给出高奖励值的部分物品，因此本文基于模型熵的方法为算法的损失函数加入了一个动态约束项，如公式(2-16)所示。

$$\theta_{t+1} \leftarrow \theta_t + \alpha \left((R_t + \gamma \hat{v}_w(S_{t+1}) - \hat{v}_w(S_t)) \nabla_\theta \ln \pi_\theta(S_t, A_t) - \epsilon \sum_i^{|A|} \nabla_\theta (\pi_\theta(A_i, S_t) \ln \pi_\theta(A_i, S_t)) \right) \quad (2-16)$$

其中 $\pi_\theta(S_t, A_t)$ 表示的是在状态 S_t 下采取动作 A_t 的概率， ∇_θ 表示约束项关于

θ 的梯度， ef 为权重系数。该约束项本身与智能体推荐结果的多样性程度呈负相关，智能体的推荐策略越保守，推荐结果多样性程度越低，约束项造成的负反馈就越大，使得整体损失增大，由于智能体的学习目标是尽可能降低损失，因此增加了动态约束项的损失函数可以使得智能体的推荐策略趋向于获得未来奖励，从而优化推荐结果的多样性。

2.3.3 实验设置

本文提出的针对多样性优化的推荐算法致力于从用户与物品的交互记录中挖掘用户偏好，因此实验需要选用包含真实交互记录的公开数据集，同时由于强化学习渐进式交互学习的特性，数据集包含的数据样本数量越多，对算法的验证效果越好。这里实验评测数据集选用的是公开数据集 MovieLens 10M，其中包含了 71567 个用户对 10681 个电影物品的共计 10000054 条评分数据，其中每个评分范围介于 0 到 5 之间，评分粒度为 0.5，平均每个用户大约有 139 条评分记录。

对于本文提出的算法进行评测，不仅要考虑推荐任务的性能指标，还要结合强化学习交互式更新的特性考察算法在强化学习领域的性能表现。本文的评测方法并没有选用传统模型验证的训练集/测试集方法，而是采用强化学习领域典型的渐进式测试方法，即在智能体交互迭代过程中周期性地启动测试程序，在不同迭代阶段对智能体的性能表现进行评估，测试程序将随机抽取若干用户进行交互，并记录模型当前的表现。

在评测指标的选择上，选取了强化学习领域常用的平均奖励值衡量智能体在强化学习训练方面的效果，同时也采用推荐任务主流的准确率、召回率和 F1 值衡量智能体在推荐任务中的表现。具体地，当智能体在数据集上进行了 K 次迭代后，调用测试程序来对智能体当前的推荐策略进行评估，评估阶段智能体会与 k 个用户进行交互，每个用户与智能体交互 T 次，每次交互都将计算上述提到的四个评测指标：

(1) 平均奖励 (Average Reward)：智能体在 $k*T$ 次交互中获得的平均奖励；

(2) 准确率 (Precision@n)：对于每个用户 i 的交互时间步 t ，检查决策模型对于当前 S_t 给出的概率最高的 n 个动作，然后根据这些动作查询其中有多少项目的实际评分大于设定的阈值，其中符合条件的动作数占全部的动作数的比率就是获得的准确率；

(3) 召回率 (Recall@n)：对于每个用户 i 的交互时间步 t ，查询所有实际评分大于等于设定阈值的动作，然后通过决策模型查询对于当前状态 S_t 给出的

动作概率最高的 n 个动作，检查有多少符合条件的动作在这 n 个动作中，这个比率就是当前时间步的召回率；

(4) F1 值 (F1@n)：对于每个用户 i 的交互时间步 t，通过公式(2-17)来计算 F1 值。

$$F1@n = 2 * \frac{precision@n * recall@n}{precision@n + recall@n} \quad (2-17)$$

对于本文提出的针对多样性优化的推荐算法在 MovieLens 10M 数据集上的实验评估，模型训练参数设置如表 2-1 所示，所有的训练参数均通过数据集实验挑选得到。

表 2-1 模型参数设置

参数名	参数值
T (回合长度)	32
K (经验重放长度)	3000
d (状态表示维度)	20
k (动作嵌入维度)	10
l (奖励编码维度)	10
n (top-n)	32
γ (折扣因子)	0.99
a (归一化后的奖励下限)	-1
b (归一化后的奖励上限)	1

T 表示在一个回合 (Episode) 中智能体与用户交互的步数或时间步的数量，设置为 32；K 是经验重放 (Experience Replay) 中存储和回放的经验样本的数量，设置为 3000；d 表示状态 (State) 的观测维度数量，设置为 20；k 表示动作 (Action) 的嵌入维度数，设置为 10；l 表示奖励 (Reward) 的编码维度数，设置为 10；n 表示在推荐任务中，从所有可选的推荐项中选择排名靠前的 n 个物品作为候选项，设置为 32； γ 是一个介于 0 和 1 之间的折扣因子，用于衡量未来奖励的重要性，设置为 0.99；a 是奖励的归一化后的下限值，设置为 -1；b 是奖励的归一化后的上限值，设置为 1。

2.3.4 模型对比实验

本小节展示了在上述数据集、测试指标和训练参数的情况下，本文提出的针

对多样性优化的推荐算法与 baseline 的性能对比，这里选取了经典强化学习推荐算法 REINFORCE^[37]作为 baseline。图 2-7 展示了针对推荐多样性优化的强化学习推荐算法与 baseline 算法的在 MovieLens 10M 数据集上的对比测试结果。

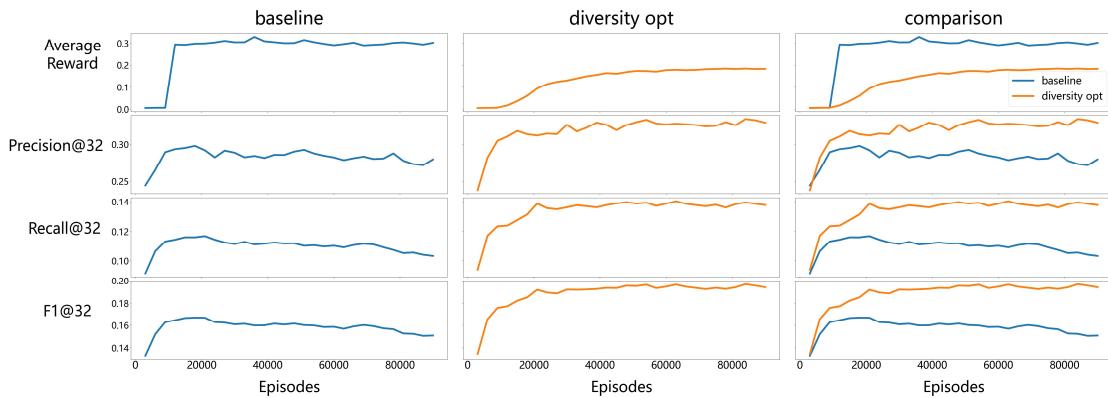


图 2-7 推荐多样性优化方法测试结果

从图中可以看出，相比 baseline 算法，本文提出的针对多样性优化的算法在准确率、召回率和 F1 值上分别有不同程度的提升。平均奖励这一指标降低的原因分析如下，优化后的算法在损失函数中增加了动态约束项，这一约束项会在智能体过度依赖即时奖励而做出重复推荐时导致较大损失，使得智能体对推荐策略进行调整，在即时奖励与长期奖励中取得平衡。而 baseline 算法并不具备动态约束项，可以看到智能体的平均奖励曲线有一个骤然升高并趋于稳定的趋势，这是因为 baseline 的智能体更倾向于获得即时奖励而不是长期奖励，为了获得高回报值而出现了重复推荐高奖励值物品的情况，因此增加了动态约束项的优化算法其推荐结果会具备更好的多样性。

表 2-2 更直观地展示了多样性优化算法相比于 baseline 的性能提升幅度，可以看到优化算法在具备更好的多样性的同时，准确率、召回率和 F1 值均优于 baseline 算法。

表 2-2 算法性能对比

Algorithm	AverageReward	Precision@32	Recall@32	F1@32
baseline	0.275	28.4%	11.0%	0.159
diversity opt.	0.154	32.5%	13.7%	0.193
Improv.	-44%	14.4%	24.5%	21.4%

2.3.5 消融实验

消融实验旨在探索算法中各个组成部分对整体性能的影响，并揭示关键因素和工作原理，是一种有力的研究方法，通过逐步削减或修改算法的特定组件或参数，控制其他参数和组件的影响，评估这些变化对算法性能的影响，加强对比实验结果的说服力。通过消融实验的设计和执行，期望能够深入分析不同组件对算法性能的影响，并获得关于算法的重要见解。本节将对本文提出的多样性优化的推荐算法进行消融实验，将算法中的改进组件进行去除，对比去除前后算法的性能表现，验证算法改进的有效性。

消融实验的数据集、测试方法、评价指标与前文描述一致，实验测试结果如表 2-3 所示。

表 2-3 算法消融实验

Algorithm	AverageReward	Precision@32	Recall@32	F1@32
rl	0.272	23.1%	8.4%	0.123
rl+DCT	0.154	32.5%	13.7%	0.193
Improv.	-43.4%	40.7%	63.1%	56.9%

从表中可以得知，算法在去除了动态约束项（Dynamic Constraint Term）后，准确率、召回率和 F1 值都有非常明显的下降，而平均奖励则有提升，通过分析控制台日志可知此时出现了重复推荐高回报物品而获得即时奖励的行为；而保留了动态约束项的算法在准确率、召回率和 F1 值上的表现均更优，且尚未出现重复推荐单一物品的问题。因此，本文针对多样性优化的算法改进是有效的。

2.4 样本效率优化的推荐算法

本小节将介绍一种针对强化学习样本效率优化的推荐算法，该算法针对智能体的策略更新流程，设计奖励预测模型优化智能体的交互更新过程，加快了模型的训练速度，提高了强化学习智能体训练的样本效率。

通常情况下，强化学习智能体与真实环境进行交互学习实现策略更新，但这种交互过程需要大量的样本和实验，存在试错成本高、环境交互受限的问题。本文提出的针对样本效率优化的推荐算法引入了奖励预测模型作为环境模型，通过真实的交互记录训练得到的奖励预测模型后，可用于模拟用户对推荐动作做出的

反馈奖励，同时结合 experience replay^[38,39]方法改进智能体的交互流程，使得智能体在与真实环境交互的同时，还可以与环境模型进行交互，降低试错成本，提高样本效率。

2.4.1 奖励预测模型设计

奖励预测模型在强化学习推荐中作为环境模型发挥着策略评估、改善样本效率等重要的作用，通过预测用户对推荐结果的反馈奖励，帮助智能体评估不同推荐策略的效果。

(1) 策略评估：奖励预测模型可用于评估不同的推荐策略的效果，智能体根据奖励预测模型输出的预测奖励值选择具有较高潜在奖励的推荐物品，从而优化推荐的性能。

(2) 改善样本效率：通过奖励预测模型，智能体可以根据预测的奖励值对策略进行学习和优化，而不必依赖实际的用户反馈，这样可以提高样本效率，减少对真实用户交互的依赖，降低试错成本，加速学习过程。

本文设计的奖励预测模型的结构如图 2-8 所示。

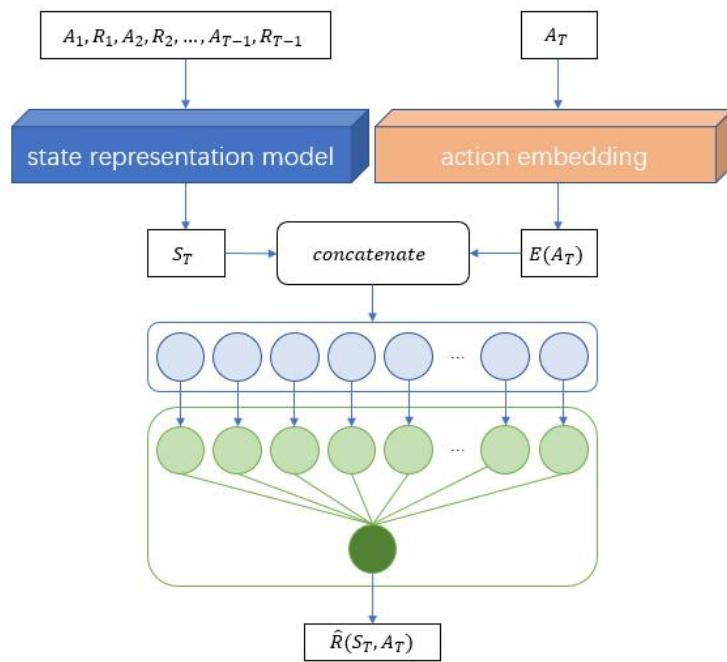


图 2-8 奖励预测模型结构

该模型以当前环境状态和推荐动作作为输入，并输出预测的奖励值，使得智能体可以通过与奖励预测模型进行交互，产生交互记录，完成策略更新。奖励预测模型可使用历史数据、用户行为模式、物品属性等进行训练，学习用户的兴趣

和偏好，预测用户对推荐物品的满意度。本文的奖励预测模型的训练在推荐流程的预训练阶段完成，通过真实的交互记录训练得到。

环境状态和智能体的推荐动作分别由状态表示模型和动作嵌入层进行处理，奖励预测模型在获得状态表示模型的环境状态后，根据环境状态和智能体做出的推荐动作，将模拟真实环境给出反馈奖励。

在 t 时刻，奖励预测模型接收状态表示模型输出的环境状态 S_t 和经过动作嵌入层处理后的推荐动作 A_t ，经过 `concatenate` 操作后，输入到由 w 参数化的全连接网络，最终得到预测的奖励值 \hat{R}_t 。

其中全连接网络的训练与状态表示模型的训练同时进行，全连接网络的参数以梯度反向传播的方式传播到状态表示模型中，网络参数 w 的更新如公式(2-18)所示。

$$w \leftarrow w - \alpha (\hat{R}_w(S_t, A_t) - R_t) \nabla_w \hat{R}_w(S_t, A_t) \quad (2-18)$$

其中 \hat{R}_w 表示由 w 参数化的奖励预测模型输出的预测奖励值， R_t 为交互记录中真实的奖励值， ∇_w 表示预测奖励对于 w 的梯度。

2.4.2 交互流程优化

本小节将介绍在加入奖励预测模型后，基于该模型结合经验回放机制对智能体交互推荐流程进行的优化改进。

通常情况下，应用于推荐任务的强化学习智能体使用状态表示模型从用户与智能体的交互记录中获取当前环境的状态表示，随后基于当前决策模型的推荐策略给出推荐动作，并等待用户对于该物品做出反馈，完成一轮交互式推荐。一个典型的强化学习交互流程如图 2-9 所示。但这种交互流程过度依赖用户的实时反馈，在智能体习得一个良好的推荐策略之前用户就可能已经失去了兴趣，这种交互方式的试错成本极高，显然在实际的推荐应用中是不可行的。

当引入奖励预测模型作为环境模型后，智能体可以同时与真实环境和模拟环境进行交互更新，通过适当增加环境模型在交互流程中的占比，可以提高样本效率，减少智能体对真实环境的依赖，降低试错成本，加速学习过程，增加了奖励预测模型的强化学习交互流程如图 2-10 所示。

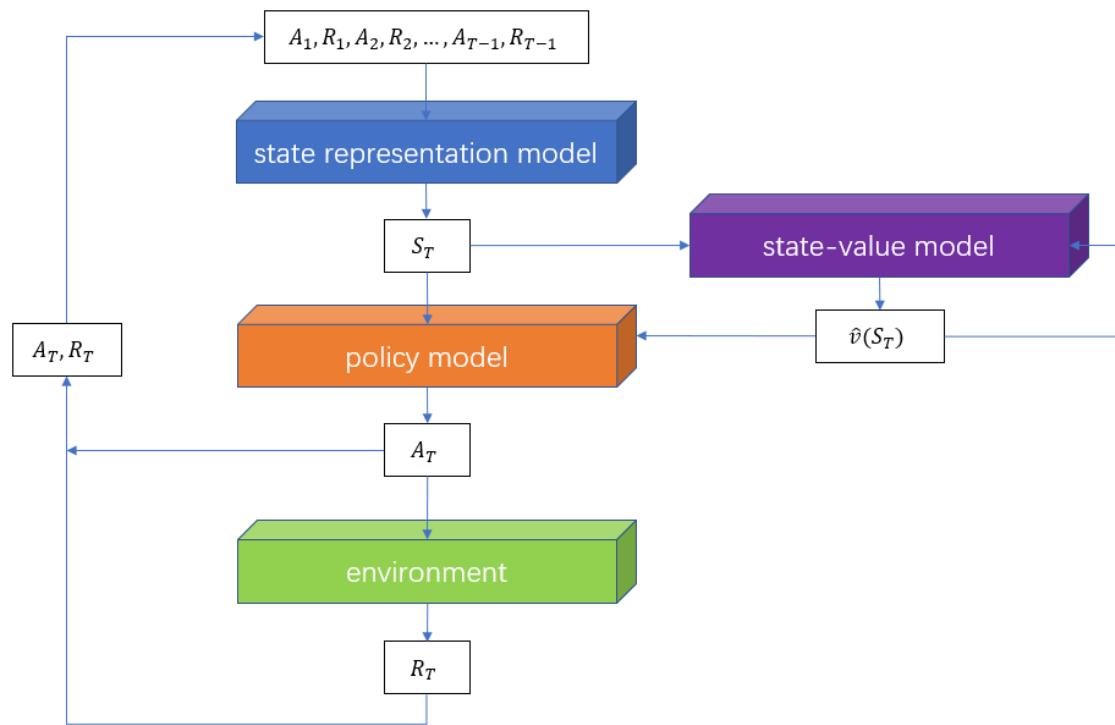


图 2-9 典型强化学习交互流程

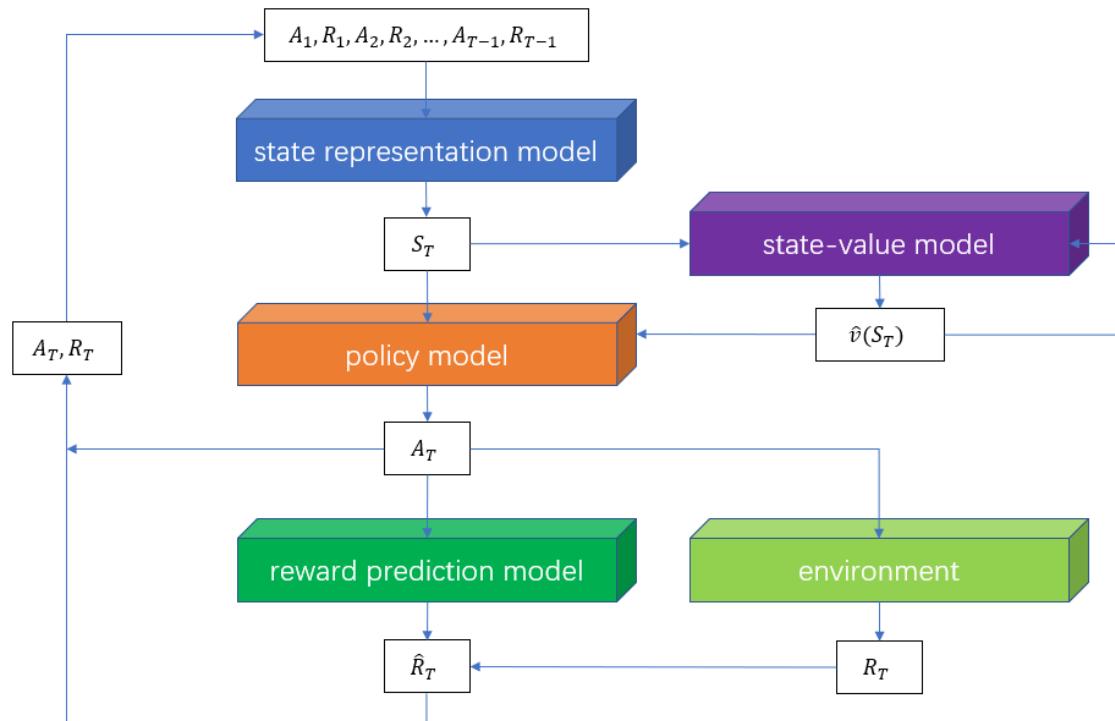


图 2-10 优化强化学习交互流程

如图所示，智能体不仅可以通过与真实环境的交互得到奖励值 R_t ，也可以通过奖励预测模型得到模拟的环境奖励值 \hat{R}_t ，在每次策略更新时，智能体从经验池中随机采样一批数据，并使用奖励预测模型模拟这些数据中的交互序列，得到一个预测的未来奖励序列，随后交替使用模拟的交互序列和真实的交互序列对行为网络进行训练更新。其中 $A_0, R_0, A_1, R_1, \dots, A_{t-1}, R_{t-1}$ 为智能体与环境的交互序列， $\hat{v}(S_t)$ 为 Critic（状态价值预测模型）基于智能体当前的交互序列做出的环境状态价值预测。通过使用经验回放方法和奖励预测模型，智能体可以利用先前的经验数据和与环境模型的交互记录加速其自身的学习过程，提高样本效率。

流程优化后的算法伪代码如表 2-4 所示。

表 2-4 强化学习推荐算法伪代码

```

1:   experience_temp ← an empty queue of max size M
2:   for 1 → M do
3:       user_id ← select a user id from 1 to num_users
4:       observation ← an empty list
5:       for 1 → N do
6:           action ← choose an action from the policy network from observation
7:           reward ← take action and receive feedback from this user
8:           append action, reward to observation
9:       end for
10:      append observation to experience_temp
11:   end for
12:   train the state representation model with experience_temp
13:   loop
14:       user_id ← select a user id from 1 to num_users
15:       observation ← an empty list
16:       for 1 → N do
17:           action ← choose an action from the policy network from observation
18:           reward ← take action and receive simulated feedback from this user
19:           invoke Actor-Critic on this transition
20:           append action, reward to observation
21:       end for
22:   end loop

```

本文提出的针对样本效率优化的推荐算法在实际应用中分为启动阶段和策略更新阶段。

对于智能体与用户交互的启动阶段，平台将为用户推荐资源类别中热度排名最高的资源作为冷启动物品，同时加载平台中已经过预训练的算法模型，等待用户对当前物品的反馈；之后，当用户对当前物品做出评分后，模型将接收到当前用户 ID 以及反馈的评分信息，并通过查询自身的策略网络为用户推荐新的物品；在推荐过程中， t 时刻的环境状态记作 S_t ，推荐系统在 t 时刻向用户推荐的物品记作 A_t ，用户对于该物品做出的评分动作记作 R_t ，推荐系统将与当前用户交互产生的 S_t, A_t, R_t 作为一条交互记录存储在缓存中，以供智能体的策略更新阶段使用。当用户结束推荐进程后，推荐系统将进入智能体推荐策略更新阶段。

对于智能体的推荐策略更新阶段，系统将采用先前储存的真实交互记录，并通过一步 Actor-Critic 算法对策略网络和状态表示模型进行更新；状态表示模型用于将用户与智能体的交互记录映射为当前环境状态，如 t 时刻的环境状态 S_t 就是状态表示模型对于数据流 $A_0, R_0, A_1, R_1, \dots, A_{t-1}, R_{t-1}$ 的输出结果。

2.4.3 实验设置

本文提出的针对样本效率优化的推荐算法，从用户与物品的交互记录中挖掘用户偏好，与前文的针对多样性优化的推荐算法类似，因此实验所使用的数据集选用公开数据集 MovieLens 10M，其中包含了 71567 个用户对 10681 个电影物品的共计 10000054 条评分数据，其中每个评分范围介于 0 到 5 之间，评分粒度为 0.5，平均每个用户大约有 139 条评分记录。

相应的，对于本文提出的算法进行评测，不仅要考虑推荐任务的性能指标，也要结合强化学习交互式更新的特性考察算法在强化学习领域的性能表现。本文的评测方法采用强化学习领域典型的渐进式测试方法，而不是传统模型验证的训练集/测试集方法，在智能体交互迭代过程中周期性地启动测试程序，对不同迭代阶段智能体的性能表现进行评估，测试程序将随机抽取若干用户进行交互，并记录模型当前的表现。

在评测指标的选择上，选取了强化学习领域常用的平均奖励值衡量智能体在强化学习训练方面的效果，同时也采用推荐任务主流的准确率、召回率和 F1 值衡量智能体在推荐任务中的表现。具体地，当智能体在数据集上进行了 K 次迭代后，将调用测试程序来对智能体当前的推荐策略进行评估，评估阶段智能体会与 k 个用户进行交互，每个用户与智能体交互 T 次，每次交互都将计算上述提到的四个评测指标：

(1) 平均奖励 (Average Reward) : 智能体在 $k*T$ 次交互中获得的平均奖励;

(2) 准确率 (Precision@n) : 对于每个用户 i 的交互时间步 t , 检查决策模型对于当前 S_t 给出的概率最高的 n 个动作, 然后根据这些动作查询其中有多少项目的实际评分大于设定的阈值, 其中符合条件的动作数占全部的动作数的比率就是获得的准确率;

(3) 召回率 (Recall@n) : 对于每个用户 i 的交互时间步 t , 查询所有实际评分大于等于设定阈值的动作, 然后通过决策模型查询对于当前状态 S_t 给出的动作概率最高的 n 个动作, 检查有多少符合条件的动作在这 n 个动作中, 这个比率就是当前时间步的召回率;

(4) F1 值 (F1@n) : 对于每个用户 i 的交互时间步 t 计算准确率和召回率的加权平均值。

对于本文提出的针对样本效率优化的推荐算法在 MovieLens 10M 数据集上的实验评估, 模型训练参数设置如表 2-5 所示。

表 2-5 模型参数设置

参数名	参数值
T (回合长度)	32
K (经验重放长度)	3000
d (状态表示维度)	20
k (动作嵌入维度)	10
l (奖励编码维度)	10
n (top-n)	32
γ (折扣因子)	0.99
p (规划步长)	5
a (归一化后的奖励下限)	-1
b (归一化后的奖励上限)	1

T 表示在一个回合中智能体与用户交互的步数或时间步的数量, 设置为 32; K 是经验重放中存储和回放的经验样本的数量, 设置为 3000; d 表示状态的观测维度数量, 设置为 20; k 表示动作的嵌入维度数, 设置为 10; l 表示奖励的编码维度数, 设置为 10; n 表示在推荐任务中, 从所有可选的推荐项中选择排名靠前的 n 个物品作为候选项, 设置为 32; γ 是一个介于 0 和 1 之间的折扣因子, 用于衡量未来奖励的重要性, 设置为 0.99; p 表示在一次规划过程中采取的迭代次

数，设置为 5； a 是奖励的归一化后的下限值，设置为 -1； b 是奖励的归一化后的上限值，设置为 1。上述训练参数均通过数据集实验挑选得到。

2.4.4 模型对比实验

本小节将本文提出的针对推荐效率优化的推荐算法在上述数据集、测试指标和训练参数的情况下，分别与 baseline 和针对多样性优化的推荐算法进行了性能对比。

图 2-11 展示了针对样本效率的优化算法与 baseline 算法在 MovieLens 10M 数据集上的对比测试结果，这里同样选取了经典强化学习推荐算法 REINFORCE^[37]作为 baseline 进行对比。

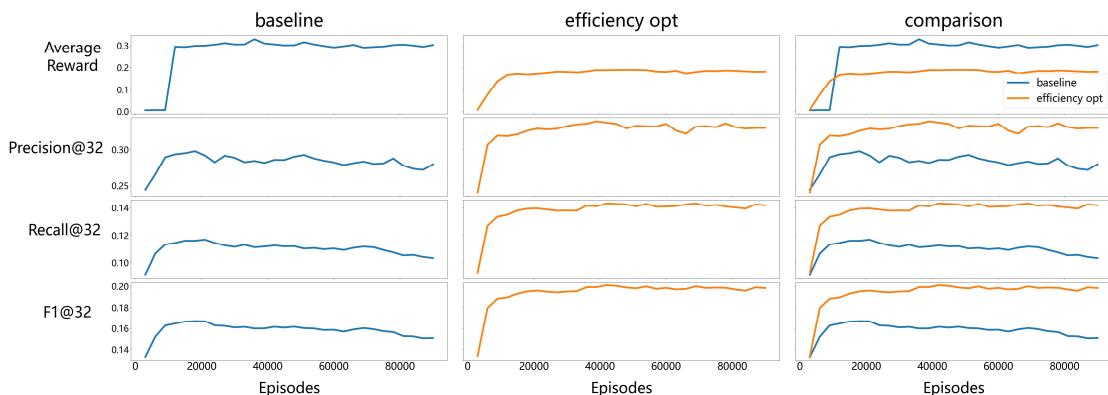


图 2-11 样本效率优化方法测试结果

从图中可知，本文提出的针对样本效率优化的算法在准确率、召回率和 F1 值方面相比 baseline 算法都有着不同程度的提升。此外，本文也将所提出的两种优化算法进行了对比，如图 2-12 所示。

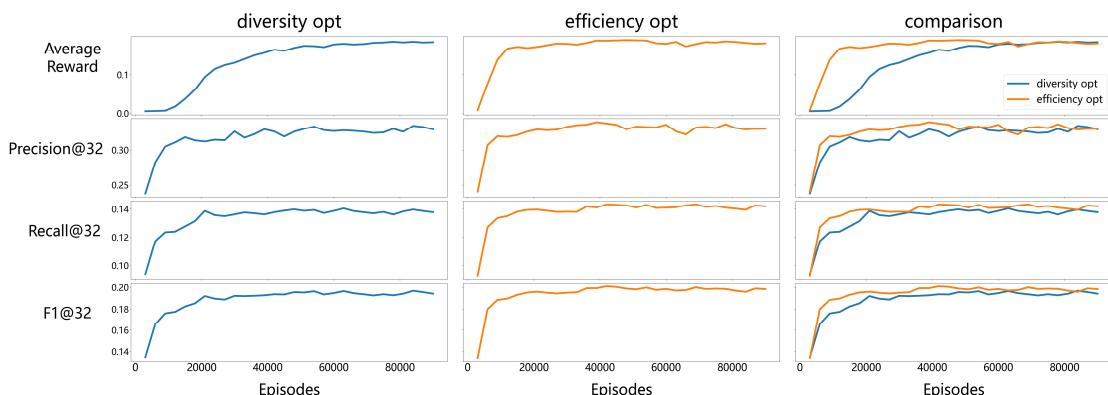


图 2-12 样本效率优化方法测试结果

根据图示可知，改进后的算法相比于未使用奖励预测模型的多样性优化算法在准确率、召回率和 F1 值上的表现有略微提升，而平均奖励曲线的上升速度有了大幅提升，这意味着算法的训练速度和样本效率获得了提升。实验结果分析如下，文中针对多样性优化的推荐算法受限于没有环境模型，智能体只能与真实环境交互学习更新，此时模型的训练效果将很大程度受限于样本数量和质量，而奖励预测模型在推荐任务中担任了环境模型的角色，对智能体的交互环境进行了补充，奖励预测模型通过真实的交互数据学习模拟出真实环境对于推荐动作的反馈，使得智能体的样本效率和训练速度获得了提升，因此增加了奖励预测模型的优化算法其平均奖励曲线上升速度会更快。

表 2-6 更直观地展示了 baseline 算法、多样性优化算法以及样本效率优化算法的性能对比，可以看到样本效率优化算法在具备更优的样本效率的同时，准确率、召回率和 F1 值均优于 baseline 算法，该算法智能体获得的平均奖励相比多样性优化算法也有着明显的提升。

表 2-6 算法性能对比

Algorithm	AverageReward	Precision@32	Recall@32	F1@32
baseline	0.275	28.4%	11.0%	0.159
diversity opt.	0.154	32.5%	13.7%	0.193
efficiency opt.	0.181	33.1%	14.1%	0.197
improv.	17.5%	16.5%	28.2%	23.9%

2.4.5 消融实验

本节将对本文提出的针对样本效率优化的推荐算法进行消融实验，将算法中的改进部分进行去除，对比去除前后算法的性能表现，验证算法改进的有效性。

消融实验的数据集、测试方法、评价指标与前文描述一致，实验测试结果如表 2-7 所示。

表 2-7 算法消融实验

Algorithm	AverageReward	Precision@32	Recall@32	F1@32
rl	0.145	32.9%	14.0%	0.196
rl+RPM	0.181	33.1%	14.1%	0.197
Improv.	24.8%	0.6%	0.7%	0.5%

对于针对样本效率优化的推荐算法，在将奖励预测模型（Reward Prediction Model）去除后，表现为平均奖励降低，而准确率、召回率和 F1 值有轻微的波动，这是因为去除了奖励预测模型的强化学习智能体仍可以通过有限的数据集进行学习更新，但训练速度和效果受限于样本质量。其中去除了奖励预测模型的算法智能体获得的平均奖励有着明显的降低，结合训练曲线分析可知，此时的模型训练速度较为缓慢。因此，本文针对样本效率优化的算法改进是有效的。

2.5 算法移植与部署

本文提出的两种优化算法在验证阶段使用了公开数据集，而本课题的研究目标是将研究算法应用到课程思政资源推荐中，因此需要对算法进行移植和部署。本节将从理论与实现两方面介绍本文算法的移植与部署工作，包含了泛化能力分析、预训练和交互更新。

2.5.1 泛化能力分析

泛化能力是机器学习和强化学习中一个重要的性能指标，它衡量了模型在未来验证的数据集上的表现能力，泛化能力的重要性主要体现在适应性、推广性以及鲁棒性。

(1) **适应性：**具备良好泛化能力的模型能够在未来面对新的、未见过的数据时具有较好的适应能力，产生更准确和可靠的预测和决策。

(2) **推广性：**具有良好泛化能力的模型可以更容易地应用于不同的任务和领域，可以将一个领域中学习到的知识和经验迁移到其他相关领域，从而加快新任务的学习速度和性能提升。

(3) **鲁棒性：**良好的泛化能力可以提高模型的稳定性，当模型的泛化能力较好时，即使在输入数据中存在一定的噪声、变化或不完整性，模型也能够产生一致和可靠的结果，有助于降低模型对数据的细微变化和噪声的敏感性，提高模型在真实场景中的鲁棒性和可靠性。

本文所提出的两种优化算法具有良好的泛化能力，理由如下：

(1) **算法具备适应性：**本文提出的优化算法中，模型的输入为用户与推荐系统交互产生的交互序列数据，其中包含的有效信息为物品 ID 和对于该物品用户给出的反馈值，而不涉及对物品内容本身的特征工程。当算法从公开数据集转移至平台的课程思政资源数据集时，在保证数据输入格式相同，数据集结构相似的情况下可以实现算法的移植，因此本文提出的优化算法具备一定的适应性。

(2) 算法具备推广性：本文提出的优化算法致力于从用户与推荐系统的交互记录中挖掘用户的兴趣偏好，对于课程思政资源推荐任务而言，由于课程思政的案例、文献、素材资源具备相似性，智能体通过用户的交互记录习得的用户对于某类资源的偏好也可以应用到其他类别的资源推荐中，因此本文提出的优化算法具备一定的推广性。

(3) 算法具备鲁棒性：本文提出的优化算法以用户的整条交互序列数据作为输入，该交互序列是用户对多个推荐物品进行的一系列的评价，交互序列越长，受单一数据扰动的影响就越低；同时任意长度的序列数据会先经过状态表示模型处理成固定长度的状态表示向量，再输入到智能体的决策模型中，这进一步降低了因数据变化导致结果偏离的可能性，因此本文提出的优化算法具备一定的鲁棒性。

综上所述，本文提出的优化算法具备一定的适应性、推广性和鲁棒性，整体表现出了良好的泛化能力，可以完成后续的课程思政资源推荐任务。

2.5.2 模型预训练

预训练阶段是模型的冷启动阶段，也是将算法应用到实际任务中必须要有的阶段，目的是通过训练得到一个可用的奖励预测模型，其流程示意如图 2-13 所示。

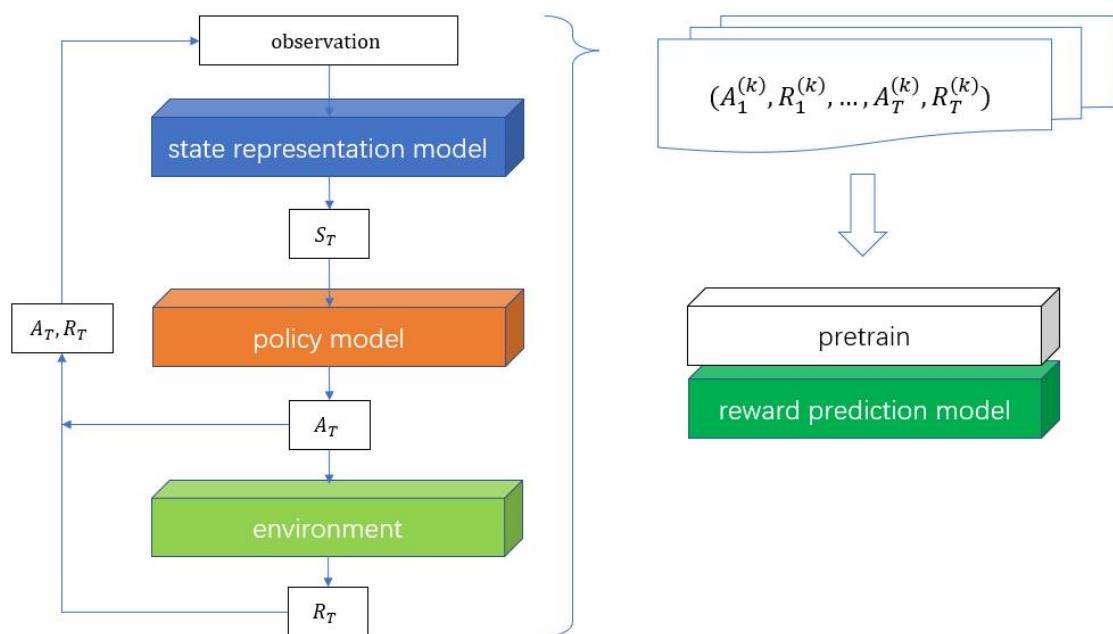


图 2-13 预训练流程示意图

初始状态下，智能体使用随机的推荐策略与真实环境的 k 个用户进行 $k*T$ 次交互，这里可以基于真实的交互记录构建用户评分表替代真实环境，当智能体做出推荐动作后，通过检索用户评分表返回该用户对于当前物品的历史评分。交互完成后，将交互产生的交互记录按照用户 ID 进行分组，即分成 k 组数据，每组数据包含了对于当前用户的 T 次交互记录。之后使用这些分组数据，分批次对奖励预测模型进行训练，使用 MSE 损失函数和 RMSProp 优化器调整模型的网络参数，从而得到一个可用的奖励预测模型。

2.5.3 交互更新

交互更新阶段是智能体执行推荐任务的阶段，在得到可用的奖励预测模型后，智能体同时与真实和模拟环境的进行交互，在交互中对自身推荐策略完成学习更新，交互更新阶段的示意如图 2-14 所示。

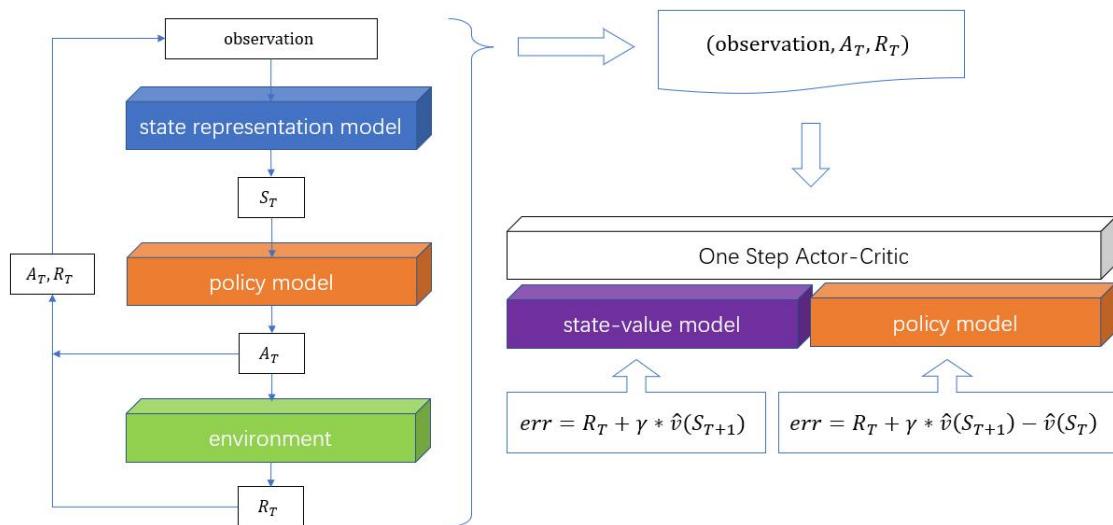


图 2-14 交互更新流程示意图

在每一个交互时间步，智能体与环境的交互将产生交互数据 (ob, A_T, R_T) ，其中 ob 表示在 T 时间步下当前的环境状态。智能体在每轮交互结束时，都将使用交互数据 (ob, A_T, R_T) 通过一步 Actor-Critic 的方法对决策模型以及状态价值模型进行训练更新，其中决策模型和状态价值模型的参数训练均采用 SGD 优化器进行，不同的是状态价值模型采用 MSE 作为损失函数，而决策模型的损失函数为本文 2.3.2 节中公式(2-16)所示。

2.6 本章小结

本章主要介绍了针对深度强化学习推荐算法的研究。首先介绍了所研究算法涉及的相关理论基础，然后通过问题陈述引出了算法设计的出发点，随后分别介绍了本文提出的针对多样性优化的推荐算法和针对样本效率优化的推荐算法，其中详细介绍了两种优化算法的模型设计与改进，并通过设置对比试验和消融实验验证了算法改进的有效性，最后对算法的移植与部署工作进行了阐述，可知本文提出的优化算法可适用于课程思政资源推荐任务。

第三章 平台设计

3.1 需求分析

本节分别从平台的整体需求、角色划分、功能需求和非功能需求进行了分析。

(1) 平台整体需求

随着高校课程思政教学的不断推进，越来越多的教师开始投入到课程思政建设中去，但目前课程思政教学存在参考案例短缺、课程思政元素设计过渡生硬等问题，一定程度上阻碍了课程思政教学。本平台针对课程思政目前存在的问题，借助成熟的项目开发技术和高效的深度强化学习技术为教师打造一个集成了资源共享、互动交流的平台。需求分析在整个项目开发周期中有着举足轻重的地位，决定了之后平台的设计、开发以及测试环节的方向，是软件开发非常重要的环节 [40]。

从教师的需求来看，教师在教学往往难以找到与课程知识点相契合的思政素材，因此通过一个课程思政资源管理平台来汇聚课程思政的教育资源对于教师的思政教学是很有帮助的。教师可以通过本平台检索自己需要的课程思政素材，并将自己的教学经验与平台中的其他教师用户进行交流讨论，不仅提高了教学效率，也推动了课程思政的发展建设。

从学生的需求来看，当今移动互联网飞速发展，年轻人每天接触到的信息量是巨大的，加强思政建设势在必行。通过将思政元素融入到专业课教学中，让学生在专业课的学习中获得思政教学，既为专业课增添了几分趣味，又避免了思政课说教导致的教学效果下降。部分教师在课程思政教学工作中，专业课知识与思政元素过渡生硬，或将二者分割开单独进行讲授，导致课程思政失去了其原本意义，影响教学效果。如果学生可以接触到优秀的课程思政教学，不仅可以减轻课程的学习压力，也能够在耳濡目染中完成思政的熏陶。

(2) 平台角色划分

平台根据使用者权限不同分为游客、普通用户、教师用户、管理员和超级管理员。游客用户可以进行浏览，无法与平台资源或平台用户产生交互。普通用户是通过平台的注册功能完成注册，但尚未完善个人信息的用户。平台对普通用户仅开放少量与资源交互的功能，如浏览、点赞、评论，但无法与其他用户进行交互。教师用户即完善了个人信息的普通用户，可使用平台中为教师提供的所有功

能，可以与平台资源和平台用户进行交互。管理员具有平台资源、公告的完全管理权限，以及对注册用户的部分管理权限。超级管理员则具有平台中所有功能的管理权限，通常为平台的实际控制人员。各角色拥有的权限如表 3-1 所示，除超级管理员外，其他角色的所有操作均通过 Web 客户端实现。

表 3-1 平台角色权限表

角色	浏览	点赞/评论	发布	管理	后台
游客	√				
普通用户	√	√			
教师用户	√	√	√		
管理员	√	√	√	√	
超级管理员	√	√	√	√	√

(3) 功能需求分析

平台功能模块主要分为用户模块、课程思政模块、推荐模块和管理员模块。

用户模块主要为注册用户提供与用户相关功能，如注册登录、修改密码、个人信息等，平台会根据登录信息自动识别当前用户的身份，区分普通用户与管理员用户。

课程思政模块是平台的关键部分，根据课程思政资源类型不同分为四大板块：文献信息、案例设计、素材资源和交流讨论，用户在平台中发布的课程思政资源会根据分类归纳到对应的分区中。

推荐模块的功能建立在课程思政模块之上，同时又独立于课程思政模块。用户可以在课程思政模块对应的板块中向后端发起推荐请求，中央调度器会将请求参数传递给推荐模块，并将推荐模块的推荐结果返回给前端加载。由于推荐模块与课程思政模块呈现低耦合的关系，二者可以分别部署于不同服务器，既可以按照功能需求的差异对服务器硬件进行调整，降低了部署成本，也可以实现服务器冗余，提高平台的整体运行稳定性。

管理员模块包含了一系列平台管理相关功能，只有管理员和超级管理员账号才可以使用管理员模块，管理员模块包含了用户管理、内容管理、公告管理、通知管理等功能。

(4) 非功能需求分析

非功能需求包括易用性需求、稳定性需求、扩展性需求和安全性需求。

① 易用性需求

平台在初期阶段需要吸引新用户，如果平台功能设计过于复杂，或功能页面

层级较深，会严重影响平台用户数量，因此平台的页面设计要清晰明了，流程要简便易于操作，以具备较好的易用性。

② 稳定性需求

平台服务于教师课程思政设计需求，需要满足教师的登录访问需求，满足管理员 24 小时的登录管理需求，能够承担百人级别的并发请求，保证平台的稳定运行，因此需要在技术选择、代码编写方面进行考量。

③ 扩展性需求

随着平台用户数量的增加，未来有可能出现新的用户需求，平台的架构设计需要考虑到未来可能会增加的功能需求。

④ 安全性需求

平台用户为校园局域网内所有具有课程思政教学需求的教师，预计平台的注册用户数量较多，需要具备一定的流量承载能力，同时也要预防用户越权等危及平台数据安全性的问题。

3.2 平台架构设计

(1) 逻辑架构

本平台是课程思政平台、资源推荐系统和数据库系统的结合体，整体逻辑架构分为表示层、业务层、数据层三层结构。

① 表示层

表示层主要为前端页面，负责向平台的使用者提供界面显示，通过图形化的交互界面展现可操作的功能。表示层与业务层相关联，表示层可以将使用者的请求操作发送给业务层，业务层收到表示层发来的业务请求后会根据业务类别进行进一步处理，并将处理结果反馈给表示层。

② 业务层

业务层负责实现平台的相关业务功能，包含了拦截器、中央调度器、业务对象和推荐系统。业务层同时与表示层和数据层相关联，是协调三层架构的关键部分，业务层在收到表示层传来的业务请求后，会先由拦截器来对请求进行核验，过滤掉非法请求，核验通过后请求到达中央调度器，中央调度器根据请求类别和参数，调用业务对象或推荐系统进行业务处理，并将处理完成后的结果返回给表示层。

③ 数据层

数据层主要为业务层提供数据访问操作，分为数据访问对象和数据库系统两

部分。当数据层接收到业务层发来的数据访问请求后，数据访问对象会根据参数和访问类型对数据库进行访问操作，并将操作结果返回业务层进行下一步处理。

视图层提供交互功能，业务层提供执行逻辑，数据层提供数据访问，三层架构相互协作，共同完成平台的各种功能。

(2) 技术架构

技术架构是根据逻辑架构进行的技术细分，本平台的技术架构如图 3-1 所示，其包括了视图层、调度层、服务层、推荐系统和持久层。

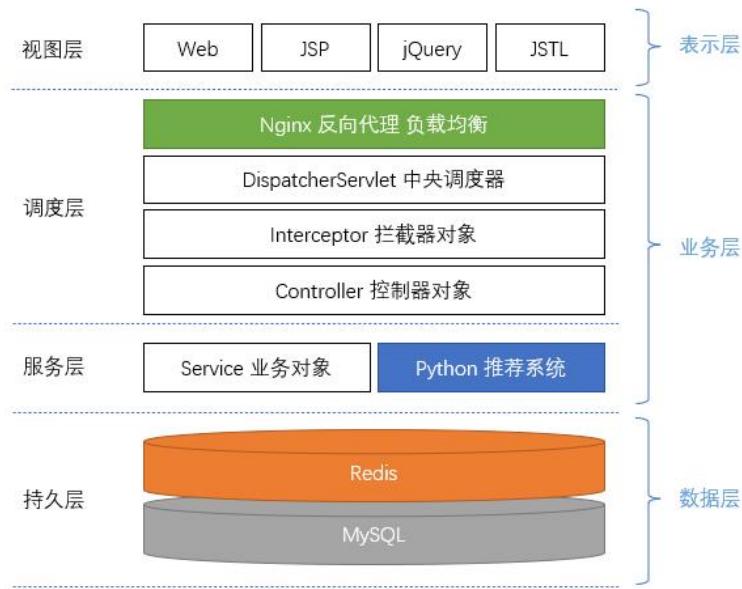


图 3-1 平台技术架构

① 视图层

视图层主要为前端 Web 页面，通过 JSP + JSTL 配合 Layui 组件库实现视图层用户界面开发。JSP 是一种基于 Java 开发的动态网页技术，实现了动态内容与静态页面的分离^[41]，并具备跨平台、响应速度快等优势；JSTL 是 JSP 的通用函数库^[42]，其封装了 JSP 的通用核心功能，增强了代码的维护性；Layui 是一套开源的 Web UI 组件库，采用模块化设计规范，遵循原生 Web 开发方式，可以加速项目开发速度。视图层选用 JSP 作为页面文件，使用 JSTL 替代原生 Java 代码完成数据操作，使用 Layui 组件库提供的组件对 UI 进行设计。

② 调度层

调度层由 SpringMVC 的中央调度器 DispatcherServlet、拦截器对象和控制器对象组成，负责对视图层发来的业务请求进行统一调度、检查和处理。DispatcherServlet 是 SpringMVC 的核心 Servlet，开发者使用注解的方式在容器中

创建控制器对象，可替代原有的通过 Servlet 实现的控制器对象。拦截器对象根据处理结果对请求进行拦截或放行，其实现了 HandlerInterceptor 接口中的 preHandle、postHandle 和 afterCompletion 方法，本平台的拦截器主要用于检查用户登录状态和确认用户角色权限，因此侧重于 preHandle 方法的编写。

③ 服务层

服务层由 Spring 容器中多个高内聚、低耦合的业务对象组成。每个业务对象仅负责实现一小部分独立的功能，调度层的控制器对象通过组合调度多个业务对象，实现对复杂业务的处理。平台与推荐系统的整合在服务层实现。

④ 推荐系统

推荐系统由 Python 语言编写，使用 TensorFlow、Keras 框架实现模型的训练与保存，可单独部署在配有 CUDA 计算能力的服务器上，推荐系统与服务层通过 Socket 进行局域网范围内的通信。当用户发起推荐请求时，服务层业务对象将与推荐系统通过 Socket 建立 TCP 链接，将推荐请求发送至推荐系统，并等待接收推荐系统产生的推荐结果，当用户结束请求或 Web 客户端意外关闭时，终止连接并释放资源。

⑤ 持久层

持久层包含 MyBatis 的 DAO 代理、Redis 和 MySQL 数据库组成，DAO 代理为服务层提供 MySQL 数据库访问操作，将 SQL 语句从代码中分离出来，单独存放在 Mapper 文件中，降低了代码的耦合度，可实现良好的复用性、扩展性和可维护性。

3.3 功能模块设计

本课程思政平台模块主要有注册登录、个人中心、点赞评论、资源上传、资源推荐和平台管理模块。前文通过分析平台的整体需求、模块需求以及非功能性需求，确定了平台的服务对象以及架构设计，本节是将平台的功能模块设计继续细化，确定具体功能和业务流程。

(1) 注册与登录

注册登录模块为用户和管理员提供注册登录、登录保持功能。用户可以通过点击导航栏主动跳转到登录页面，也可以在未登录的情况下由平台跳转到登录页面。用户发起业务请求时，平台的调度层的拦截器将会检查用户当前的登录状态，为未登录的用户跳转到登录页面，登录页面集成了注册功能，采用选项卡的形式进行功能切换。用户在登录界面可以选择使用登录保持功能，该功能将为平台的

活跃用户提供无感的登录状态保持，期间用户无需重复进行登录操作，若用户长时间未访问平台则注销登录状态。同时平台将根据用户的登录信息，主动辨别普通用户与管理员用户，根据用户角色跳转到对应的个人中心页面。

(2) 点赞评论

登录用户在完善了个人资料信息后，可以在浏览平台的课程思政相关资源时进行点赞、评论操作，并在操作成功后获得弹窗提示，平台将针对单个资源对点赞和评论数量进行统计，并显示到资源页面中，同时用户可以在个人中心查看自己的点赞记录和评论详情。以点赞功能为例，其逻辑流程图如图 3-2 所示。

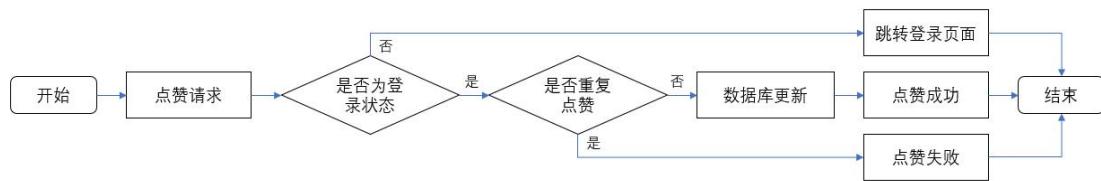


图 3-2 点赞功能逻辑流程图

(3) 资源上传

用户的个人中心页面提供资源上传功能，按照平台课程思政资源类型分别设计了课程思政案例上传、思政素材上传和教学话题发布三个功能入口。教师用户点击对应入口即可进入资源上传页面，页面中主要包含资源分类选择框、标题输入框、富文本编辑器和评论开关按钮，其中富文本编辑器可以实现字体调整、布局调整、表情插入、链接插入和图片上传，可以基本满足教师的资源共享需求。资源上传逻辑流程图如图 3-3 所示。

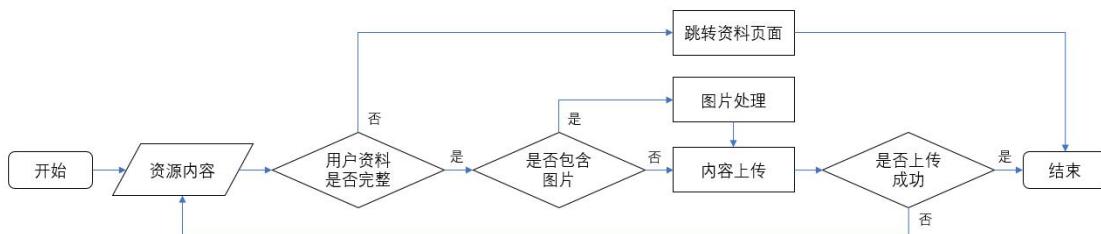


图 3-3 资源上传逻辑流程图

(4) 个人中心

根据用户角色不同，个人中心可分为普通用户的个人中心页面和管理员用户的管理中心页面。个人中心页面为普通用户提供个人资料修改、点赞评论记录查看、课程思政资源发布、消息通知和退出登录功能；管理中心页面是在个人中心页面的基础上，为管理员用户增加了平台相关的管理功能。个人中心访问逻辑流程图如图 3-4 所示。

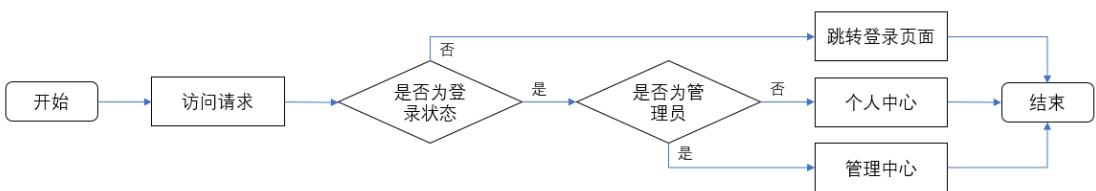


图 3-4 访问个人中心逻辑流程图

(5) 平台管理

平台为管理员用户提供平台管理功能，管理员可使用管理员账户登录访问管理中心页面，在管理中心可对用户登录、用户资料、用户通知、资源内容、点赞评论和平台公告进行管理。管理员账户不可自行创建，需要由平台的实际管理者在后台手动创建，以增强平台的安全性。平台管理功能模块如图 3-5 所示。

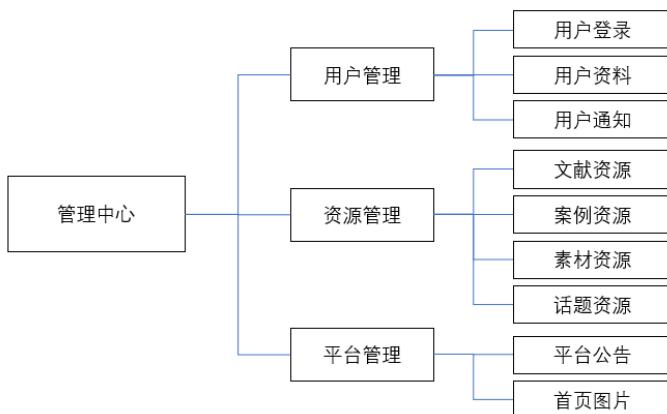


图 3-5 管理功能模块图

(6) 资源推荐

平台为用户提供课程思政文献资源的推荐服务，教师用户可通过资源首页的推荐功能入口使用平台的资源推荐功能。用户的推荐请求将经过调度层到达服务层，再由服务层对应的业务对象发送至推荐系统，请求中携带了当前用户以及当前浏览资源的部分信息，推荐系统在接收到推荐请求后，会根据用户信息和资源信息调用训练好的深度强化学习模型，生成推荐结果并将推荐结果返回到服务层，服务层对推荐结果进行处理、封装，最终由调度层将推荐结果返回给视图层。资源推荐的逻辑流程图如图 3-6 所示。

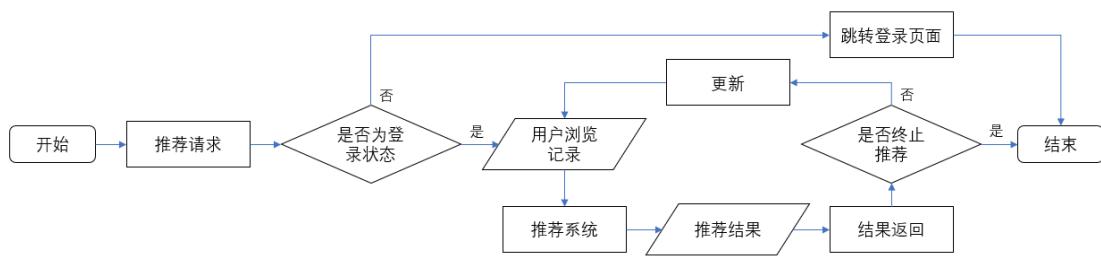


图 3-6 推荐功能逻辑流程图

3.4 数据库设计

经过需求分析、架构设计和功能设计，基本确定了平台的整体设计。本小节将分析平台中的实体关系，完成平台数据库的设计。

(1) 数据库 E-R 模型

数据库设计的关键是将实体与属性、实体与实体之间的关系考虑清楚，因为随着项目开发工作的进行，越靠近项目后期更改数据库表结构的代价就越高，数据库设计要分析数据关系，减少数据冗余^[43]。梳理数据库实体关系模型是数据库设计的第一步，后续数据库的详细设计需要实体关系模型作为支撑，本节从平台所依赖的主要实体及其关系进行设计和阐述，先对各个实体及其属性进行分析，再将这些实体组合起来，形成整个平台的实体关系模型。

根据平台需求分析和功能模块设计，可以抽象出的主要实体有：用户、管理员、文献、案例、素材和公告。

- ① 用户实体，用于存储用户的基本信息，其中包括用户的系统 ID、用户名、账户、密码、邮箱以及个人资料。
- ② 管理员实体，用于存储管理员的相关信息，包括管理员的系统 ID、用户 ID、用户名和管理等级。
- ③ 文献实体，即平台中课程思政相关文献信息，包括标题、作者、摘要、关键词、文献类型和发表时间。
- ④ 案例实体，即平台中课程思政案例设计信息，包括作者信息、学科类别、标题和内容。
- ⑤ 素材实体，即平台中思政元素素材信息，包括用户信息、所属类别、标题和内容。
- ⑥ 公告实体，用于存储平台发布的公告信息，包括公告编号、管理员信息、标题和内容。

最后，分析抽象实体之间的关联方式，将抽象实体组合为一个整体，即为数据库的整体 E-R 模型，如图 3-7 所示。

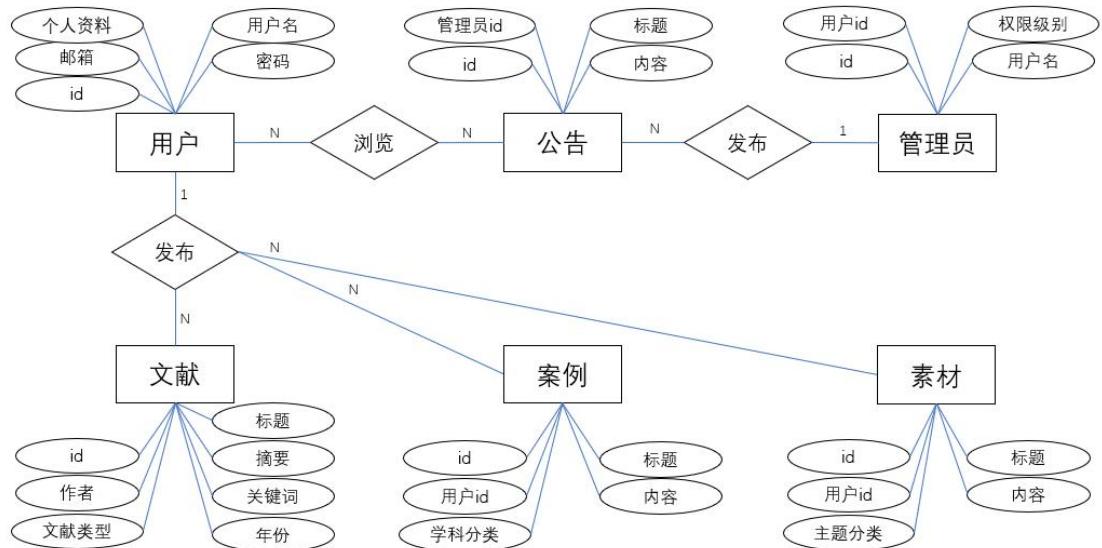


图 3-7 数据库 E-R 图

(2) 数据库表结构

本小节将根据数据库 E-R 模型设计，对所涉及的实体进行字段名、字段类型、长度和描述进行详细设计。

表 3-2 用户登录表

序号	字段名	字段类型	长度	描述
1	user_id	int	default	联合主键，自增
2	user_name	varchar	default	联合主键
3	user_pwd	varchar	default	
4	user_email	varchar	default	

表 3-3 用户资料表

序号	字段名	字段类型	长度	描述
1	profile_id	int	11	主键，自增
2	user_name	varchar	default	
3	actual_name	varchar	default	
4	gender	varchar	default	

第三章 平台设计

续表 3-3 用户资料表

序号	字段名	字段类型	长度	描述
5	university	varchar	default	
6	title	varchar	default	
7	signature	varchar	default	
8	avatar_path	varchar	255	用户头像路径

表 3-4 管理员表

序号	字段名	字段类型	长度	描述
1	admin_id	int	11	主键
2	level	int	2	权限等级
3	user_id	int	11	
4	user_name	varchar	default	

表 3-5 文献资源表

序号	字段名	字段类型	长度	描述
1	id	int	11	主键, 自增
2	title	varchar	default	
3	abstract	text	default	
4	author	varchar	default	
5	department	varchar	default	
6	keyword	varchar	default	
7	journal	varchar	default	
8	date	date	20	发表时间

表 3-6 案例资源表

序号	字段名	字段类型	长度	描述
1	id	int	11	主键, 自增
2	user_name	varchar	default	用户名
3	category_1	varchar	20	一级分类

第三章 平台设计

续表 3-6 案例资源表

序号	字段名	字段类型	长度	描述
4	category_2	varchar	30	二级分类
5	title	varchar	default	案例标题
6	content	text	default	案例内容
7	date	datetime	default	发布时间
8	is_available	int	1	可用标志位

表 3-7 素材资源表

序号	字段名	字段类型	长度	描述
1	id	int	11	主键, 自增
2	user_name	varchar	default	
3	category_1	varchar	20	
4	category_2	varchar	30	
5	title	varchar	default	素材标题
6	content	text	default	素材内容
7	date	datetime	default	
8	is_available	int	1	可用标志位

表 3-8 公告表

序号	字段名	字段类型	长度	描述
1	id	int	11	主键, 自增
2	user_name	varchar	default	管理员用户名
3	title	varchar	default	公告标题
4	content	text	default	公告内容
5	date	datetime	default	发布时间
6	visit_count	int	11	访问数量
7	is_available	int	1	可用标志位

3.5 本章小结

本章对平台的整体设计进行了梳理，首先进行了需求分析，然后从逻辑架构和技术架构两方面对平台架构进行了设计，随后将平台的主要功能进行拆分，并对它们进行逐个分析设计，最后根据平台的功能需求，设计了数据库的 E-R 模型以及实体类的数据库表结构。通过系统性的梳理，为后续的平台开发工作指明了方向。

第四章 平台实现

4.1 相关开发技术

(1) Spring

Spring 是一种轻量级的开源 Java 企业级框架，旨在替代 EJB 等重量级 Java 企业框架。经过多年发展，Spring 已经发展成为一个庞大的框架系统，其中包括 Spring Security、Spring Boot 等重要组件^[44]。Spring 框架的核心技术包括控制反转（IOC）和面向切面编程（AOP）。Spring 的 IOC 容器可以自动地实例化、装配和管理 Java 对象，使得应用程序可以解耦和减少代码冗余，而 AOP 则提供了一种可插拔式的编程方式，可以对多个业务逻辑进行统一的管理和横向扩展。

控制反转 IOC，将以往由开发者管理的对象转移给代码之外的容器，即反转对象的控制权，由容器完成对对象的管理与控制，开发者在编写代码时仅需提供使用的对象名称就可以从容器中获取相应的对象，从而降低代码的耦合度，增强了项目的灵活性。Spring 框架使用依赖注入 DI (Dependency Injection) 实现控制反转。面向切面编程 AOP 可以将一些重复性高的非业务功能模块化，当原有业务需要功能增强时，面向切面编程可以在不变动原有业务逻辑的情况下，在程序执行期间生成代理对象，通过代理对象执行业务方法进行功能增强，从而实现业务功能与非业务功能的耦合。AspectJ 是一种面向切面编程的实现。

使用 Spring 可以使开发者更专注于程序的功能开发，而不必过多在意繁杂的属性配置，同时 Spring 对各种优秀的框架提供直接支持。

(2) SpringMVC

SpringMVC 是 Spring 框架的一个重要组成部分，是一种轻量级的 Web 框架。它基于 MVC (Model-View-Controller) 设计模式，可以帮助开发人员轻松地开发 Web 应用程序。同时 SpringMVC 提供了方便而强大的前后端分离支持，帮助开发人员更好地实现 Web 应用程序的前端和后端的分离。SpringMVC 的核心组件是中央调度器 (DispatcherServlet)，负责接收和管控 Web 请求，返回请求处理的结果，以及管理其生命周期^[44]。MVC 即 Model、View 和 Controller，遵循 MVC 规范设计的程序可以达到高内聚低耦合的效果。

SpringMVC 的工作原理如图 4-1 所示。

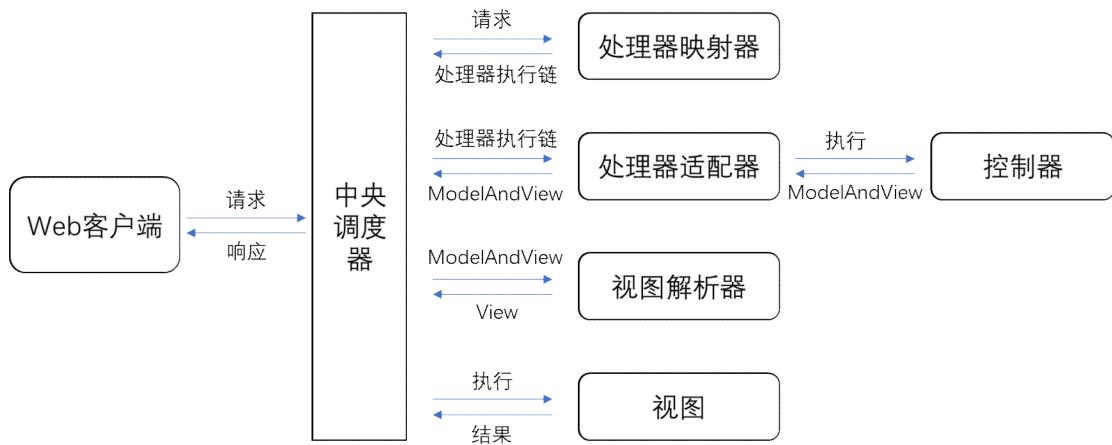


图 4-1 SpringMVC 原理图

SpringMVC 的具体工作原理如下：

- ① Web 客户端发送业务请求，由中央调度器（DispatcherServlet）接收并处理；
- ② 中央调度器根据请求检索处理器映射器（HandlerMapping），查询请求路径和控制器之间的对应关系；
- ③ 处理器映射器返回一个执行链给到中央调度器；
- ④ 中央调度器向处理器适配器（HandlerAdapter）发送执行 Handler 请求；
- ⑤ Handler 处理器去执行相应的逻辑，并返回 ModelAndView 中处理完的结果数据；
- ⑥ 处理器适配器将 ModelAndView 返回给中央调度器；
- ⑦ 中央调度器向视图解析器（ViewResolver）发送视图解析请求；
- ⑧ 视图解析器将试图解析结果返回给中央调度器；
- ⑨ 中央调度器进行视图渲染，将模型数据填充到请求作用域；
- ⑩ 中央调度器将响应发送给 Web 客户端，最终解析后显示出视图。

(3) MyBatis

MyBatis 是一种优秀的数据持久层框架，可用于操作数据库并实现增删改查功能。该框架采用了基于 XML 或注解的配置方式，同时支持多种数据源和动态 SQL 语句的生成，为开发人员提供了更高效、灵活的数据访问解决方案。MyBatis 封装了大量的 JDBC 操作，并且直接使用 Java 对象表示数据结果，大大降低了开发者使用 JDBC 的复杂性，使得开发者可以专注于 SQL 语句本身。MyBatis 使用灵活，对数据库的操作发生变动时只需将 SQL 语句写在 xml 文件中，不会

破坏现有的程序逻辑，也无需改变数据库设计。通过 DAO 层将业务逻辑代码和数据库访问操作进行分离，实现了业务代码和 SQL 语句之间的解耦，使得程序轻便简洁、易于维护测试。MyBatis 实现流程如图 4-2 所示。

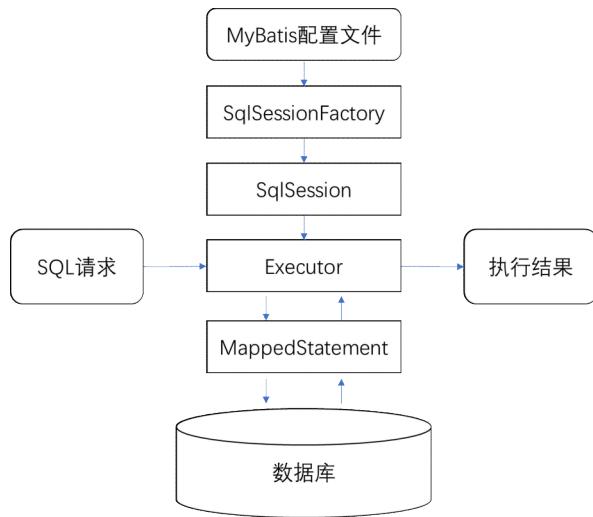


图 4-2 MyBatis 原理图

MyBatis 的实现原理如下：

- ① 加载 MyBatis 配置文件；
- ② 使用 SqlSessionFactory 创建 SqlSession 对象；
- ③ 通过 SqlSession 对象 Executor 的 execute 方法执行 SQL 语句；
- ④ 使用 MappedStatement 对象将执行结果进行转换，维护一条 CRUD 节点的封装，得到最终的处理结果。

(4) JWT

JWT 是 Token 的一种实现，用于实现平台的登录保持功能。JWT 将 Token 经服务器加密后存放在用户的 Web 客户端中，相比于传统的 Cookie 方法和 Session 方法，JWT 在具备良好的安全性的同时也解决了服务器存储资源占用的问题。JWT 由三部分组成，分别是头部、装载内容和签名，其中头部信息用于声明生成签名的算法，装载内容包含了一些特定的数据，如用户的登录信息、有效期等，签名部分的组成是 JWT 安全性的关键，其是由服务器密钥、头部和负载各自的 Base64 编码三部分经过算法运算得到的。由于 JWT 的三个部分相互关联，避免了 Token 被篡改的风险，同时 JWT 的加密密钥存放在服务器端，因此即使用户的 Token 被泄露也难以解密。

(5) Nginx 和 Tomcat

Nginx 是由 Igor Sysoev 发布的一款高性能的开源 Web 服务器，具有轻量级、

高性能、可扩展等特点^[45]。其设计目标是具备处理高并发连接的能力，相比于传统的基于线程或基于进程的服务器，Nginx 采用了事件驱动、非阻塞的架构，可以高效地处理连接同时保持较低的资源消耗，在实际应用中 Nginx 常用于负载均衡和反向代理。负载均衡即把平台收到的业务请求分摊到多个物理服务器上处理，避免由于单一服务器过载导致宕机，Nginx 支持多种负载均衡算法，包括轮询、IP 哈希、最少连接等，运维人员可以根据实际的需求来配置最适合的负载均衡策略，同时 Nginx 还可以为 Tomcat 服务器集群实现反向代理，配合负载均衡算法能够确保任务被分发到负载较轻的服务器中。

Tomcat 是由 Apache 开发的一种开源的 Web 应用服务器软件。随着多年的发展，Tomcat 已经成为一种同时满足 JSP 和 Servlet 规范以及商业 Java Servlet 特性的 Web 应用服务器，使得它适用于各种类型的 Web 项目。Tomcat 是一个开源的 Java Servlet 容器，也可以作为 Web 服务器使用，一个 Tomcat 容器可以运行多个 Web 项目。由于 Tomcat 资源占用少、扩展性好和性能稳定等优势，备受广大开发者和软件公司的青睐。作为当前主流的 Web 应用服务器之一，Tomcat 在 Java Web 开发领域拥有广泛的应用^[44]。

Tomcat 的工作流程如下：

- ① Tomcat 容器启动时会读取 server.xml 中的配置信息，包括数据库连接池、Session 存活时间和端口号等，并初始化运行环境，初始化完成后进入监听状态。
- ② 当 Tomcat 接收到客户端的请求时，会根据请求的地址进行匹配，找到相应的 Servlet 对象，并将请求数据转发给该对象。
- ③ Servlet 对象接收请求数据后，首先创建 HttpServletRequest 和 HttpServletResponse 对象，然后根据请求调用 Get 或 Post 方法进行处理，处理完成后 Servlet 对象将处理结果交给 HttpServletResponse 对象。

本平台采用一台 Nginx 和多台 Tomcat 组成服务器系统，Nginx 作为反向代理和负载均衡器，映射域名请求并根据当前的服务器资源动态分配任务，Tomcat 作为应用服务器部署实际业务应用。

(6) Redis 和 MySQL

Redis 是一款开源的非关系型数据库，现已被广泛应用于缓存、消息队列、实时应用等领域^[46,47]，本课题使用 Redis 作为缓存数据库，增强平台处理并发请求的能力。使用 Redis 作为缓存具备以下优势：

- ① 优秀的 I/O 处理能力。Redis 将数据存储在内存中，具备低延迟的同时拥有强大的数据吞吐能力；
- ② 丰富的数据结构和功能。Redis 支持字符串、哈希表、列表、集合、有序

集合等多种数据结构，提供排序、交并集计算等丰富的操作功能，为开发人员实现业务逻辑提供了便利；

③ 多种持久化方式。持久化是指将内存中的内容写入到硬盘，以实现断电后的数据恢复，Redis 支持 RDB-AOF 混合持久化方式，既提供了安全冗余也保持了良好的性能。

MySQL 是一款流行的关系型数据库产品，隶属于 Oracle 公司旗下。虽然市场上存在多种关系型数据库系统，但 MySQL 凭借其出色的性能和易用性成为了其中的佼佼者，常见的数据库使用排名如图 4-3 所示。

Rank			DBMS	Database Model	Score		
Feb 2023	Jan 2023	Feb 2022			Feb 2023	Jan 2023	Feb 2022
1.	1.	1.	Oracle	Relational, Multi-model	1247.52	+2.35	-9.31
2.	2.	2.	MySQL	Relational, Multi-model	1195.45	-16.51	-19.23
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	929.09	+9.70	-19.96
4.	4.	4.	PostgreSQL	Relational, Multi-model	616.50	+1.65	+7.12
5.	5.	5.	MongoDB	Document, Multi-model	452.77	-2.42	-35.88
6.	6.	6.	Redis	Key-value, Multi-model	173.83	-3.72	-1.96
7.	7.	7.	IBM Db2	Relational, Multi-model	142.97	-0.60	-19.91
8.	8.	8.	Elasticsearch	Search engine, Multi-model	138.60	-2.56	-23.70
9.	10.	10.	SQLite	Relational	132.67	+1.17	+4.30
10.	9.	9.	Microsoft Access	Relational	131.03	-2.33	-0.23

图 4-3 数据库使用统计排名

从图中可知，MySQL 数据库是比较流行的开源关系型数据库系统，由于其资源占用少、支持多线程、功能完备，可以承担大型数据库系统任务，因此，本平台选用 MySQL 作为平台的关系型数据库系统。

本平台使用 Redis 作为 MySQL 的缓存池，平台通过计划任务定期将 MySQL 数据库中资源数据加载到 Redis 缓存，用户在请求资源时会优先检索 Redis 中的内容，若检索不成功再去查询 MySQL 数据库，以此增强平台的请求处理能力。

(7) Maven

Maven 是 Apache 软件基金会下的一个综合性项目管理工具，其广泛应用于项目的完整生命周期管理。Maven 功能强大，可对项目开发构建过程进行抽象和统一^[44]。Maven 覆盖了项目的清理、初始化、编译、测试、打包、集成测试、验证、部署各个生命周期，提高了构建过程的标准化程度。此外，Maven 还具有跨平台能力，可规范开发团队，降低开发成本。

Maven 仓库分为远程仓库和本地仓库，远程仓库一般由权威机构或组织进行维护，本地仓库则由开发者或软件公司自己管理^[44]。每个 Maven 项目都有一个 pom.xml (POM) 文件，文件内容定义了项目将要使用的各种依赖，Maven 根据

POM 文件中定义的依赖，在远程仓库中检索对应版本的 jar 包并下载到本地仓库，项目能够直接使用本地仓库中已下载的依赖包，从而实现 Maven 对项目依赖的管理。Maven 工作原理如图 4-4 所示。

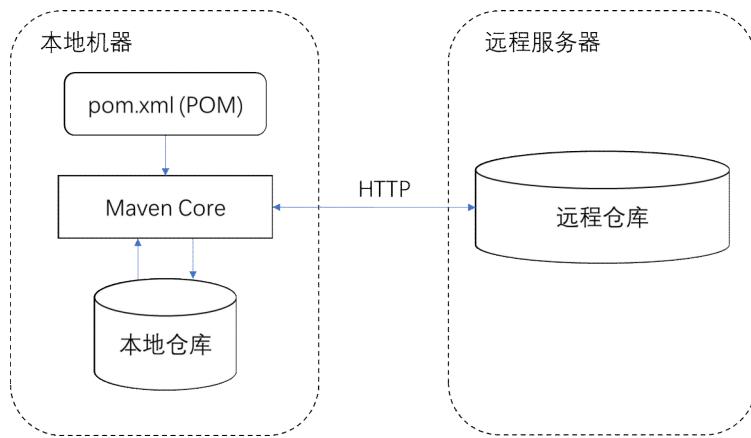


图 4-4 Maven 原理图

(8) TensorFlow

TensorFlow 是由谷歌公司开发的深度学习框架，其源头可以追溯至 2011 年谷歌 Brain 团队的内部项目 DistBelief，该项目旨在为深度神经网络构建机器学习工具。随着谷歌开发者的不断完善，TensorFlow 于 2015 年 11 月以 Apache License 2.0 的开源协议正式发布，并于 2017 年 2 月发布了第一个正式版本 1.0。在 2018 年 9 月发布的 1.2 版本中，TensorFlow 融合了高级 API Keras，这是 TensorFlow 发展历程中的重要里程碑。2019 年 9 月，TensorFlow 发布了 2.0 版本，该版本引入了新的库的同时保留了与旧库的兼容性，大幅增强了 TensorFlow 的功能^[48]。

相比于其他深度学习框架，TensorFlow 在 API 丰富程度、模型丰富程度、训练可视化、生产部署方面具备优势，K.D.^[49]将 TensorFlow 与 PyTorch 之间的差异进行总结，结合 StackOverflow、Github 等开发者社区的建议，了解到相比于 PyTorch，TensorFlow 具备入门资源丰富、可视化程度高等优势，因此，本平台的推荐系统选用 TensorFlow 作为开发框架。

4.2 开发环境搭建

本课程思政资源推荐管理平台的开发环境如表 4-1 所示。

第四章 平台实现

表 4-1 平台开发环境

类别	详情
CPU	AMD Ryzen 7 4800H 2.9GHz
GPU	NVIDIA GeForce RTX2060 6G
JDK	1.8
容器框架	Spring 5.2.5
MVC 框架	SpringMVC 5.2.5
前端组件库	Layui 2.6.8
前置服务器	Nginx 1.18.0
应用服务器	Tomcat 9.0.60
非关系型数据库	Redis 5.0.14.1
关系型数据库	MySQL 5.5.36
持久层框架	MyBatis 3.5.1
数据库连接池	Druid 1.1.12
安全框架	JWT 3.4.0
项目管理工具	Maven 3.3.9
版本控制工具	Git 2.35.1
深度学习框架	TensorFlow 1.15.0

平台开发环境的搭建步骤如下：

(1) 新建 Maven 项目

在 IDEA 中新建项目之后，使用 Maven 的 maven-archetype-webapp 骨架创建平台项目，在 Maven 仓库网站 mvnrepository.com 中搜索项目依赖的组件，将其对应的 maven 标签添加到 pom.xml 文件中，实现对平台开发所依赖的 jar 包进行统一的管理。

(2) 完善目录结构

当前创建的项目中目录结构不完整，需要手动进行补充，目录结构如图 4-5 所示。

在 src/main 目录下新建 java 文件夹并设置为 Source Root，根据 GroupId 配置 java 中的包目录，该目录将存放平台所有的 Java 代码，以实现业务逻辑；

在 src/main 目录中新建 resources 文件夹并设置为 Resources Root，该目录将存放所有的配置文件；

在 resources 目录中分别创建 applicationContext.xml、applicationRedis.xml、dispatcherServlet.xml、mybatis.xml、redis.properties 和 jdbc.properties 实现对 Spring、

SpringMVC、MyBatis、MySQL 和 Redis 的配置；

在 src 目录下新建 test/java 文件夹，并设置为 Test Source Root，该目录将存放所有的测试文件，通过 junit 对代码进行单元测试。

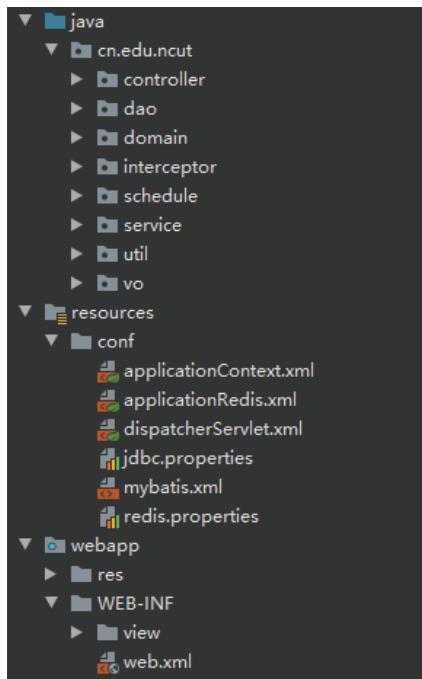


图 4-5 项目目录结构图

(3) 配置 Spring

在 applicationContext.xml 文件中对 Spring 容器进行配置，关键配置信息如下：使用 component-scan 标签，配置服务层业务对象 service 包所在的位置；使用 property-placeholder 标签，配置 jdbc.properties 文件位置；使用 bean 标签，分别设置数据库连接信息、MyBatis 配置文件和持久层数据访问对象 dao 包所在位置；使用 scheduled-tasks 标签配置平台的定时计划。

(4) 配置 SpringMVC

在 dispatcherServlet.xml 文件中对 SpringMVC 进行配置，关键配置信息如下：使用 component-scan 标签，配置调度层控制器对象 controller 包所在的位置；使用 bean 标签，为视图解析器配置解析目标；使用 interceptor 标签，为拦截器配置拦截目标和对应的要使用的拦截器。

(5) 配置 MyBatis

在 mybatis.xml 文件中对 MyBatis 进行配置，关键配置信息如下：使用 settings 标签，配置驼峰命名自动规则映射；使用 typeAliases 标签，配置实体类 domain

包所在位置；使用 `plugins` 标签，配置分页查询 `pagehelper` 插件；使用 `mappers` 标签，指定持久层数据访问对象对应的 `mapper` 文件位置。

(6) 配置 Redis

在 `applicationRedis.xml` 文件中配置 Redis 连接池，关键配置信息如下：使用 `property-placeholder` 标签，配置 `redis.properties` 文件位置；使用 `bean` 标签设置 Redis 连接池配置信息和 `jedis` 连接池。

(7) 配置 Git

Git 可实现对代码的版本控制和远程同步，在计算机中下载并安装 Git 客户端，为其配置连接 Github 项目仓库的密钥信息，之后在项目的根目录中建立本地仓库对代码文件、配置文件、资源文件等进行追踪，并通过 `.gitignore` 文件设置忽略追踪的文件路径。

4.3 平台架构实现

根据第四章的平台架构设计，按照技术实现分析，本平台可分为四层架构，分别为与用户交互的视图层，接收请求、预处理请求和返回处理结果的调度层，处理实际业务的服务层和提供数据库访问的持久层。

(1) 视图层

视图层为用户提供交互界面，采用 JSP 动态页面技术配合 Layui 组件库进行模块化编写，通过请求作用域将请求参数以 `get` 或 `post` 方式发送到调度层，使用 JSTL 将调度层发来的处理结果解析并显示到页面中。视图层所有的 JSP 文件均存储在 `webapp/WEB-INF/view` 目录下，并按照服务类别分类存放，`part` 目录存放公共组件页面。

(2) 调度层

用户的业务请求将会在调度层汇集，通过拦截器检验后，根据实现方式调用服务层的多个业务类对象进行处理，调度层由 SpringMVC 提供的中央调度器 `DispatcherServlet`、多个自定义拦截器和控制器对象组成。每个拦截器实现一种或多种预处理功能，本平台编写的拦截器有登录状态校验器 `JWTInterceptor` 和管理权限校验器 `AdminInterceptor`，可分别实现确认用户登录状态和确认用户是否具备管理权限，实现了对用户请求的预处理。中央调度器 `DispatcherServlet` 是 SpringMVC 的核心 `Servlet`，视图层的业务请求会首先到达中央调度器，并由中央调度器协调业务处理，其配置信息如图 4-6 所示。

```
<context:component-scan base-package="cn.edu.ncut.controller"/>

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/view/"/>
    <property name="suffix" value=".jsp"/>
</bean>

<mvc:annotation-driven/>

<mvc:interceptors>
    <mvc:interceptor>
        <mvc:mapping path="/**/home.do"/>
        <bean class="cn.edu.ncut.interceptor.JWTInterceptor"/>
    </mvc:interceptor>
    <mvc:interceptor ...>
    <mvc:interceptor ...>
    <mvc:interceptor ...>
    <mvc:interceptor>
        <mvc:mapping path="/admin/**"/>
        <bean class="cn.edu.ncut.interceptor.AdminInterceptor"/>
    </mvc:interceptor>
</mvc:interceptors>

<bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <property name="maxUploadSize" value="104857600" />
    <property name="defaultEncoding" value="UTF-8"/>
    <property name="resolveLazily" value="true"/>
</bean>
```

图 4-6 SpringMVC 部分配置信息

在中央调度器接收请求后，请求首先被传递给处理器映射器(HandleMapping)。处理器映射器负责在应用程序上下文中查找控制器对象，并将控制器对象和所有相关的拦截器组合成一个处理器执行链(handler chain)。处理器执行链被返回给中央调度器。随后，中央调度器将控制器对象传递给处理器适配器(Handler Adapter)。处理器适配器负责执行控制器对象的方法，并返回一个包含视图和模型数据的 ModelAndView 对象。中央调度器将这个 ModelAndView 对象传递给视图解析器(ViewResolver)，视图解析器将返回视图对象的完整路径。最后，中央调度器调用视图类(View)的方法，将模型数据放入请求(request)作用域中，并将请求转发到对应的视图层。视图层完成响应并返回结果。该模块化的实现有助于提高系统的可维护性和扩展性。

(3) 服务层

服务层由 Spring 容器中的多个业务类对象组成，业务类对象为调度层提供了关于实体的模块化功能，由调度层进行组合调用，从而完成业务处理。要实现 Spring 容器管理业务类对象，首先需要在 Spring 配置文件 applicationContext.xml 中声明业务类对象的包位置，本项目中使用的业务类均放在 java/cn.edu.ncut/service 目录下，每个业务类对象都对应着接口文件和实现文件，开发时先在接口文件中声明方法名、参数类型和返回类型，然后在实现文件中根据具体情况对接口的方法进行实现，并在方法名的上方添加 Service 注解，以此实现良好的维护性和拓展性。

(4) 持久层

持久层由 MyBatis 数据访问对象、Redis 和 MySQL 数据库组成。数据访问对象通过 MyBatis 提供的动态代理创建，并由 Spring 容器进行管理，开发者通过数据访问对象执行 SQL 语句。每个数据访问对象都有一个对应的同名 Mapper 文件（也叫 SQL 映射文件）。使用时需要首先在 DAO 接口类中定义好方法名、参数类型和返回值类型，然后在 Mapper 文件中为接口定义的方法设计 SQL 语句，参数类型通过 parameterType 使用 Java 类型的全限定名称指定，对于有返回值的方法则需要通过 resultType 指定返回值类型。通过数据访问对象对数据库进行操作，可以实现逻辑业务代码与 SQL 语句的分离，增强了代码的可维护性。

4.4 基本功能实现

平台首页如图 4-7 所示，本节将按照前文对平台功能模块的梳理与设计，针对其中的基本功能的实现进行详细阐述。



图 4-7 平台首页

(1) 注册与登录

游客可通过点击平台首页顶部导航栏中的“我的”跳转到注册与登录页面，在该页面可通过点击“注册”或“登录”选项卡切换注册与登录功能，默认情况下页面选中“登录”选项卡。视图层将用户的注册或登录请求以表单的形式封装发送到调度层的中央调度器，再由中央调度器将请求发送至控制器对象；控制器对象收到业务请求后，首先从请求作用域中取出表单信息，即用户的注册、登录

信息，随后调用服务层的用户相关业务对象并传递参数，执行注册或登录业务；业务对象根据传递的参数信息调用持久层数据访问对象，对注册信息检查是否重复，对登录信息检查账号密码是否有效；数据访问对象会将查询参数填充到 SQL 语句中执行，并返回语句的执行结果；之后业务处理结果将依次返回至上一级，并进行相关的处理，调度层控制器对象会根据最终业务的处理结果，返回给予用户不同的反馈，登录业务时序图如图 4-8 所示。

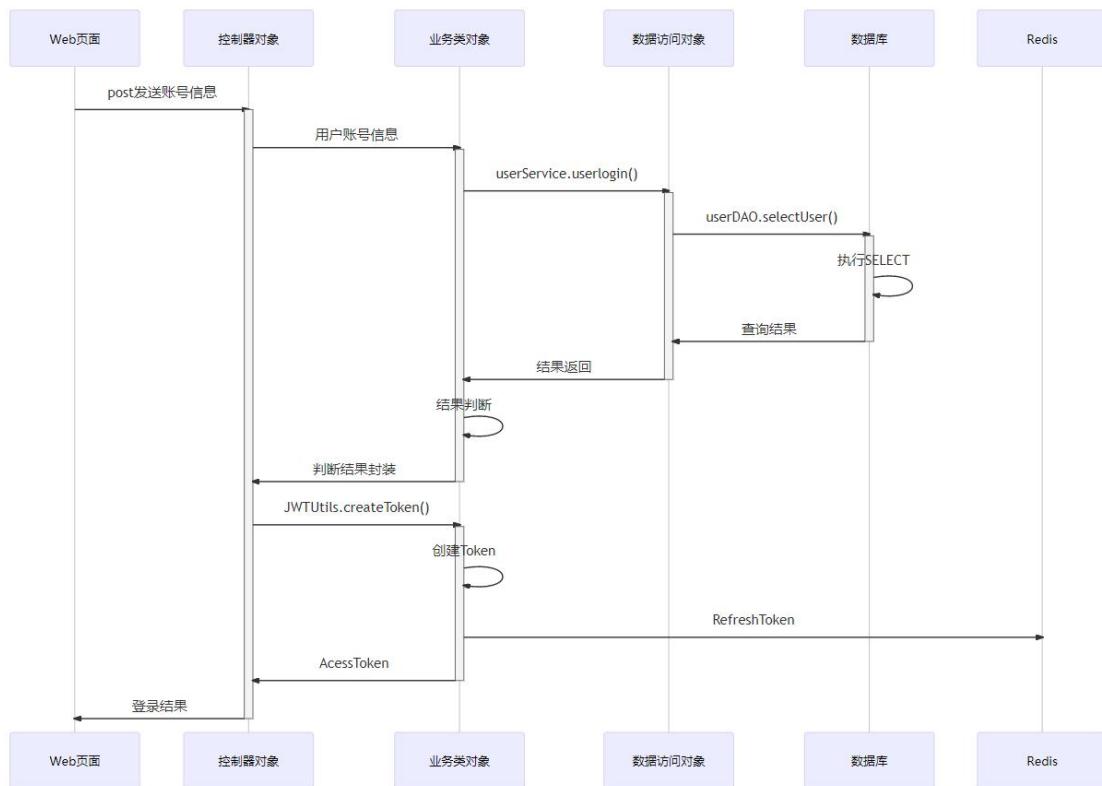


图 4-8 登录业务时序图

(2) 登录保持

平台采用长短令牌（Token）机制实现活跃用户的登录状态保持功能，用户在登录时可以选中“记住登录状态”复选框来启用此功能。当用户名和密码验证通过后，服务器将根据登录信息使用 JWT 生成两个有效期不等长的令牌，分别是短令牌 accessToken 和长令牌 refreshToken，短令牌会封装成 Cookie 返回到用户的浏览器中，而长令牌则存储在 Redis 数据库中。当用户访问平台时有以下三种情况：1) 短令牌和长令牌同时有效，仅进行令牌校验，用户的登录状态为已登录；2) 短令牌失效，长令牌有效，销毁 Redis 中原有的长令牌，为当前用户生成新的短令牌和长令牌，用户登录状态为已登录；3) 短令牌和长令牌均失效，

此时的用户为非活跃用户，登录状态为未登录。

服务端的一系列操作对于用户而言是无感知的，只要用户访问平台的间隔时间不超过长令牌的有效期，就可以保留登录状态继续使用平台。相比于传统的单令牌实现方案，长短令牌机制既实现了良好的用户体验，也节约了服务器资源。

(3) 退出登录

用户在个人中心页面左侧可以看到“退出登录”按钮，用户点击后将会退出当前的登录状态，并返回平台首页。该功能由控制器对象的 userLogout 方法实现，在收到用户的退出登录请求后，该控制器对象会将 Redis 中存储的长令牌销毁，并返回给视图层一个空的短令牌，设置其存活时间为 0，由于本平台依靠存储在 Redis 和 Cookie 中的 Token 来辨别用户的登录状态，用空白替代 Token 可以实现清除用户登录状态的效果，该功能时序图如图 4-9 所示。

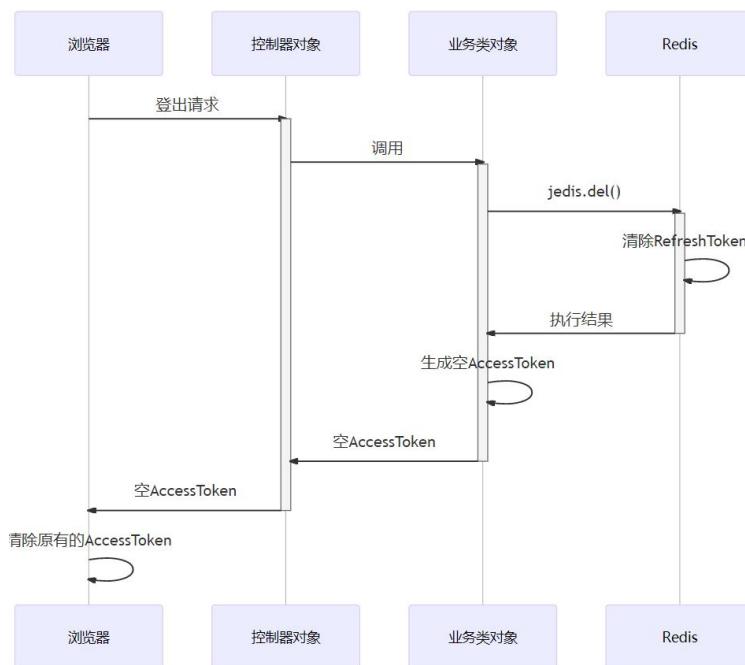


图 4-9 登出业务时序图

(4) 点赞评论

用户可以对平台中的资源进行点赞与评论操作，以课程思政案例资源为例，点赞与评论功能界面如图 4-10 所示。

第四章 平台实现



图 4-10 点赞与评论页面

普通用户可以点击资源详情页面下方的点赞按钮为该资源增加一个赞，也可以在文本框中输入评论内容后点击评论按钮为该资源增加一条评论，每个资源获得的点赞数量和评论内容都会在资源详情页显示出来，以便用户参考。

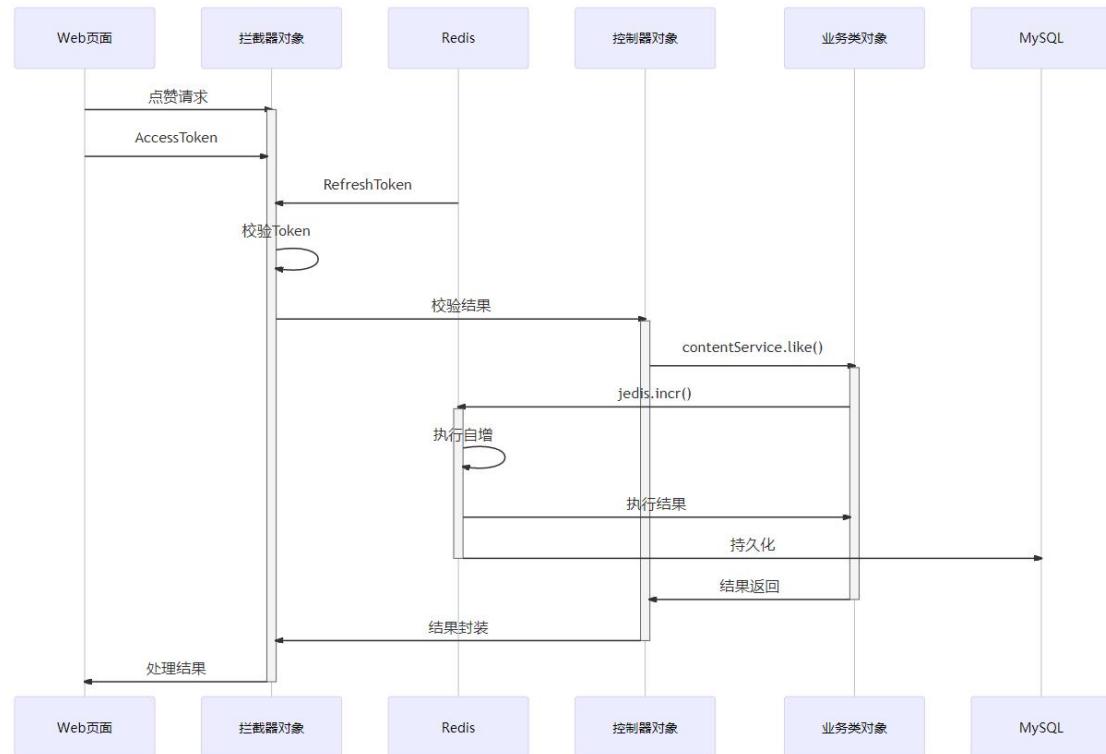


图 4-11 点赞业务时序图

点赞评论功能根据资源类型的不同，其关联的控制器对象也不相同，点赞产生的记录将由 Redis 进行存储和计数，而评论数据则直接写入 MySQL 中，当用户对案例资源发起点赞评论请求时，调度层的拦截器 JWTInterceptor 会首先检验用户的登录状态，随后调度层对应的控制器对象从 Token 中解析出当前的用户名，由服务层根据点赞或评论请求调用 Redis 连接池或持久层的 MySQL 数据访问对象，在数据库中增加一条记录，随后业务的处理结果逐级返回，由调度层将处理结果代码和信息返回到视图层，最终视图层根据返回的结果为用户提供反馈，评论功能的时序图如图 4-11 所示。

平台为教师用户提供课程思政案例、素材资源的上传功能，以及讨论区的话题发布功能，由于功能的实现原理基本相同，这里以课程思政案例上传为例，阐述功能实现方式，资源上传页面表单布局，包含了分类选择框、标题输入框和富文本编辑器，页面布局如图 4-12 所示。

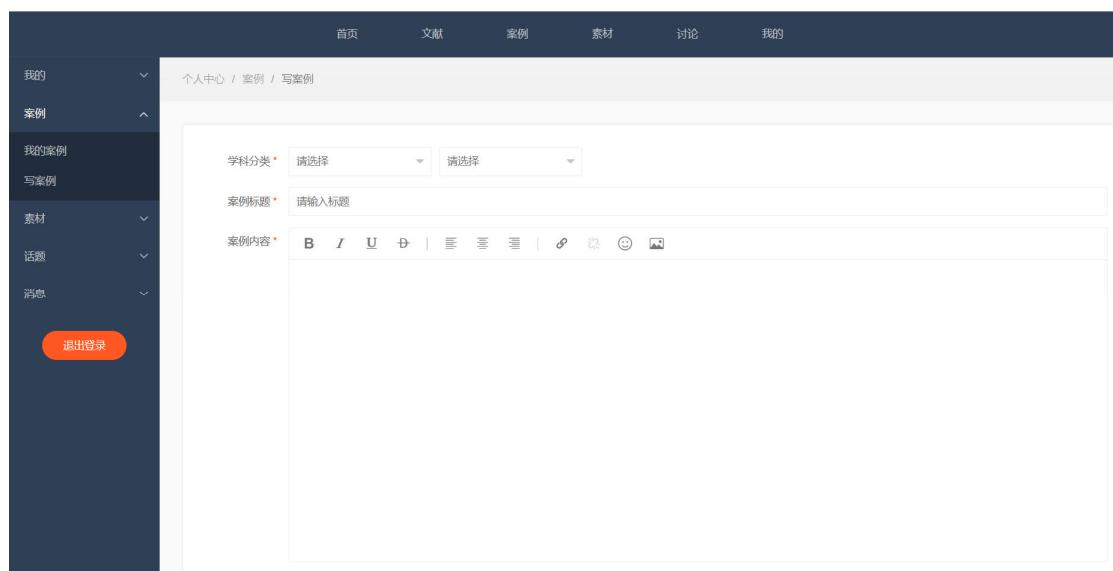


图 4-12 资源上传页面

(5) 分类选择

分类选择框中的分类内容存储在数据库中，用户首次打开页面时，调度层控制器会将一级分类的内容随着页面信息一起发送给视图层，当用户点击一级分类后，视图层将携带一级分类信息向调度层发起二级分类查询请求，持久层的数据访问对象查询完成后，查询结果由服务层封装，再由调度层返回给视图层，最终填充到二级分类选择框中。

(6) 图片插入

富文本编辑器提供了插入图片的功能，用户可以点击图片按钮，选择计算机

第四章 平台实现

中的图片插入到案例内容中，当用户选择图片并点击确定后，图片会被切分成若干数据包发送到调度层，调度层对应的控制器对象会根据用户名和系统时间为图片创建路径信息，将图片保存在服务器的硬盘中，并将图片在服务器中的地址返回给视图层，图片最终会以路径信息插入到案例中，视图层会根据图片地址从服务器中获取图片并显示。

(7) 资源上传

用户在编辑完成后，可以点击表单下方按钮将课程思政案例上传到平台中，用户点击上传按钮后，视图层会以表单的形式对当前页面内容（所属分类、标题、正文和图片路径）进行封装后到达调度层，调度层对封装的内容进行拆包，之后由服务层的业务对象调用持久层数据访问对象将用户上传的案例写入到数据库中，写入完成后调度层将向视图层发送页面跳转指令，最终由视图层为用户跳转到个人中心首页。

(8) 个人资料

用户可以在个人中心的个人资料页面中修改或完善个人信息，个人资料页面如图 4-13 所示。用户在请求个人资料页面时，调度层会调用服务层的用户业务类对象，通过持久层检索当前用户的个人资料，并将查询到的信息发送给视图层，以实现表单中自动填充已经存在的信息；用户点击“更新”按钮后，页面以表单的形式将用户填写的个人资料封装并发送到调度层，经服务层业务类对象，由持久层数据访问对象执行数据更新操作。

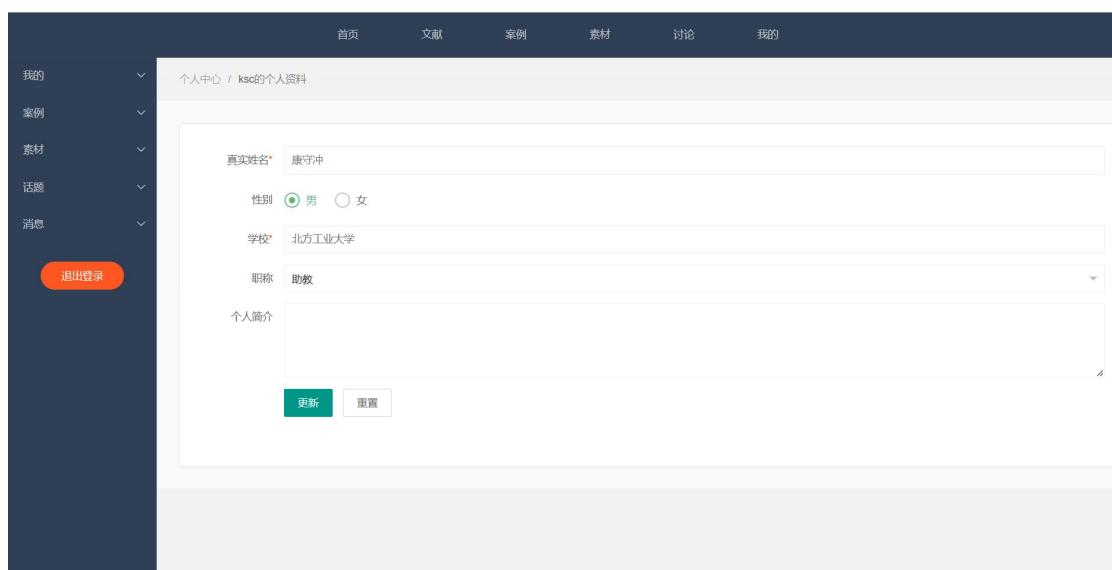


图 4-13 个人资料页面

(9) 记录管理

记录管理包含了用户在平台中的点赞、评论记录，用户可以在个人中心页面找到相应入口，记录管理页面以列表的形式展现用户的点赞、评论记录，其页面如所示。记录页面根据资源分区划分成了多个标签页，用户点击对应的标签页即可发起查询记录请求，请求经调度层、服务层到达持久层，由数据访问对象根据用户名在多个表中执行查询操作，查询结果以 List 列表的形式经 ModelAndView 封装后返回到视图层，视图层使用 JSTL 对列表进行遍历显示。

(10) 公告管理

管理员可通过管理页面的公告管理入口，对平台中发布的公告信息做出调整，公告管理界面如图 4-14 所示。管理员首次打开公告管理页面时，页面请求先由调度层的 AdminInterceptor 拦截器对管理员身份进行核验，之后调度层将通过服务层业务对象和持久层数据访问对象查询收集公告数据，并以 List 的形式随页面信息发送到视图层；视图层以列表的形式对公告信息进行显示，列表上端是发布公告的入口，列表每行包含了当前公告标题、访问数量、发布时间和操作按钮，管理员可以点击公告标题来查看公告详情，可以点击垃圾篓按钮删除过时或错误的公告，也可以点击上方的发布公告按钮打开公告发布页面，编辑并发布新的公告。

标题	访问数	发布时间	操作
上传资源步骤说明	6	2022-04-23 18:18	
用户须知	2	2022-04-23 18:17	

图 4-14 公告管理页面

(11) 通知管理

通知一般为平台或管理员发送给用户的通知，所有通知均存在于数据库中，并设有已读标志位，管理员可以通过平台管理页面向用户发送通知，用户可以通过个人中心页面查看收到的通知，管理员端的通知管理界面如图 4-15 所示。

第四章 平台实现

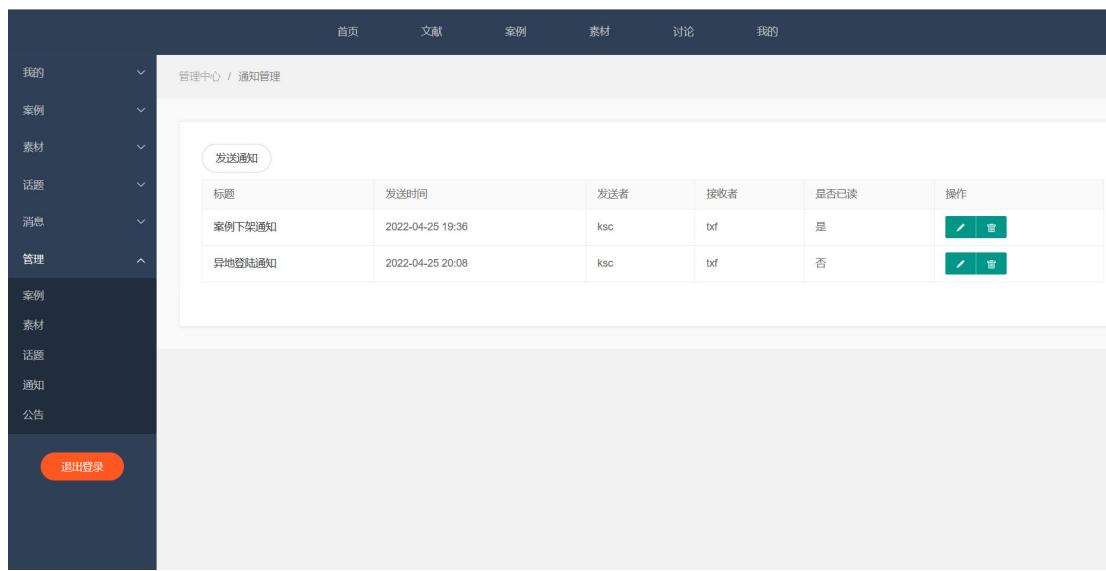


图 4-15 通知管理页面

与公告管理类似，管理员在打开通知管理页面时，服务层的 AdminInterceptor 拦截器会首先检查管理员的身份信息；请求到达调度层后，由调度层调度服务层的业务对象，再由持久层的数据访问对象执行数据查询，最后调度层控制器将对通知数据进行封装，随页面信息一起发送给视图层；视图层经过解析后，将通知数据以列表的形式展示给用户，列表的每行都包括了通知标题、发送时间、发送方、接收方、已读标志和操作按钮，发送通知的入口位于列表上方，管理员可以点击通知的标题查看内容、点击撤回按钮将通知撤回，也可以通过发送通知的入口打开通知编辑页面，给用户发送新通知。

4.5 基于优化算法的课程思政案例推荐

本小节将介绍基于本文提出的深度强化学习优化算法开发的课程思政资源推荐系统，将分别从数据集与数据清洗、推荐系统集成和课程思政案例推荐实现三个方面来叙述。

4.5.1 数据集与数据清洗

平台用于推荐的数据集为课程思政资源数据集，其数据结构如 3.4 小节中的表 3-6 所示，由于本文提出的推荐算法并不涉及对物品内容的特征工程，因此在保证模型对于实验数据集和平台数据集的输入输出格式一致的情况下，可以实现推荐算法从实验数据集到平台数据集的移植部署。

为了展示本文提出的推荐算法在平台中的推荐效果，这里使用了实验室服务器中两万余条课程思政文献信息数据集，在将数据集载入平台之前对数据集进行了如下的数据清洗操作。

(1) 缺失值处理：检测和识别数据集中存在缺失值的样本，对于缺失关键信息的样本进行删除，对于缺失非关键信息的样本进行插值填充。

(2) 异常值处理：检测和处理数据集中的异常值，对于关键信息异常的样本进行删除，对于非关键信息异常的样本进行数据替换。

(3) 重复值处理：检测和删除数据集中的重复样本，确保数据的唯一性和一致性。

(4) 数据类型转换：检测和识别数据集中每个样本的数据类型，对于不正确或不一致的数据类型进行转换。

数据集原有数据样本 26618 个，经过数据清洗处理后，剩余有效数据样本 21672 个。

4.5.2 推荐系统集成

Java 技术栈是 Web 项目开发中最成熟的技术，使用 Java 技术栈开发的平台系统具备安全性强、框架稳定、扩展性强以及支持高并发等优势；而 TensorFlow 是模型训练领域最主流的框架，具备灵活的模型构建能力、强大的计算能力以及拥有丰富的文档和教程资源。为了使平台在实现良好的推荐效果的同时兼顾高可用性，本文的平台系统部分基于 Java 技术栈进行开发实现，而推荐系统部分则采用 Python 语言基于 TensorFlow 框架开发，并对本文提出的针对样本效率优化的推荐算法进行部署实现，为用户提供交互式推荐功能。

本文开发的平台系统与推荐系统使用 Socket 编程通过 TCP 协议实现数据交换，推荐系统既可以与平台系统部署在同一台服务器中，也可以部署到局域网中其他具备 CUDA 计算能力的服务器上，这样既保留了两种编程语言各自的生态优势，也使得平台的部署更加灵活，提高了平台整体的容错率。

4.5.3 课程思政案例推荐实现

在实际应用中，平台通过服务层调用推荐系统，实现对课程思政资源的推荐。推荐系统跟随平台系统启动，使用平台中课程思政资源数据集进行预训练，训练后的模型参数以.h5 文件的形式保存在服务器本地。推荐系统在将深度强化学习推荐模型加载到内存后转入监听状态，等待平台的推荐请求，当接收到推荐请求时，模型通过.h5 文件加载决策模型参数，进而执行推荐动作。推荐功能入口在

第四章 平台实现

平台的资源专区右侧，用户在已登录状态下可点击右侧入口使用平台的推荐服务，下面将详细解释针对课程思政资源的推荐功能实现方式。推荐功能时序图如图 4-16 所示。

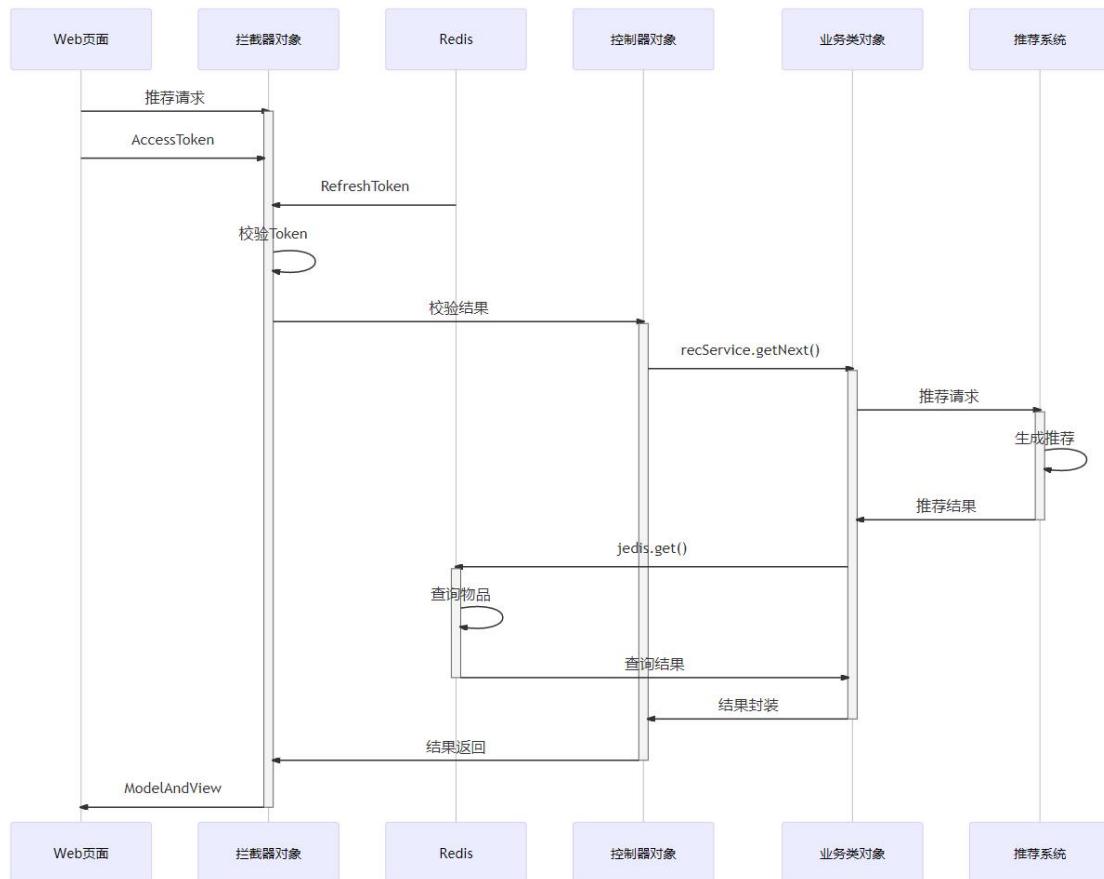


图 4-16 资源推荐时序图

- (1) 当用户发起推荐请求后，视图层首先将用户信息、当前浏览的资源信息和用户评价封装发送给调度层；
- (2) 调度层收到推荐请求后，从请求作用域中获取到用户信息、资源信息和用户评分，并将这些信息作为参数提交到服务层的推荐业务代理对象；
- (3) 推荐业务对象并不负责实际的推荐任务，而是与服务器中的推荐系统建立 TCP 链接，将推荐请求和参数发送至推荐系统并等待推荐结果；
- (4) 推荐系统在接收到推荐请求后，将调用深度强化学习推荐模型，基于当前的推荐策略为用户生成单个物品推荐，推荐结果以物品编码的形式返回，在单次推荐结束后模型会根据交互记录对推荐策略进行小幅更新，以此来适应用户变化的兴趣偏好；
- (5) 服务层业务对象根据返回的物品编码调用持久层数据访问对象，在数

据库中查询该资源并生成推荐结果，将该结果返回到调度层，随后由调度层将推荐结果发送至视图层呈现给用户。

4.6 本章小结

本章对平台的功能实现进行了阐述，首先介绍了平台的开发技术、开发环境以及相关的配置，随后介绍了平台架构和基本功能的实现方法，最后重点介绍了基于本文算法开发的深度强化学习课程思政案例推荐系统。至此，平台和推荐系统的开发工作基本完成，接下来将对平台整体进行功能和性能的测试验证。

第五章 平台测试

本章将对课程思政资源推荐管理平台进行功能和性能测试，功能测试可以确保项目按照预期功能工作，满足用户需求和业务规范，而性能测试则评估项目在负载下的性能表现，可以帮助优化项目的性能和响应能力。功能测试和性能测试在 Web 项目开发过程中都是非常重要的，通过功能测试和性能测试以验证平台的可行性，对发现的问题进行修复和完善，提高平台的软件质量，增强安全性和稳定性^[50]，以提供优秀的用户体验和可用性。

5.1 测试环境

测试环境如表 5-1 所示。

表 5-1 测试环境

类别	详情
CPU	AMD Ryzen 7 4800H 2.9GHz
GPU	NVIDIA GeForce RTX2060 6G
内存	DDR4 16G
硬盘	SSD 1T
系统版本	Windows 10
浏览器	Chrome, Edge
测试工具	IntelliJ IDEA, PyCharm, Jmeter

5.2 功能测试

功能测试是对本平台基本功能和组合功能的测试，以测试用例为单位，对类型相同或返回结果类似的功能采用同种测试用例^[44]。测试用例是对一个具体业务的场景分解，包含了测试名称、描述信息、测试人员、前置条件、测试步骤和测试结果等。本小节将对课程思政资源推荐管理平台的注册登录、资源浏览、个人资料、点赞评论、资源上传、公告管理和资源推荐进行测试

(1) 注册登录

第五章 平台测试

注册登录功能是实现平台中所有其他功能的基础，包括注册、登录和保持，能够为用户提供良好的使用体验。

表 5-2 注册登录测试用例

名称	注册登录
用例描述	实现用户的注册登录相关功能
前置条件	无
测试步骤	<ol style="list-style-type: none">1) 在地址栏中输入平台的 URL;2) 点击顶部导航栏“我的”选项;3) 点击注册选项卡;4) 输入注册信息，点击注册按钮;5) 输入登录信息，勾选“记住我”复选框，点击登录按钮;6) 关闭浏览器，重新访问平台;7) 点击退出登录按钮。
测试要点	<ol style="list-style-type: none">1) 未登录状态下，点击导航栏“我的”是否可以成功跳转到登录页面;2) 点击注册后，数据库表是否有新增记录;3) 勾选“记住我”复选框并登录后，浏览器是否收到 Token;4) 关闭浏览器重新访问平台，登录状态是否得以保持;5) 点击退出登录后，浏览器是否清除了 Token。
测试人员	康守冲
测试结果	测试通过

(2) 资源浏览

用户在登录后可以在分区中浏览平台中的资源，由于实现原理和方式类似，这里仅对课程思政案例资源进行测试。

表 5-3 案例浏览测试用例

名称	案例浏览
用例描述	实现用户对案例资源的访问功能
前置条件	无

第五章 平台测试

续表 5-3 案例浏览测试用例

测试步骤	1) 在地址栏中输入平台的 URL; 2) 点击顶部导航栏“案例”选项; 3) 点击页面显示的第一条结果; 4) 返回案例分区首页并刷新页面。
测试要点	1) 点击导航栏“案例”是否可以成功跳转到课程思政案例分区; 2) 点击案例条目后，是否可以成功打开案例详情页; 3) 返回分区首页并后，案例条目的访问数量是否有增加。
测试人员	康守冲
测试结果	测试通过

(3) 个人资料

用户登录后，可在个人中心中完善個人資料，以使用平台更多的功能。

表 5-4 修改個人資料测试用例

名称	個人資料修改
用例描述	用户对個人資料的编辑与提交
前置条件	用户登录
测试步骤	1) 点击顶部导航栏“我的”选项; 2) 点击左侧我的-個人資料; 3) 编辑個人資料; 4) 点击提交。
测试要点	1) 点击导航栏“我的”是否可以成功进入个人中心; 2) 個人資料页面是否可以显示用户信息; 3) 点击提交后，数据库個人資料表中数据是否更新。
测试人员	康守冲
测试结果	测试通过

第五章 平台测试

(4) 点赞评论

已完善个人资料的用户可以对平台中的资源进行点赞和评论操作，这里以课程思政案例资源为例，进行点赞评论功能的测试。

表 5-5 点赞评论测试用例

名称	案例点赞和评论
用例描述	实现用户对案例资源的点赞评论功能
前置条件	用户完善个人资料
测试步骤	<ol style="list-style-type: none">1) 在地址栏中输入平台的 URL;2) 点击顶部导航栏“案例”选项;3) 点击页面显示的第二条结果;4) 点击点赞按钮;5) 编辑评论内容，点击评论按钮;6) 刷新页面。
测试要点	<ol style="list-style-type: none">1) 案例详情页是否可以显示点赞和评论数量;2) 点赞成功是否有提示;3) 评论成功是否有提示;4) 刷新页面后，该案例条目的点赞和评论数量是否有增加。
测试人员	康守冲
测试结果	测试通过

(5) 资源上传

用户在完善个人资料后，可以通过个人中心发布新的资源与其他教师共享，这里以课程思政案例资源为例，测试上传功能。

表 5-6 案例上传测试用例

名称	案例上传
用例描述	实现用户发布新的案例
前置条件	用户完善个人资料
测试步骤	<ol style="list-style-type: none">1) 点击顶部导航栏“我的”选项;2) 点击左侧案例-新案例;3) 依次点击一级、二级分类选择框;

第五章 平台测试

续表 5-6 案例上传测试用例

	4) 编辑案例内容; 5) 点击插入图片按钮, 选择测试图片; 6) 点击发布案例按钮; 7) 点击我的-我的案例。
测试要点	1) 一级分类选择框是否有数据; 2) 二级分类选择框数据是否与一级联动; 3) 图片是否上传正常; 4) 图片是否插入正常; 5) 案例是否上传正常; 6) 我的案例页面是否能看到新发布的案例。
测试人员	康守冲
测试结果	测试通过

(6) 公告管理

管理员可使用管理功能对平台的公告、通知、资源等进行管理, 这里以公告管理为例, 测试平台的管理功能。

表 5-7 公告管理测试用例

名称	案例浏览
用例描述	实现用户对案例资源的访问功能
前置条件	管理员账户登录
测试步骤	1) 点击顶部导航栏“我的”选项; 2) 点击左侧管理-公告管理; 3) 点击发布公告; 4) 编辑公告内容, 点击发布; 5) 返回公告管理页面;
测试要点	1) 管理员点击“我的”是否正确跳转到管理中心页面; 2) 公告管理页面的公告数据是否加载正常; 3) 新发布的公告是否在数据库中新增了记录; 4) 公告发布后, 公告管理页面是否显示正常。

第五章 平台测试

续表 5-7 公告管理测试用例

测试人员	康守冲
测试结果	测试通过

(7) 资源推荐

平台为已完善资料的用户提供资源推荐服务，以课程思政案例资源为例，测试案例资源推荐功能。

表 5-8 案例推荐测试用例

名称	案例推荐
用例描述	实现平台对用户的案例推荐功能
前置条件	用户完善个人资料
测试步骤	<ol style="list-style-type: none">1) 点击顶部导航栏“案例”选项；2) 点击页面右侧的“开始探索”按钮；3) 点击下方的评分按钮。
测试要点	<ol style="list-style-type: none">1) 点击开始探索按钮后，页面是否加载正常；2) 点击评分按钮后，评分和案例数据能否到达推荐系统；3) 推荐系统能否给出推荐结果；4) 推荐结果能否正确展示给用户。
测试人员	康守冲
测试结果	测试通过

5.3 性能测试

性能测试通过模拟请求考察平台整体的负载能力，属于非功能性测试，平台的负载能力直接关系到用户体验。本文使用 Jmeter 测试工具对平台进行性能测试，通过设计测试用例，模拟批量用户同时发起请求的场景，并记录响应时间和吞吐量，从而测试平台的并发性能和负载性能，有利于开发者摸清平台的性能瓶颈，避免因高并发或高负载导致服务器崩溃。

性能测试既要求服务器具备良好的硬件规格，也对平台架构设计提出了较高要求，根据功能原理，本节针对首页访问、资源访问、用户注册和资源推荐功能

进行并发压力测试，平均响应延迟以 5s 为目标，若单次实验发送的所有请求均成功处理并响应则视为测试成功。

(1) 首页访问

首页访问是本平台最频繁、最基本的业务请求，用户访问首页产生的数据流量较大，是一种典型的并发压力场景。对于首页访问设置请求总数为 5000，单次并发量为 1000，进行 5 轮并发测试，首页访问压力测试结果如表 5-9 所示。

表 5-9 首页访问场景测试用例

目标接口	http://127.0.0.1:8889/common/index.do
请求方法	GET
请求总数	5000
单次并发线程数	1000
成功处理并响应数	5000
最小响应时间 (ms)	143
最大响应时间 (ms)	2694
平均响应时间 (ms)	1496.14
每秒事务数	368.13

(2) 资源访问

资源访问即用户进入分类专区后，点击条目获取资源详情，是平台的主要功能之一，这里以对文献资源的访问为例进行测试。压力测试请求总数设置为 5000，单次并发量为 1000，进行 5 轮并发测试，资源访问压力测试结果如表 5-10 所示。

表 5-10 文献访问场景测试用例

目标接口	http://127.0.0.1:8889/common/paper/5000/view.do
请求方法	GET
请求总数	5000
单次并发线程数	1000
成功处理并响应数	5000
最小响应时间 (ms)	5
最大响应时间 (ms)	1625
平均响应时间 (ms)	758.45
每秒事务数	396.57

(3) 用户注册

由于用户注册功能关系到数据库的写入操作，并且其请求频率较低，因此对于用户注册功能单次模拟执行 100 个用户的并发注册操作，进行 5 轮共计 500 次注册操作，用户的注册信息通过.csv 文件加载，其压力测试结果如表 5-11 所示。

表 5-11 用户注册场景测试用例

目标接口	http://127.0.0.1:8889/common/user/register.do
请求方法	POST
请求参数	username, email, password
请求总数	500
单次并发线程数	100
成功处理并响应数	500
最小响应时间 (ms)	1218
最大响应时间 (ms)	7967
平均响应时间 (ms)	5226.31
每秒事务数	62.72

(4) 资源推荐

平台结合推荐系统，为用户提供资源推荐服务，由于推荐系统需要消耗 CUDA 计算资源，受限于当前测试机的 GPU 计算能力，资源推荐压力测试设置并发量 100，进行 5 轮并发，共计 500 次访问。这里以文献资源的推荐为例进行压力测试，测试结果如表 5-12 所示。

表 5-12 资源推荐场景测试用例

目标接口	http://127.0.0.1:8889/common/paper/5000/0/next.do
请求方法	GET
请求总数	500
单次并发线程数	100
成功处理并响应数	500
最小响应时间 (ms)	450
最大响应时间 (ms)	5548
平均响应时间 (ms)	2182.7
每秒事务数	72.06

5.4 测试结果分析

通过功能测试和性能测试，对平台的实现效果和响应速度进行了测试，测试结果分析如下：

(1) 在功能实现方面，平台的基本功能、主要功能和关键功能均通过了测试用例，业务逻辑正常，符合开发预期；

(2) 在性能表现方面，平台可处理千人级别的并发请求并且表现稳定，具备高可用性，符合当前环境下的使用需求。

综上所述，本文所开发的平台整体表现良好，后续随着用户量的增加可考虑通过分布式等技术对平台的架构进行优化。

5.5 本章小结

本章首先介绍了测试的软硬件环境，随后测试了平台的功能实现效果，对平台的注册登录、资源浏览、个人资料、点赞评论、资源上传、公告管理和资源推荐功能进行了测试，然后测试了平台在高并发负载下的性能表现，针对首页访问、资源访问、用户注册和资源推荐功能进行并发压力测试，最后对测试结果进行了分析与总结，当前开发的平台整体具备良好的使用体验和高可用性，符合开发预期。

第六章 总结与展望

6.1 总结

课程思政教学是近年来的一个研究热点，当前课程思政的发展主要存在缺少课程思政案例参考的问题。有许多学者系统地设计并开发了课程思政资源平台，但存在平台健壮性不足和推荐系统理论支撑较弱的问题，对此，本文在以下几个方面进行了探讨：

(1) 本文以经典强化学习算法为基础，将用户与推荐系统的交互建模为马尔可夫决策过程，结合推荐任务场景，通过模型熵的方法优化算法的损失函数，提出了一种针对推荐多样性优化的深度强化学习推荐算法，提高了推荐结果的多样性。

(2) 在上述算法的基础上，为了实现更好的用户体验，针对智能体的策略更新流程，设计奖励预测模型优化智能体的交互更新过程，提出了一种针对样本效率优化的深度强化学习推荐算法，提高了强化学习的样本效率，加快了模型的训练速度。

(3) 对于教师用户的课程思政资源检索需求，本文从架构和功能层面系统地设计了一套课程思政资源平台，并借助容器、缓存、反向代理、负载均衡等技术开发实现了一个可用性高，健壮性强的课程思政资源平台。

(4) 对于平台的课程思政推荐功能实现，本文针对平台的课程思政数据集，基于本文提出的算法设计并开发了课程思政案例推荐系统，实现了为教师用户提供个性化的课程思政资源推荐，该推荐系统独立于平台之外，与平台的服务层通过 Socket 相关联，在实际应用中可将推荐系统部署到单独的具备 CUDA 计算能力的服务器中，降低了平台的运维成本，增强了平台的稳定性。

6.2 展望

本文针对深度强化学习推荐算法优化和课程思政资源平台开发都做了深入的尝试，并最终将二者相结合，在课程思政资源平台中基于提出的优化算法实现了对课程思政案例的个性化推荐。

但是，由于本人技术能力限制和其他客观因素影响，未来仍然可以从以下两

方面进行更深一步的探索：

(1) 算法性能优化。当前平台使用的推荐算法在实验评估中的性能表现与推荐领域的顶尖推荐算法仍有差距；经过实验数据分析，发现在模型训练的数据预处理阶段输入了大量的零值作为反馈值，导致智能体获得的奖励值整体偏低，影响了模型训练效果，后续可针对数据的稀疏性进行优化。

(2) 平台架构优化。平台当前的软硬件环境能够承载千人级别的并发压力场景，可以满足校园局域网内教师用户的使用，但是性能表现仍存在优化空间，后续可使用分布式等技术对平台架构进行改进，提高平台处理并发请求的能力。

参考文献

- [1] 习近平：把思想政治工作贯穿教育教学全过程 - 中华人民共和国教育部政府门户网站[EB/OL]. http://www.moe.gov.cn/jyb_xwfb/s6052/moe_838/201612/t20161208_291306.html.
- [2] 中共教育部党组关于印发《高校思想政治工作质量提升工程实施纲要》的通知 - 中华人民共和国教育部政府门户网站[EB/OL]. http://www.moe.gov.cn/srcsite/A12/s7060/201712/t20171206_320698.html.
- [3] 戚静. 高校课程思政协同创新研究[D]. 上海师范大学, 2020.
- [4] 教育部关于印发《高等学校课程思政建设指导纲要》的通知_教育_中国政府网[EB/OL]. http://www.gov.cn/zhengce/zhengceku/2020-06/06/content_5517606.htm.
- [5] 贺武华. 我国课程思政研究的回顾与展望[J]. 学校党建与思想教育, 2021(643): 26-30.
- [6] 李羽佳. “课程思政”网络教育平台建设的实践探索[J]. 学校党建与思想教育, 2020(627): 47-49.
- [7] 孙倩文, 谢泰, 齐悦, 等. 面向教员的课程思政资源网站开发与应用[J]. 信息与电脑(理论版), 2021(15): 107-109.
- [8] 胡晓丽, 王文燕. 基于微服务构架的计算机应用基础课程思政网站[J]. 电脑知识与技术, 2021, 17(31): 13-14,19.
- [9] 刘伟, 钟艳, 宾强, 等. 智能课程思政教学素材管理系统的设计与研发[J]. 电脑知识与技术, 2021, 17(31): 82-84.
- [10] 赵营颖, 高志宇, 曹莉. 基于大数据的课程思政教学资源推荐系统研究[J]. 信息与电脑(理论版), 2021, 33(18): 146-148.
- [11] 洪雅敏. 在线课程思政资源智能推荐系统设计[J]. 鄂州大学学报, 2023, 30(2): 103-105.
- [12] 何胜, 巢海远, 史航. 基于画像建模的大学生精准课程思政及应用研究[J]. 电脑知识与技术, 2023, 19(12): 64-66.
- [13] HOPFIELD J J. Hopfield network[J/OL]. Scholarpedia, 2007, 2(5). http://www.scholarpedia.org/article/Hopfield_network
- [14] JORDAN M I. Chapter 25 - Serial Order: A Parallel Distributed Processing

参考文献

- Approach[M]//DONAHOE J W, PACKARD DORSEL V. Advances in Psychology: 卷 121. North-Holland, 1997: 471-495.
- [15] ELMAN J L. Finding Structure in Time[J]. Cognitive Science, 1990, 14(2): 179-211.
- [16] 杨丽, 吴雨茜, 王俊丽, 等. 循环神经网络研究综述[J]. 计算机应用, 2018, 38(S2): 1-6,26.
- [17] SUN X, FU Y, LIU T. A hybrid ACO algorithm for capacitated vehicle routing problems[C]//2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). 2017: 510-514.
- [18] SHANI G, HECKERMAN D, BRAFMAN R I. An MDP-Based Recommender System[J]. Journal of Machine Learning Research, 2005, 6: 1265–1295
- [19] KONDA V, TSITSIKLIS J. Actor-Critic Algorithms[C]//Advances in Neural Information Processing Systems: 卷 12. MIT Press, 1999: 1008-1014.
- [20] 杨博, 赵鹏飞. 推荐算法综述[J]. 山西大学学报(自然科学版), 2011, 34(03): 337-350.
- [21] 余力, 杜启翰, 岳博妍, 等. 基于强化学习的推荐研究综述[J]. 计算机科学: 1-28.
- [22] KOREN Y, BELL R, VOLINSKY C. Matrix Factorization Techniques for Recommender Systems[J]. Computer, 2009, 42(8): 30-37.
- [23] BOBADILLA J, ORTEGA F, HERNANDO A, 等. Recommender systems survey[J]. Knowledge-Based Systems, 2013, 46: 109-132.
- [24] 周望. 基于机器学习的推荐系统关键技术及其应用研究[D]. 电子科技大学, 2020.
- [25] 项亮. 推荐系统实践[M]. 人民邮电出版社, 2012.
- [26] LI L, CHU W, LANGFORD J, 等. A contextual-bandit approach to personalized news article recommendation[C]//Proceedings of the 19th international conference on World wide web. New York, NY, USA: Association for Computing Machinery, 2010: 661-670.
- [27] CHEN M, BEUTEL A, COVINGTON P, 等. Top-K Off-Policy Correction for a REINFORCE Recommender System[C]//Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. New York, NY, USA: Association for Computing Machinery, 2019: 456-464.
- [28] ZHENG G, ZHANG F, ZHENG Z, 等. DRN: A Deep Reinforcement Learning

参考文献

- Framework for News Recommendation[C]//Proceedings of the 2018 World Wide Web Conference. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018: 167-176.
- [29] WILLMOTT C J, MATSUURA K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance[J]. Climate Research, 2005, 30(1): 79-82.
- [30] BUCKLAND M, GEY F. The relationship between Recall and Precision[J]. Journal of the American Society for Information Science, 1994, 45(1): 12-19.
- [31] BAWDEN D, HOLTHAM C, COURTNEY N. Perspectives on information overload[J]. Aslib Proceedings, 1999, 51(8): 249-255.
- [32] DULAC-ARNOLD G, EVANS R, VAN HASSELT H, 等. Deep Reinforcement Learning in Large Discrete Action Spaces[J]. arXiv.org, 2015.
- [33] ZHAO X, ZHANG L, XIA L, 等. Deep Reinforcement Learning for List-wise Recommendations[J]. arXiv:1801.00209, 2019.
- [34] ZHAO X, XIA L, ZHANG L, 等. Deep reinforcement learning for page-wise recommendations[C]//Proceedings of the 12th ACM Conference on Recommender Systems. New York, NY, USA: Association for Computing Machinery, 2018: 95-103.
- [35] ZHAO X, ZHANG L, DING Z, 等. Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York, NY, USA: Association for Computing Machinery, 2018: 1040-1048.
- [36] ZHAO X, XIA L, ZOU L, 等. Whole-Chain Recommendations[C]//Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020: 1883-1891.
- [37] WILLIAMS R J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning[M]//SUTTON R S. Reinforcement Learning. Boston, MA: Springer US, 1992: 5-32.
- [38] SCHAUL T, QUAN J, ANTONOGLOU I, 等 . Prioritized Experience Replay[M]. arXiv, 2016.
- [39] ZHANG S, SUTTON R S. A Deeper Look at Experience Replay[M]. arXiv, 2018.
- [40] 王继成, 高珍. 软件需求分析的研究[J]. 计算机工程与设计, 2002, 08: 18-21.

参考文献

- [41] 李平. 基于 JSP 的动态网页开发技术[J]. 微计算机信息, 2009, 25(21): 108-110.
- [42] 李岳峰, 王奉涛. 基于 JSTL 的 Web 应用研究[J]. 电脑知识与技术, 2006, 35: 43,55.
- [43] 王毅, 周峰, 孙更新. J2EE 经典案例设计与实现[M]. 北京: 电子工业出版社, 2017.
- [44] 吴小东. 基于 Dubbo 和 Zookeeper 的学校自助证明打印系统设计与实现[D]. 兰州大学, 2021.
- [45] SONI R. Introduction to Nginx Web Server[M]//SONI R. Nginx: From Beginner to Pro. Berkeley, CA: Apress, 2016: 1-15.
- [46] CHEN S, TANG X, WANG H, 等. Towards Scalable and Reliable In - Memory Storage System: A Case Study with Redis[C]//2016 IEEE Trustco m/BigDataSE/ISPA. 2016: 1660-1667.
- [47] EDDELBUETTEL D. A Brief Introduction to Redis[M]. arXiv, 2022.
- [48] 黄玉萍, 梁炜萱, 肖祖环. 基于 TensorFlow 和 PyTorch 的深度学习框架对比分析[J]. 现代信息科技, 2020, 4(4): 80-82,87.
- [49] KIRILL D. PyTorch vs TensorFlow-spotting the difference[J/OL]. Medium-Towards Data Science, 2017. <https://towardsdatascience.com/pytorch-vs-tensorflow-spotting-the-difference-25c75777377b>
- [50] CHEN X, LIU Q, YAO S, 等. Hardware Trojan Detection in Third-Party Digital Intellectual Property Cores by Multilevel Feature Analysis[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(7): 1370-1383.