

# 记忆与认知融合的个性化 OJ 习题推荐方法

金天成<sup>1),2)</sup> 窦亮<sup>2)</sup> 肖春芸<sup>2)</sup> 张伟<sup>1),2)</sup> 周爱民<sup>1),2)</sup>

<sup>1)</sup>(华东师范大学上海智能教育研究院 上海 200062)

<sup>2)</sup>(华东师范大学计算机科学与技术学院 上海 200062)

**摘 要** 面对程序设计 OJ (Online Judge, 在线判题) 平台上的大量习题, 学生选题时往往带有盲目性, 不利于学生快速提高程序设计能力. 现有的习题推荐方法往往忽视了认知心理学与学生答题行为的联系, 没有充分关注导致学生信息认知过程具有差异性的因素对解题的影响. 本文基于认知心理学提出一种记忆与认知融合的 OJ 习题推荐方法, 该方法结合认知心理学中关于工作记忆、长时记忆、外部信息以及非连续性认知发展的工作, 通过如下方式考虑这 4 种因素对学生答题行为的影响: 首先, 以 OJ 习题中 4 个会对学生的工作记忆造成挑战的要素作为习题特征的一部分来辅助推荐方法估计习题的难度, 分别为习题描述词数、输入格式词数、输出格式词数和是否为英文题干; 然后, 使用 4 个要素 (相同知识的学习时间间隔、相同知识的学习次数、学习时间间隔和总体学习次数) 建模学生长时记忆的遗忘与学习行为, 拟合学生因这两种行为导致的知识状态变化, 帮助推荐方法更准确地估计学生的知识点和编程语言掌握程度; 接着, 将 OJ 习题加入本文所构造的程序设计领域知识图谱, 并在习题与其所考查的知识点之间建立关联, 利用知识图谱嵌入方法将习题编码为一个融入了外部知识点相关性信息的向量作为习题的知识点表征, 令推荐方法能够处理知识点间复杂多样的关系; 最后, 提出了 2 种策略用以指导推荐方法处理非连续性认知发展 (指学生的知识状态在某个时间点突然发生较大变化) 的情况, 提升推荐结果的准确性, 第 1 种策略运用强化学习技术并以推荐方法生成的答题序列与未来真实答题序列的相似度作为奖励, 支持推荐方法生成与未来真实答题序列更一致的答题序列, 从而使推荐方法在给学生推荐下一道习题时可以考虑学生的知识状态在未来发生的突然变化, 第 2 种策略修改了高效用项集挖掘算法, 使其能够通过学生行为挖掘知识点之间的隐式后序关系, 帮助推荐方法预测学生在未来突然掌握或关注的知识点. 通过在业界知名 OJ 平台洛谷和 CodeForces 数据集上的实验, 表明该方法相较于目前常见的习题推荐算法具有更高的性能.

**关键词** 推荐系统; 认知心理学; 强化学习; 在线判题系统; 工作记忆; 长时记忆; 非连续性认知发展  
中图法分类号 TP181 DOI号 10.11897/SP.J.1016.2023.00103

## Personalized OJ Exercise Recommendation Method with Memory and Cognition Merging

JIN Tian-Cheng<sup>1),2)</sup> DOU Liang<sup>2)</sup> XIAO Chun-Yun<sup>2)</sup> ZHANG Wei<sup>1),2)</sup> ZHOU Ai-Min<sup>1),2)</sup>

<sup>1)</sup>(Shanghai Institute of AI for Education, East China Normal University, Shanghai 200062)

<sup>2)</sup>(School of Computer Science and Technology, East China Normal University, Shanghai 200062)

**Abstract** Faced with a large number of exercises on the programming OJ (Online Judge) platform, students are usually mindless when choosing exercises, which is not conducive to helping students quickly improve their programming ability. Existing exercise recommendation methods often ignore the relationship between cognitive psychology and students' exercise answering behavior, pay little attention to the in-

收稿日期: 2021-09-30; 在线发布日期: 2022-07-01. 本课题得到国家自然科学基金青年科学基金 (No.61907015) 和上海市科学技术委员会高新技术领域项目 (20511102502) 资助. 金天成, 博士研究生, 主要研究领域为教育数据挖掘、推荐系统. E-mail: 52205901026@stu.ecnu.edu.cn. 窦亮 (通信作者), 博士, 副教授, 中国计算机学会 (CCF) 会员, 主要研究领域为软件方法和技术、智能教育. E-mail: ldou@cs.ecnu.edu.cn. 肖春芸 (通信作者), 硕士, 讲师, 主要研究领域为程序设计和算法. Email: cyxiao@cs.ecnu.edu.cn. 张伟, 博士, 副研究员, 中国计算机学会 (CCF) 会员, 主要研究领域为数据挖掘. 周爱民, 博士, 研究员, 中国计算机学会 (CCF) 会员, 主要研究领域为演化计算与最优化、机器学习、智能教育.

fluence of factors that lead to differences in students' information cognitive processes on exercise answering. This paper proposes a personalized OJ exercise recommendation method with memory and cognition merging based on cognitive psychology, the method combines the related works of working memory, long term memory, external information and discontinuous cognitive development in cognitive psychology, and considers the influence of these four factors on students' exercise answering behavior in following ways: Firstly, taking four elements in OJ exercise that challenge students' working memory as part of exercise features to assist the recommendation method to estimate difficulty of exercise, including the number of words in exercise description, the number of words in input format, the number of words in output format and the type of language. Then, using four elements (the time interval between students' learning of same knowledge points, the number of repeated learning of same knowledge points, the time interval between sequential learning and the number of repeated learning) to model forgetting and learning behaviors of students' long term memory and fitting the changes of students' knowledge states caused by these two behaviors to help the recommendation method estimate students' mastery level of knowledge points and programming language more accurately. Thirdly, adding OJ exercises to the programming knowledge graph constructed in this paper and establishing relationships between exercises and knowledge points examined by the exercises. Applying knowledge graph embedding method to encode exercises into vectors incorporating external correlation information between knowledge points as the knowledge point representation of exercises, in order that the recommendation method is able to deal with complex and diverse relationships between knowledge points. Finally, proposing two strategies to guide the method to handle the situation of discontinuous cognitive development (discontinuous cognitive development refers to the sudden great change of students' knowledge states at some time) and improve the accuracy of recommendation results. The first strategy applies reinforcement learning technology and utilizes the similarity between sequence of exercises answered by students in the future and sequence generated by recommendation method as reward to support recommendation method to generate the sequence which is more consistent with sequence of exercises answered by students in the future, so that sudden changes of students' knowledge states in the future can be well considered when recommending the next exercise. The second strategy modifies the high utility itemsets mining algorithm to mine implicit postorder relationships between knowledge points through students' behaviors and assist recommendation method to predict the knowledge points that students may suddenly master or pay attention to in the future. This paper conduct extensive experiments on datasets of two well-known OJ platforms named Luogu and CodeForces, experimental results show that our approach can achieve higher performance than the state-of-the-art methods.

**Keywords** recommender system; cognitive psychology; reinforcement learning; online judge system; working memory; long term memory; discontinuous cognitive development

## 1 引 言

OJ (Online Judge, 在线判题) 是一种基于 Web 的在线判题平台, 包含大量计算机程序设计习题, 学生根据习题描述编写程序进行提交, OJ 自动检测提交代码的正确性并反馈结果. OJ 不但用于国内外程序设计竞赛, 也已广泛用于程序设计教学与训练中. 然而现有各大 OJ 平台上往往有大量计算机程序设计习题, 题目类型多样且难度跨越很大, 导致学

生在选择习题的过程中容易存在盲目性, 出现“信息过载”问题. 因此, 需要向学生推荐适合的习题, 帮助学生快速提高程序设计能力.

近年来, 研究者提出了许多习题推荐方法. 这些方法主要通过传统推荐算法、认知诊断、知识追踪或强化学习来实现习题推荐<sup>[1,2]</sup>. 有部分研究者应用传统推荐算法给学生提供恰当的学习资源 (如习题、课程)<sup>[3-10]</sup>, 这类方法将学习资源推荐类比为传统电子商务领域的推荐, 将学生视为用户, 学习

资源（如习题、课程）视为商品，学生的学习行为记录（如答题记录、上课记录）视为用户的消费行为记录，所以学习资源推荐任务就转变成了商品推荐任务，向目标学生推荐相似学生解答过的习题。这种方法根据学生之间的共性给出推荐结果。

基于认知诊断的推荐方法<sup>[11,12]</sup>和基于知识追踪的推荐方法<sup>[13-16]</sup>通过对学生的知识状态进行建模从而实现习题推荐，所推荐的习题能提高学生的知识掌握程度。认知诊断<sup>[17-22]</sup>技术假设学生的知识状态在一定时间内固定不变，利用学生的历史答题情况和学生作答的每个习题所包含的知识点来评估学生目前的知识水平并预测学生在未来学习中的表现。Corbett 等人<sup>[23]</sup>认为学习知识是一个循序渐进的过程，并提出了知识追踪技术<sup>[23-29]</sup>，通过学生的历史学习数据，动态追踪学生知识水平的改变情况。基于认知诊断或知识追踪的推荐方法<sup>[11-16]</sup>可以大致分为两个步骤：首先使用认知诊断或知识追踪技术对学生的历史答题情况进行分析，得到学生对于每道习题的答对概率；然后将“未掌握”的习题（即答对概率不高的习题）推荐给学生，以帮助学生掌握薄弱知识点。进一步地，Huang 等人<sup>[1]</sup>认为这种仅根据习题答对概率大小进行习题推荐的做法存在局限性，在给学生推荐习题时应该还要考虑“复习与探索之间的权衡”、“难度的平滑性”和“学生的参与度”三个因素。因此，Huang 等人提出了基于强化学习的 DRER（Deep Reinforcement Learning Framework for Exercise Recommendation with Recurrent Manner）网络，将上述三个因素量化为三个不同的奖励机制并组成目标函数，使用这个目标函数来训练整个网络，实现了习题推荐的多元化。

虽然上述方法在习题推荐任务中取得了较好成果，但这些方法忽视了认知心理学与学生答题行为的联系，没有充分关注认知心理学中能够对学生答题行为造成影响的因素，包括工作记忆、长时记忆、外部信息和非连续性认知发展。本文认为在设计模型时应当考虑这些因素，理由如下：

工作记忆<sup>[30]</sup>是一种容量固定的短时记忆，已有科学研究证明学生工作记忆的容量大小会对其学习成绩造成影响<sup>[31-33]</sup>，所以可以考虑将工作记忆容量信息加入到学生知识水平建模的过程，使模型能够更准确地估计学生的知识水平。

长时记忆是能够保持几天到几年的记忆。它与工作记忆不同，后者只保持几秒到几小时。关于长时记忆的研究<sup>[34-36]</sup>表明，长时记忆的遗忘和巩固程度会对学生的成绩造成影响，并且随着时间的推移，

记忆遗忘率会上升，增加知识学习次数有助于学生巩固记忆。如果在模型中不考虑长时记忆的遗忘和巩固，那么学生的知识状态不会随着时间和知识学习次数的变化而变化。这种假设在现实中并不合理。因此，在模型中考虑长时记忆的遗忘与巩固对学生知识水平的影响，有利于更精确地评估学生的知识水平。

在认知心理学推理、决策与判断的相关工作<sup>[37]</sup>中经常强调，人相对于机器而言，最大的优势在于能够使用外部信息，而知识图谱是机器学习模型引入外部信息的常用媒介。在习题推荐模型中引入知识图谱，可以解决冷启动的问题，同时提升模型推荐结果的准确性与合理性。

现有工作或者假定学生的认知是固定不变的（如基于认知诊断的推荐方法），或者认定学生的认知发展是一个连续的、循序渐进的过程（如基于知识追踪的推荐方法）。但认知心理学的大量科学研究<sup>[38-40]</sup>表明认知发展不仅有连续性的，还有非连续性的。认知发展的非连续性指的是学生的知识状态在某个时间点突然发生了较大变化，即学生突然掌握了某些较难的知识点或者学习兴趣发生了较大改变。为了捕捉到学生知识状态的突然变化，提升模型推荐结果的准确性，使模型更符合认知理论，应当在模型中设计策略，以考虑学生的非连续性认知发展现象。

鉴于此，本文在 DRER 模型的基础上，针对 OJ 习题推荐任务提出记忆与认知融合的 MC-DRER（Memory and Cognition Merged DRER）模型。该模型结合认知心理学中工作记忆<sup>[31-33]</sup>、长时记忆<sup>[34-36]</sup>、外部信息<sup>[37]</sup>以及非连续性认知发展<sup>[38-40]</sup>的相关工作，考虑了这 4 种因素对学生答题行为的影响。主要创新包括：

（1）首次考虑了工作记忆对知识水平评估的影响，并针对 OJ 习题的特点，提炼了习题中 4 个会对学生的工作记忆造成挑战的要素，作为习题特征的一部分，分别为习题描述词数、输入格式词数、输出格式词数和是否为英文题干。若习题是用非母语描述的，并且习题描述词数、输入格式词数和输出格式词数越多，那么该习题就越会对学生的工作记忆产生挑战，习题也往往越难。因此，习题描述词数、输入格式词数、输出格式词数和是否为英文题干可以作为习题的特征，从工作记忆的角度反映习题的难度。本文通过实验证明了这 4 个要素与习题难度的关联性，并验证了这 4 个要素的加入能提升习题推荐模型的准确性。

(2) 基于 OJ 习题的特点, 首次在长时记忆的遗忘与学习机制中考虑了学生编程语言熟悉程度, 并结合认知心理学中遗忘曲线<sup>[34,35]</sup>和学习曲线<sup>[36]</sup>描述的人类大脑对长时记忆遗忘与学习的规律, 在相同知识的学习时间间隔和学习次数的基础上, 再选取了 2 个影响编程语言遗忘与学习的要素来建模学生的知识状态, 分别为学习时间间隔、总体学习次数. 当学生总体学习次数越大, 学生的编程语言熟练度也越高, 学生答题错误率就越低. 当学习时间间隔越大, 学生越会对编程语言感到生疏, 编程语言熟练度也越低, 学生答题错误率就越高. 本文通过实验验证了这 2 个要素、编程语言的熟悉程度和学生答题错误率四者之间的相关性, 并证明了本文所提出的遗忘与学习机制的优越性.

此外, 失误与猜对行为对于习题推荐模型也较关键. 若没有考虑学生的猜对行为, 模型会高估学生的知识水平; 若没有考虑失误行为, 模型会低估学生的知识水平. 本文首次探索并选取习题会造成学生失误与猜对的要素来拟合习题的失误率与猜对率, 并用这两个概率值对学生的知识状态进行提纯, 从而排除失误与猜对行为对评估知识点掌握程度的干扰. 所选取的 6 个要素分别为习题的学生作答正确率与提交作答正确率之差、习题描述词数、输入格式词数、输出格式词数、是否为英文题干以及习题考查的知识点, 选取这些要素的理由如下: 学生作答正确率与提交作答正确率的差较大, 就说明有较多学生在尝试解答该题时先做错后做对, 很可能存在失误行为. 题干越复杂, 学生越有可能对题意产生误解, 从而导致失误, 所以习题描述词数、输入格式词数、输出格式词数和是否为英文题干这几个要素会影响学生的失误行为. 某些知识点所涉及到的 OJ 习题存在简单暴力的解决方法, 比如动态规划的相关习题, 如果测试用例设计得不好, 学生虽然不会使用动态规划, 但用递归算法也能通过, 所以知识点也与学生的猜对行为有关. 本文通过实验证明了本文所提出的失误与猜对机制能提升习题推荐模型的准确性.

(3) 程序设计领域的知识点之间关系种类较多, 而现有习题推荐方法没有考虑知识点之间关系多样化的情况. 本文首次提出能够考虑知识点间复杂多样的关系的方法: 首先, 建立了专门应用于 OJ 习题推荐任务的程序设计领域知识图谱; 然后, 使用知识图谱嵌入方法将习题转为向量输入到习题推荐模型中. 该方法引入了外部信息, 能够提升习题推荐的准确性与合理性, 并且本文通过实验验证了这

两点.

(4) 首次考虑了学生认知发展的非连续性, 设计了两种策略用以指导模型处理非连续性认知发展的情况. 策略一: 基于强化学习技术并参考机器翻译中的 BLEU 评价指标设计了奖励函数, 支持推荐模型生成与未来真实习题交互序列更一致的习题交互序列, 从而使推荐模型在给学生推荐下一道习题时可以考虑学生的认知在未来发生的突然变化. 策略二: 修改了高效用项集挖掘算法<sup>[41]</sup>, 使其能够通过学生行为挖掘知识点之间的隐式后序关系, 帮助模型预测学生突然掌握或关注与之前作答习题毫无显式关联的知识点, 预知学生未来的认知突变. 本文通过实验证明了非连续性认知发展在学生学习程序设计的过程中是一种普遍现象, 并验证了两种策略的加入能提升习题推荐准确性.

## 2 相关工作及 OJ 介绍

### 2.1 相关工作

#### 2.1.1 传统推荐算法在学习资源推荐中的应用

Walker 等人<sup>[3]</sup>通过比较目标学生和其他人所提交的学习资源评价来搜索相似的其他学生并产生个性化推荐; Hsu 等人<sup>[4]</sup>分析学生的阅读记录, 使用协同过滤算法生成课程推荐分数为学生推荐感兴趣的课程; 文献[5-8]使用协同过滤实现了 OJ 习题推荐方法, 其中, Toledo 等人<sup>[5]</sup>在推荐方法中引入了模糊逻辑, Mao 等人<sup>[8]</sup>考虑了时间对学习兴趣的影响; Gong 等人<sup>[9]</sup>提出了基于异构信息网络与图卷积神经网络的 ACKRec 模型, 使用用户、课程、视频、教师和知识概念这五种实体创建异构信息网络, 并使用图卷积网络对每种实体进行表示学习从而实现知识点推荐; Shao 等人<sup>[10]</sup>提出 PLAN-BERT 模型来对学生未来的课程学习路径做一个完整的规划, 将学生的未来参考课程引入到 BERT 模型中, 来减少多时间片推荐任务的误差, 还使用学生的个人信息和课程信息构成学生和课程的特征向量, 取代通过随机方式生成的用户和课程向量, 提升模型的精度.

#### 2.1.2 认知诊断

教育心理学中的认知诊断通过学生历史答题记录评估学生的知识水平, 并预测学生未来答题表现<sup>[17]</sup>. 从不同角度看, 认知诊断模型可分为离散型和连续型, 也可分为一维技能型和多维技能型. 在认知诊断模型中, 目前应用最广泛的是一维连续型的项目反应理论 (IRT)<sup>[18]</sup>和多维离散型的有确定性输入、噪声和门 (DINA) 模型<sup>[19]</sup>.

IRT 通过学生在习题上的答题情况和习题难度推断出学生能力值, 并以这个单一的能力值表示学生的综合能力. 有不少后续的工作在 IRT 模型的基础上进行改进, 如 Reckase 等人<sup>[20]</sup>提出 MIRT 模型, 将学生的建模从一维能力值拓展到多维向量; Cheng 等人<sup>[21]</sup>提出 DIRT 模型, 挖掘习题文本及其与知识之间的关系来增强学生与习题的参数诊断过程. DINA 模型由二进制变量表示学生是否掌握知识点, 即它只能给出学生知识点的离散化掌握程度(知识点掌握或未掌握), 为了解决离散化问题, Zhu 等人<sup>[11]</sup>提出了 PMF-CD 模型, 用概率(0 到 1 之间的连续值)对学生的知识点掌握程度进行建模; Wang 等人<sup>[22]</sup>利用深度学习技术, 并结合认知诊断理论, 提出了 NeuralCD 模型, 分解出学生-因子特征与习题-因子特征, 并设计交互函数和应用单调性假设实现学生得分预测.

### 2.1.3 知识追踪

Corbett 等人<sup>[23]</sup>第一次使用知识追踪模型预测学生成绩, 称为贝叶斯知识追踪(BKT), 能够动态跟踪学生的知识水平; Zhang 等人<sup>[24]</sup>利用三分法的优势将学习过程分为三部分, 开发了三个学习状态模型, 在一定程度上模拟了学生知识状态的分布; Agarwal 等人<sup>[25]</sup>采用学生近期学习权重代替静态学习率获取学生的进步程度, 更适当地融合历史和最新数据. 深度知识追踪(DKT)由 Piech 等人<sup>[26]</sup>提出, 通过门控制循环单元(GRU)网络来建模学生的知识状态; Zhang 等人<sup>[27]</sup>发现学生行为特征的重要性, 通过特征拼接和自编码器降维等功能改进了 DKT 模型, 并取得了良好的效果. Minn 等人<sup>[28]</sup>注意到知识追踪缺乏个性化的特点, 因而结合学生聚类改进了 DKT 算法. Liu 等人<sup>[29]</sup>将 attention 机制融入 DKT 模型, 实现学生的成绩预测.

已有学者提出考虑遗忘和学习行为可以帮助知识追踪模型更准确地评估学生的知识状态. 关于长时记忆的研究<sup>[34-36]</sup>表明, 随着时间的推移, 学生记忆保留率呈指数下降, 此外, 增加知识学习次数有助于学生避免遗忘. 如果在知识追踪模型中不考虑遗忘和学习行为, 那么学生的知识状态不会随着时间和知识学习次数的变化而变化. 这种假设在现实中并不合理. Zhang 等人<sup>[42]</sup>在设计模型时考虑了遗忘和学习行为, 为每一道习题在被正确解答和错误解答的两种情况下各分配了一个的遗忘和学习向量进而对学生的知识水平进行评估, 然而学生的遗忘与学习行为会受时间等因素所影响, 并不能用训练而得的简单固定值进行描述; Khajah 等人<sup>[43]</sup>通过相

同知识的学习次数来扩展 BKT 模型以估量遗忘率; Qiu 等人<sup>[44]</sup>以天为单位计算相同知识的学习时间间隔来建模学生的遗忘行为; Pelánek 等人<sup>[45]</sup>通过以时间为自变量的分段函数来建模记忆激活系数, 并使用记忆激活系数估计学生的知识状态; Huang 等人<sup>[46]</sup>选用相同知识的学习次数和双曲线形式定义记忆因子, 相同知识的学习时间间隔和指数曲线形式定义遗忘因子, 从而建模学生的遗忘与学习行为; Shen 等人<sup>[47]</sup>首先通过当前所做习题、解题时间和作答情况来建模学生当前的学习表现, 然后使用前一刻的学习表现、间隔时间、当前学习表现和前一刻知识状态来建模学习收益, 再利用前一刻的知识状态、当前的学习收益和间隔时间来拟合遗忘率, 通过当前学习收益、遗忘率和前一刻知识状态来计算当前知识状态, 最后用当前的知识状态来预测学生下一时刻的作答情况.

### 2.1.4 强化学习在学习资源推荐中的应用

在教育领域, 有不少工作将强化学习技术应用在学习资源推荐任务中: Zhang 等人<sup>[48]</sup>提出 HRL-NASR 模型应用于课程推荐任务, 它指出了基于 attention 机制的序列推荐模型中存在的一个问题, 那就是当一个学生的历史学习兴趣很复杂时, 对目标推荐课程有贡献的课程的影响效果会被不同种类的其他历史学习课程所稀释, 因而导致 attention 机制表现变差, HRL-NASR 使用强化学习技术与目标课程相关度很小或者贡献度很低的课程去除, 从而提升课程推荐的效果; Liu 等人<sup>[49]</sup>提出 CSEAL 模型, 使用学生在学习一系列课程前的考试成绩以及学完课程后的考试成绩之差作为基于 Actor-Critic 的课程推荐模型奖励值来训练网络, 从而使学生在学习系统所推荐的课程时, 能够让学生在下次考试时考的分数更高; Huang 等人<sup>[1]</sup>认为在给推荐习题时应该要考虑“复习与探索之间的权衡”、“难度的平滑性”和“学生的参与度”这三个因素. “复习与探索之间的权衡”是指推荐系统不仅要利用学生目前的学习兴趣进行推荐, 还要让学生去探索新的知识点; “难度的平滑性”指连续推荐的题目难度不会变化过大; “学生的参与度”从提高学生参与度的角度出发, 如果当前学生的答题正确率过低, 就推荐简单的题目, 如果学生的答题正确率过高, 就推荐难题. DRER 模型将上述三个因素量化为三个不同的奖励机制并组成目标函数, 使用这个目标函数来训练 DQN(Deep Q-Network), 实现了习题推荐的多元化. 基于强化学习的习题推荐方法可以加入人为指定的长期推荐策略从而指导

模型向学生推荐习题.

## 2.2 OJ介绍

Online Judge 是一种基于 Web 的在线判题平台, 用来检测计算机程序源代码的正确性. OJ 平台上包含了大量计算机程序设计习题, 图 1 是洛谷 OJ 平台的部分习题列表, 目前该平台上有 66.14% 的习题被学生标注了知识点. 学生在 OJ 平台上通过解答习题来学习程序设计的相关知识.

-	P1230	智力大冲浪	贪心 排序 快速排序
-	P1231	教铺的组成	网络流
-	P1232	[NOI2013] 数的计数	递推 广度优先搜索 BFS 深度优先搜索 DFS
-	P1233	木棍加工	动态规划, dp 贪心
-	P1234	小A的口头禅	搜索

图 1 洛谷 OJ 平台部分习题

学生需要阅读习题题干 (如图 2 所示) 来理解习题, 运用计算思维求解问题, 使用 OJ 平台支持的编程语言 (如 C、Java、Python 等) 编写程序, 将程序代码提交到 OJ 平台上, 以这种方式来解答习题. 平台通过预先设计的测试数据来检验程序的正确性与时空效率, 直接反馈给用户. 习题题干包含三部分, 如图 2 所示, 分别为习题描述、输入格式和输出格式. 与传统学习平台的习题不同, OJ 习题是开放性应用实践问题. 一方面, 对同一道题, 学生们可以采用不同的算法解决, 如图 2 所示的习题既可以用“动态规划”解决, 也可以用“贪心”和“二分查找”解决. 另一方面, OJ 习题很大一部分是来源于计算机实践的抽象, 常借助各种风趣幽默的故事进行描述, 避免学生直接从题干中看出所考查的知识点, 如图 2 所示的习题对应的知识点包括“动态规划”、“贪心”和“二分查找”, 但在题干中没有直接出现与这三个知识点相关的名词. 即便是知识点标签相同的习题, 习题描述所使用的故事也会完全不同. 作为对比, 图 3 是一道传统学科数学的中考题, 题干直接包含了关键词“圆”、“割线”和“半径”, 即习题所考查的知识点.

## 3 问题定义

在一个 OJ 系统中, 假设有  $|U|$  个学生、 $|E|$  道习题和  $|K|$  个知识点. 每个学生  $u (u \in U)$  的历史答题序列为  $i^u = \{(e_1, p_1), (e_2, p_2), \dots, (e_T, p_T)\}$ ,  $e_i (e_i \in E)$  是学

### P1020 [NOIP1999 普及组] 导弹拦截

#### 题目描述

某国为了防御敌国的导弹袭击, 发展出一种导弹拦截系统. 但是这种导弹拦截系统有一个缺陷: 虽然它的第一发炮弹能够达到任意的高度, 但是以后每一发炮弹都不能高于前一发的高度. 某天, 雷达捕捉到敌国的导弹来袭. 由于该系统还在试用阶段, 所以只有一套系统, 因此有可能不能拦截所有的导弹.

输入导弹依次飞来的高度 (雷达给出的高度数据是  $\leq 50000$  的正整数), 计算这套系统最多能拦截多少导弹, 如果要拦截所有导弹最少要配备多少套这种导弹拦截系统.

#### 输入格式

1 行, 若干个整数 (个数  $\leq 100000$ )

NOIP 原题数据规模不超过 2000.

#### 输出格式

2 行, 每行一个整数, 第一个数字表示这套系统最多能拦截多少导弹, 第二个数字表示如果要拦截所有导弹最少要配备多少套这种导弹拦截系统.

图 2 OJ 习题实例

11. (沈阳市) 如图,  $PA$  切  $\odot O$  于点  $A$ ,  $PBC$  是  $\odot O$  的割线且过圆心,  $PA=4$ ,  $PB=2$ , 则  $\odot O$  的半径等于 ( )

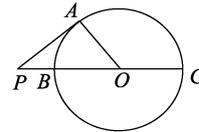


图 3 数学中考习题实例

生在  $t$  时刻解答的习题,  $p_t$  表示学生解答情况,  $p_t = 1$  代表学生回答正确,  $p_t = 0$  代表学生回答错误.

已知知识图谱  $g = (\mathcal{E}, R)$ , 其中,  $\mathcal{E}$  为实体集合,  $R$  为关系集合, 并且  $(E \cup K) \subseteq \mathcal{E}$ .

对于每道习题  $e \in E$ , 用四元组  $e = (c, o, k, d)$  进行表示, 其中,  $c$  代表习题题干的内容文本, 表示为一组词语序列,  $o$  代表习题中会对学生的工作记忆造成挑战的要素,  $k \in K$  代表习题所考查的知识点,  $d$  代表习题的难度, 习题的难度取决于学生作答错误率、提交作答错误率和区分度.

本文将习题推荐的过程形式化为马尔可夫决策过程 (Markov Decision Process, MDP), 对马尔可夫决策过程中的四种要素, 状态集合  $S$ 、行动集合  $A$ 、奖励函数  $R$  和状态转移函数  $\Gamma$ , 进行定义:

状态集合  $S$ :  $S$  代表学生的状态空间. 状态  $s_t^u \in S$  由学生  $u$  在  $t$  时刻及之前的历史答题序列和知识图谱所构成, 即  $s_t^u = [i_t^u, g]$ .

行动集合  $A$ :  $A$  代表行动空间, 包含 OJ 平台中所有的习题. OJ 习题推荐系统在状态  $s_t^u$  下采取了行动  $a_t^u \in A$  相当于给学生  $u$  推荐了习题  $e_{t+1}$ .

奖励函数  $R$ :  $R$  是奖励函数,  $r_t^u = R(s_t^u, a_t^u)$  代表

的是 OJ 习题推荐系统在状态  $s_t^u$  下采取行动  $a_t^u$  所获得的及时奖励, 奖励值的大小是由专家制定的推荐策略所决定的。

状态转移函数  $\Gamma$ : 当 OJ 习题推荐系统在状态  $s_t^u$  下采取行动  $a_t^u$  之后, 状态转移函数  $\Gamma$  能将当前的状态从  $s_t^u$  转移为  $s_{t+1}^u$ , 其中,  $s_{t+1}^u = [r_{t+1}^u, g]$ 。

本文的目标就是要找到一个向学生推荐习题的最优策略  $\pi: S \rightarrow A$ 。当学生状态为  $s_t^u$  时, OJ 习题推荐系统能够根据策略  $\pi$  从习题集合  $E$  中选取一个习题  $e_{t+1}$  推荐给学生  $u$ , 使得整个推荐过程所累积的奖励值最大化。

## 4 方法描述

本文在 DRER 模型基础上, 针对 OJ 习题推荐任务, 结合认知心理学中工作记忆<sup>[31-33]</sup>、长时记忆<sup>[34-36]</sup>、外部信息<sup>[37]</sup>和非连续性认知发展<sup>[38-40]</sup>的相关工作提出了记忆与认知融合的 MC-DRER 模型, 整体架构如图 4 所示。为了便于阅读, 后文从单个学生  $u$  的视角来描述推荐方法, 将第 3 节中出现的上标  $u$  从公式中去除。

### 4.1 习题编码

当前的习题推荐工作往往通过随机向量来表征习题, 这种编码方式无法很好地表达习题之间的相关性, DRER 选取习题所考查的知识点向量、习题的学生作答错误率和习题语义向量作为习题的特征进行编码, 使习题推荐模型在性能上有了一定的提升。其中, 习题语义向量由习题题干中每个词的词向量经过双向 LSTM 网络和池化操作后获得。

DRER 将习题语义向量作为习题特征的一部分, 试图通过习题题干中的词挖掘习题所考查的知识点以及习题之间相关性, 但如前所述, OJ 的命题风格使得题干中的词与习题所考查的知识点之间关联较少, 并且较难通过题干中的词有效挖掘习题之间相关性。因此, 仅选取习题语义向量表征习题题干则不够全面。工作记忆<sup>[30]</sup>是一种容量固定的短时记忆, 已有科学研究证明学生工作记忆的容量大小会对其数学和阅读理解成绩造成影响<sup>[31-33]</sup>, 而在 OJ 习题解答过程中, 学生需要阅读并理解题干所描述的问题, 运用计算思维来设计程序对问题进行求解, 所以 OJ 的程序设计习题与数学和阅读理解高度相关, 可以考虑将学生的工作记忆容量信息加入到学生知识水平建模的过程中, 但学生的工作记忆容量信息又难以从真实数据集中获取。因此, 本文结合认知心理学中研究工作记忆的相关工作<sup>[31-33]</sup>, 考虑了习题题干中 4 个会对学生的工作记忆造成挑战的

要素: 习题描述词数、输入格式词数、输出格式词数以及是否为英文题干, 用这四个要素作为习题特征的一部分, 这四个要素的选取原因分析如下:

(1) 习题描述词数: 学生工作记忆的容量会对其阅读理解成绩造成影响<sup>[31,32]</sup>。因为学生在阅读文章时会使用其工作记忆对文章进行记忆, 工作记忆容量小的学生在阅读长文章后半部分的内容时, 会对前半部分的内容产生遗忘, 所以长度越大的文章就越会对学生的工作记忆产生挑战。同样地, OJ 习题也同样适用于这个道理, 习题描述越长的习题越会对学生的工作记忆产生挑战。因此, “习题描述词数”可以从工作记忆的角度反映习题的难度。

(2) 输入格式词数、输出格式词数: 研究表明学生工作记忆的容量会对其数学成绩造成影响<sup>[33]</sup>。当一个数学公式或者函数的因变量与自变量越多时, 它就越复杂, 越难以记忆和运用, 会对学生的工作记忆产生挑战。同样地, 可以将程序看作一个函数或者公式, 输入变量与输出变量越多, 它就越复杂, 也就越难。因此, 可以将“输入格式词数”与“输出格式词数”作为 OJ 习题特征的一部分, 从而反映习题的难度。

(3) 是否为英文题干: 学生大脑每秒钟能为学生提供 5-9 个记忆单元来存放工作记忆, 但是每个单元能够存放母语的容量要大于非母语, 这也就是大多数中国学生看中文时能每秒阅读一句话, 而看英语时只能每秒阅读一个词的原因。习题的题干如果是用英文书写的, 那么势必会加大难度, 对学生的工作记忆造成挑战。因此, 可以将“是否为英文题干”作为题目的特征, 反映题目的难度。

此外, 针对有少数习题, 其题干内容较简单, 但实际难度较大的情况, 本文还选用了习题的学生作答错误率、提交作答错误率和区分度作为特征来评价习题的难度。其中, 习题区分度表示一道习题对于学生的区分能力, 具有良好区分度的习题, 能够将不同水平的学生区分开, 即在具有良好区分度的试题上, 能力强的学生能够得高分, 而能力弱的学生只能得到低分。根据教育心理学理论, 习题  $e_i$  的区分度  $b_i$  等于在解答过习题  $e_i$  的学生中, 总体习题作答正确率最高的 27% 学生与最低的 27% 学生, 其在习题  $e_i$  上的正确率之差。

综上, 本文选用习题描述词数  $w_i^{desc}$ 、输入格式词数  $w_i^{input}$ 、输出格式词数  $w_i^{output}$ 、是否为英文题干  $l_i$  (如果是英文题干,  $l_i$  就为 1, 否则,  $l_i$  就为 0)、学生作答错误率  $r_i^a$ 、提交作答错误率  $r_i^s$ 、区分度  $b_i$ 、习题所考查的知识点  $k_i$  的表征向量  $\mathbf{k}_i$  (第 4.3 节中

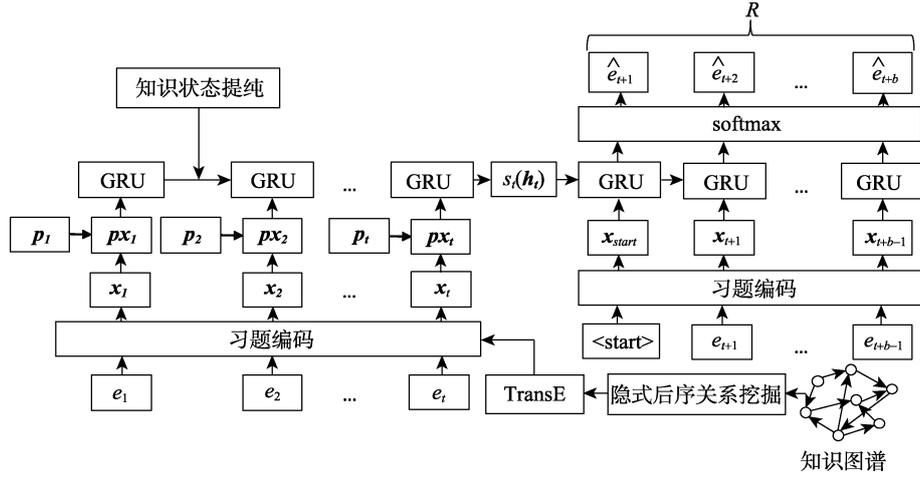


图 4 MC-DRER 整体架构

介绍知识点向量的由来)、习题语义向量  $v_t^c$  来对习题  $e_t$  进行编码, 构建习题  $e_t$  的特征向量  $x_t$ , 如公式 (1) 所示:

$$x_t = [w_t^{desc}, w_t^{input}, w_t^{output}, l_t, r_t^a, r_t^s, b_t] \oplus k_t \oplus v_t^c \quad (1)$$

其中,  $\oplus$  表示向量拼接操作.

#### 4.2 知识状态提纯

考虑学生的遗忘、学习、失误和猜对行为可以帮助模型更准确地评估学生的知识水平, 从而提升习题推荐的准确性. 已有学者认识到遗忘和学习这两种行为对知识水平评估的重要性, 在设计模型时考虑了这两种行为. 这些工作可以按照建模遗忘和学习行为所使用的外界要素大致分为四类: 无外界要素<sup>[42]</sup>; 相同知识的学习次数<sup>[43]</sup>; 相同知识的学习时间间隔<sup>[44]</sup>; 相同知识的学习次数和相同知识的学习时间间隔<sup>[45,46]</sup>. 但是程序设计习题不同于其它种类的习题, 想要答对程序设计习题不仅需要掌握习题所考查的知识点, 而且需要熟练运用编程语言. 因此, 在建模时不应该只考虑相同知识的学习次数与时间间隔, 还需考虑学生学习时间间隔与总体学习次数以反映学生编程语言熟悉程度. 对于失误与猜对行为, 在认知诊断领域, 同样也有学者<sup>[17,21]</sup>认识到这两种行为的重要性, 通过专家来标注每道题目的失误率与猜对率进而评估学生的知识水平. 但在真实场景下, 绝大多数在线学习平台 (包括 OJ) 没有这样的失误率与猜对率数据, 需要探索习题会造成学生失误与猜对的要素, 通过这些要素来拟合习题的失误率与猜对率.

DRER 在得到习题  $e_t$  的特征向量  $x_t$  以后, 根据学生在习题  $e_t$  上的答题结果  $p_t$ , 使用公式 2 对特征向量  $x_t$  进行处理, 得到考虑了答题结果的习题向量  $px_t$ .

$$px_t = \begin{cases} x_t \oplus \theta, & p_t = 1 \\ \theta \oplus x_t, & p_t = 0 \end{cases} \quad (2)$$

其中,  $\theta = [0, 0, \dots, 0]$  是与  $x_t$  同维度的“零”向量,  $\oplus$  表示向量拼接操作.

将  $t-1$  时刻学生的知识状态向量  $h_{t-1}$  和  $t$  时刻学生所做习题的向量  $px_t$  输入到 GRU 网络中, 从而得到  $t$  时刻的学生知识状态向量  $h_t$ , 如公式 3 所示, 其中,  $\Phi_{gru}$  为 GRU 网络的相关参数. 此外, 本文使用学生知识状态向量  $h_t$  来表征学生状态  $s_t$ .

$$h_t = GRU(h_{t-1}, px_t; \Phi_{gru}) \quad (3)$$

DRER 的方法没有考虑学生的遗忘、学习、失误和猜对行为, 会导致对学生的知识状态的评估不准确. 如果没有考虑学生的遗忘、猜对行为, 那么模型就会高估学生的知识水平; 如果没有考虑学生的学习、失误行为, 那么模型就会低估学生的知识水平.

本文提出的 MC-DRER 模型选取了在  $t$  时刻与学生的遗忘和学习行为相关的 4 个要素: 学习时间间隔  $ST_t$ 、相同知识的学习时间间隔  $RT_t$ 、总体学习的次数  $LT_t$  以及相同知识的学习次数  $KT_t$ , 根据认知心理学中遗忘曲线<sup>[34,35]</sup>的相关研究所提出的指数曲线形式描述学生知识水平下降与时间间隔的关系, 如公式 (4)、(5) 所示:

$$FK_t = e^{-\frac{RT_t}{S_k}} \quad (4)$$

$$FL_t = e^{-\frac{ST_t}{S_L}} \quad (5)$$

其中,  $FK_t$  和  $FL_t$  是遗忘因子, 分别描述了学生知识水平的下降与相同知识的学习时间间隔  $RT_t$  和学习时间间隔  $ST_t$  的关系, 表明学生在  $t$  时刻对知识的遗忘程度.  $S_k$  和  $S_L$  是控制记忆程度的可训练参数.

根据学习曲线的相关研究<sup>[36]</sup>所提出的双曲线形

式描述知识水平增长与学生学习次数的关系, 如公式(6)、(7)所示:

$$LK_t = \frac{(D_K + 1)KT_t}{KT_t + r_K} \quad (6)$$

$$LL_t = \frac{(D_L + 1)LT_t}{LT_t + r_L} \quad (7)$$

其中,  $LK_t$  和  $LL_t$  是记忆因子, 分别描述了学生知识水平增长与相同知识的学习次数  $KT_t$  和总体学习的次数  $LT_t$  的关系, 表明学生在  $t$  时刻经过一定次数练习后的知识水平增长幅度.  $D_K$ 、 $D_L$ 、 $r_K$  和  $r_L$  是可训练参数, 分别控制了知识水平增长的幅度和系数.

此外, 根据程序设计教师的建议, 考虑了 6 个与学生在解答习题  $e_t$  过程中的失误、猜对行为有关的要素: 习题  $e_t$  的学生作答正确率  $RS_t$  与提交作答正确率  $RA_t$  之差、习题描述词数  $w_t^{desc}$ 、输入格式词数  $w_t^{input}$ 、输出格式词数  $w_t^{output}$ 、是否为英文题干  $l_t$  以及习题考查的知识点  $k_t$ , 用这 6 个要素来估计习题的失误率与猜对率, 如公式(8)、(9)、(10)所示:

$$ER_t = \text{sigmoid}(D_t^T W_s + b_s) \quad (8)$$

$$GR_t = \text{sigmoid}(k_t^T W_g + b_g) \quad (9)$$

$$D_t = [RS_t - RA_t, w_t^{desc}, w_t^{input}, w_t^{output}, l_t] \quad (10)$$

其中,  $ER_t$  的含义为习题  $e_t$  的失误率,  $GR_t$  的含义为习题  $e_t$  的猜对率.  $W_s$  和  $W_g$  为可训练的权重向量, 形状分别为  $5 \times 1$ 、 $d_k \times 1$ ,  $d_k$  为知识点  $k_t$  的向量  $k_t$  的维度.  $b_s$  和  $b_g$  为可训练的偏置项参数.  $\text{sigmoid}$  函数用来做归一化.

学生作答正确率与提交作答正确率的差较大, 就说明有较多学生在尝试解答该题时先做错了, 之后做对了, 这就说明很可能存在失误行为, 所以学生作答正确率与提交作答正确率的差与失误率相关. 题干越复杂, 学生越有可能误解题意导致失误, 所以习题描述词数、输入格式词数、输出格式词数和是否为英文题干这几个要素会影响学生的失误行为, 与失误率相关. 值得注意的是, 某些知识点所涉及的 OJ 习题存在简单暴力的解决方法, 如“动态规划”的相关习题, 如果测试用例设计得不好, 学生虽然不会动态规划, 使用递归算法也能通过, 因此习题考查的知识点与猜对率有关.

MC-DRER 结合公式(4)~(10)来建模由学生的遗忘、学习、失误和猜对行为所引起的知识状态改变, 具体如下:

在  $t-1$  时刻将学生的知识状态向量  $h_{t-1}$  输入到 GRU 网络之前, 先对向量  $h_{t-1}$  进行知识状态提纯, 方法如公式(11)所示:

$$h'_{t-1} = E_t^T \text{softmax}(W_e) \cdot h_{t-1} \quad (11)$$

其中,

$$E_t = [FK_t, FL_t, LK_t, LL_t, 1 + ER_t, 1 - GR_t],$$

$W_e$  是平衡遗忘、学习、失误和猜对行为的可训练权重向量, 形状为  $6 \times 1$ . 之后, 将经过提纯后的知识状态向量  $h'_{t-1}$  和考虑了答题结果的习题向量  $px_t$  输入到 GRU 网络中, 得到  $t$  时刻的学生知识状态向量  $h_t$ , 如公式(12)所示:

$$h_t = \text{GRU}(h'_{t-1}, px_t; \Phi_{gru}) \quad (12)$$

### 4.3 知识图谱

当前的习题推荐工作往往用随机生成的向量来表征不同的知识点, 但随机生成的向量无法很好地表达知识点之间的相关性. 当新的在线学习平台上线, 在可用来训练模型的内部数据比较稀少的情况下, 这些方法很难通过内部数据挖掘知识点相关性信息. 因此, 需考虑通过外部信息将知识点之间的相关性信息引入到模型中. 在认知心理学推理、决策与判断的相关工作<sup>[37]</sup>中经常强调, 人相对于机器而言, 最大的优势在于能够使用外部信息, 而知识图谱是机器学习模型引入外部信息的常用媒介.

在习题推荐<sup>[9,50]</sup>和知识追踪<sup>[51,52]</sup>领域, 有学者认识到知识图谱的重要性, 并提出了相关模型. Zhu 等人<sup>[50]</sup>认为学生的学习场景共有 4 种, 分别为初始学习、常规复习、考前学习和考前复习, 根据这 4 种场景设计了 7 种通往指定目标知识点的学习路径约束, 按需求提供适合的学习路径. 但从本文所使用的 OJ 源数据中难以挖掘出用户的学习场景, 并且大多数 OJ 用户往往都是以掌握所有知识点作为其目标, 没有某个明确的目标知识点. Gong 等人<sup>[9]</sup>提出了 ACKRec 模型, 使用学生与知识、学生与课程、学生与视频、课程与教师这四种关系构建异构信息网络, 并使用图卷积网络对每种实体进行表示学习从而实现知识点推荐. Yang<sup>[52]</sup>等人构建了习题-知识关系图, 并使用图卷积神经网络与循环神经网络来跟踪学生知识状态的动态变化. 但文献[9]和文献[52]所提出的工作没有考虑知识点之间的关系. Chen 等人<sup>[51]</sup>在所提出的知识追踪模型中量化知识先后学习顺序为约束, 再转换约束为目标函数的一部分. 然而, 程序设计领域的知识关系种类较多, 只考虑先后顺序则不够充分. 为了考虑知识点间复杂多样的关系, 本文设计了程序设计领域知识图谱专门应用于 OJ 习题推荐任务, 并使用知识图谱嵌入技术将习题转为向量输入到习题推荐模型中.

本文所设计的程序设计领域知识图谱中包含了

“数据结构”（如栈、队列和链表）、“算法”（如广度优先搜索、遗传和快速排序）和“问题”（如 NP 问题、最短路和割点）三类知识点实体。同时，知识图谱还包含了五种语义关系：“include”表示同类知识点实体之间的包含关系；“next”指明下一个该学的知识点，表示同类知识点实体之间的显式后序关系；“nickname”表示同义知识点；“solve”表示算法类实体与问题类实体之间的解决关系，即某算法能够解决某个问题；“implement”表示数据结构类实体与算法类实体之间的实现关系，即某数据结构能够实现某算法。目前在程序设计领域知识图谱中，有 93 个数据结构实体，158 个算法实体，169 个问题实体，总计 420 个知识点实体。整个程序设计领域知识图谱由 864 个三元组组成，三元组形态如（二叉树，include，红黑树）、（冒泡排序，next，

快速排序）、（最大流，nickname，最小割）、（广度优先搜索，solve，最短路）、（优先队列，implement，Dijkstra）等，其中有 629 个包含“include”关系的三元组，有 51 个包含“next”关系的三元组，有 71 个包含“solve”关系的三元组，有 108 个包含“implement”关系的三元组，有 5 个包含“nickname”关系的三元组。

程序设计领域知识图谱可以被看作是一个三层的立体图谱，自下向上分别为数据结构层、算法层和问题层，三层图谱分别包含了数据结构、算法和问题类知识点实体。同层内的实体通过“include”、“next”和“nickname”关系建立连接，数据结构层和算法层的实体通过“implement”关系建立连接，算法层和问题层的实体通过“solve”关系建立连接。知识图谱中部分知识点之间的关系如图 5 所示。

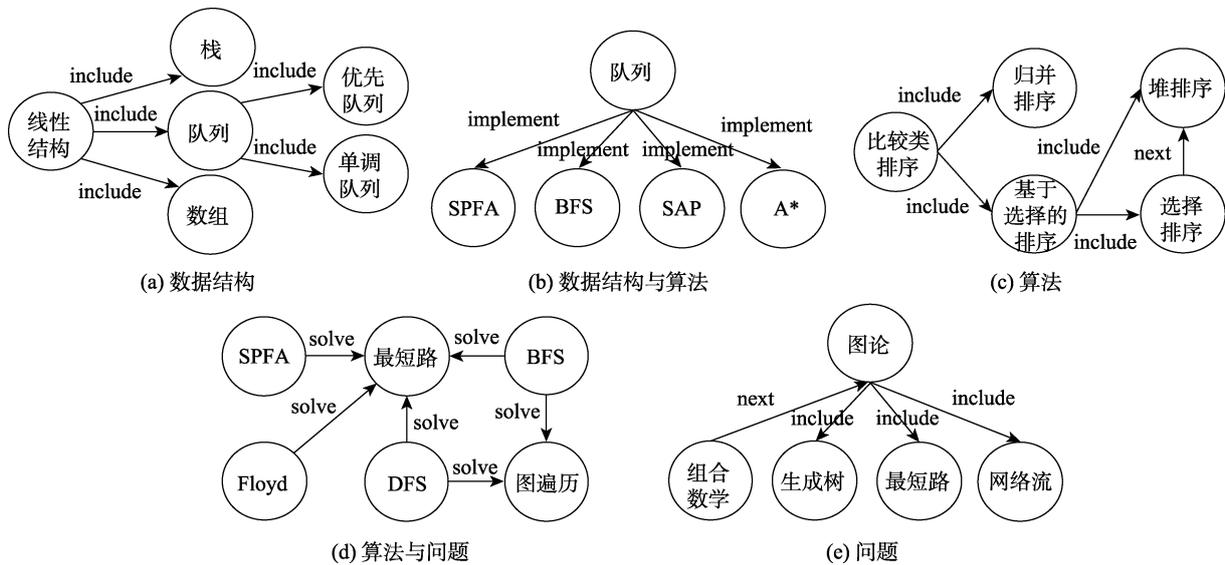


图 5 部分知识点之间的关系

在将知识图谱应用于某个 OJ 平台的习题推荐任务时，该平台的习题也会被加入到知识图谱中作为实体，通过 genre 关系与其所考查的知识点建立关联，如（习题 1，genre，快速排序）。在习题也被加入到知识图谱中之后，MC-DRER 模型以 TransE<sup>[53]</sup>算法生成的习题向量取代在 DRER 中随机生成的知识点向量，输入到 OJ 习题推荐模型中。

#### 4.4 非连续性认知发展

现有习题推荐工作或者假定学生的认知发展是固定不变的（如基于认知诊断的推荐方法），或者认定学生的认知发展是一个连续的、循序渐进的过程（如基于知识追踪的推荐方法）。但在认知心理学中，有大量的科学研究<sup>[38-40]</sup>表明认知发展不仅有连

续性的，还有非连续性的。认知发展的非连续性指的是学生的知识状态在某个时间点突然发生了较大变化，即学生突然掌握了某些较难的知识点（可能是在当前学习平台之外进行了学习）或者学习兴趣发生了较大改变（可能是受到了同伴或教师的影响）。但目前习题推荐领域中的工作忽略了认知发展的非连续性。因此，本文设计了两种策略用以指导模型处理非连续性认知发展的情况。

##### 4.4.1 基于高效用项集的非连续认知挖掘算法

现有基于知识点学习顺序的认知诊断<sup>[54]</sup>、知识追踪<sup>[51]</sup>和习题推荐<sup>[55,56]</sup>工作中，知识点学习顺序的获取方法有两种：专家标注知识点学习顺序<sup>[51,54]</sup>；通过专家所设计的教学目录中的课程顺序和课程所

涉及的知识点来计算知识点学习顺序<sup>[55,56]</sup>。但是采用专家直接或间接标注的学习顺序对于预测学生突然掌握或关注与之前作答习题毫无显式关联的知识点并没有显著效果，因此本文尝试通过学生的学习行为，使用数据挖掘算法来挖掘知识点学习顺序，并将挖掘到的顺序加入原有的程序设计领域知识图谱，从而使习题推荐模型能够考虑学生的非连续性认知发展现象，提升模型的准确性。目前已有工作提出了基于频繁项集或高效用项集挖掘算法的知识点隐式关联分析方法<sup>[57,58]</sup>，但是若将通过这些数据挖掘算法所挖掘到的知识点项集应用于习题推荐任务则存在局限性，比如将知识点项集{冒泡排序，快速排序}应用于习题推荐任务，则会在学生学习完知识点“快速排序”后给学生推荐知识点“冒泡排序”相关的习题，该推荐结果明显不合理，因为“快速排序”是“冒泡排序”的后序知识点，学会了“快速排序”的学生必然已经掌握了“冒泡排序”，因此不用再推荐“冒泡排序”相关的习题。为了解决这个问题，本文对高效用项集挖掘算法进行了修改，使其能够挖掘知识点之间的隐式后序关系，从而获得所有知识点的隐式学习顺序。具体步骤如下所示：

(1) 遍历所有学生，获取每一位学生的习题交互序列  $e_{1T} = \{e_1, e_2, \dots, e_T\}$ ，遍历该学生的习题交互序列，找到每一道习题所考查的知识点，将习题交互序列中的每一道习题替换为该习题所考查的知识点，得到该学生的知识点序列  $i^k = \{k_1, k_2, \dots, k_T\}$ ；

(2) 如果  $k_m$  和  $k_n$  在所有学生的知识点序列中的效用值（同时出现的次数）大于等于最小效用阈值  $E$ ，那么就作为高效用 2-项集保留下来；

(3) 如果在所有高效用 2-项集中，元素  $k_m$  出现在  $k_n$  前面的次数大于等于  $k_n$  出现在  $k_m$  前面的次数，则在知识图谱中的知识点  $k_m$  和  $k_n$  之间建立 next-algo 关系（代表由算法挖掘出的隐式后序关系），构成三元组  $(k_m, \text{next-algo}, k_n)$ 。如果  $k_m$  出现在  $k_n$  前面的次数小于  $k_n$  出现在  $k_m$  前面的次数，则建立三元组  $(k_n, \text{next-algo}, k_m)$ ；

(4) 将生成的三元组与知识图谱中原有的三元组一同输入 TransE 算法，得到涵盖了  $k_m$  和  $k_n$  之间隐式后序关系的表征向量，辅助模型处理学生认知发展突变的情况。

下面以伪代码的形式给出基于高效用项集的非连续认知挖掘算法，如算法 1 所示：

**算法 1.** 基于高效用项集的非连续认知挖掘算法。  
输入：学生集合  $U$ ，知识点集合  $K$ ，知识图谱  $g$ ，最

小效用阈值  $E$

输出：知识点向量集合  $V_k$

1.  $occur\_times = dict()$
2. FOR  $u$  IN  $U$  DO
3. 获取学生  $u$  的习题交互序列  $e_{1T} = \{e_1, e_2, \dots, e_T\}$
4.  $i^k = list()$
5. FOR  $t = 1$  TO  $T$  DO
6. 通过知识图谱  $g$  获取习题  $e_t$  所考查的知识点  $k_t$
7. 将  $k_t$  插入列表  $i^k$  的末尾
8. END FOR
9. FOR  $m = 1$  TO  $T$  DO
10. FOR  $n = m+1$  TO  $T$  DO
11. IF  $occur\_times$  包含键  $(i_m^k, i_n^k)$  DO
12.  $occur\_times[(i_m^k, i_n^k)] += 1$
13. ELSE
14.  $occur\_times[(i_m^k, i_n^k)] = 1$
15. END IF
16. END FOR
17. END FOR
18. END FOR
19. FOR  $k_m, k_n (m \neq n)$  IN  $K$  DO
20. IF  $occur\_times[(k_m, k_n)] + occur\_times[(k_n, k_m)] \geq$

$E$  DO

21. IF  $occur\_times[(k_m, k_n)] \geq occur\_times[(k_n, k_m)]$

DO

22. 将三元组  $(k_m, \text{next-algo}, k_n)$  加入知识图谱  $g$
23. ELSE
24. 将三元组  $(k_n, \text{next-algo}, k_m)$  加入知识图谱  $g$
25. END IF
26. END IF
27. END FOR
28.  $V_k = TransE(g)$

#### 4.4.2 认知突变奖励函数

基于序列推荐算法的相关工作所提出的模型往往只使用学生过去的学习行为来推测学生下一个时间点会交互的学习资源，无法很好地考虑学生在下一个时间点认知发展上可能产生的突然变化，由于这些工作在推荐时只参考了学生过去的学习资源交互记录，所以只会推荐与学生过去学习过的资源相关的资源。为解决这个问题，并受机器翻译评估指标 BLEU 的启发，本文重构了基于强化学习的 DRER 网络的奖励函数，在 DRER 奖励函数的基础上添加了认知突变奖励函数来训练模型，重构之后的奖励函数如公式 (13) 所示：

$$R(s_t, a_t) = R_{DRER} + R_{seq} \quad (13)$$

其中,  $R_{seq}$  为认知突变奖励函数, 认知突变奖励函数以学生在  $t+1$  到  $t+b$  时刻的习题交互序列 (时刻跨度  $b$  为正整数, 并且  $t+b \leq T$ ):

$$e_{t+1:t+b} = \{e_{t+1}, e_{t+2}, \dots, e_{t+b}\} \quad (14)$$

和模型利用学生在  $t$  时刻及其之前的历史答题序列:

$$i_{1:t} = \{(e_1, p_1), (e_2, p_2), \dots, (e_t, p_t)\} \quad (15)$$

所预测的学生在  $t+1$  到  $t+b$  时刻的习题交互序列  $\hat{e}_{t+1:t+b}$  作为输入, 如公式 16 所示:

$$R_{seq}(e_{t+1:t+b}, \hat{e}_{t+1:t+b}) = \exp\left(\frac{1}{b} \sum_{m=1}^b \ln prec_m\right) \quad (16)$$

$prec_m$  为修正的精度, 定义如下:

$$prec_m = \frac{\sum_{p_m \in e_{t+1:t+b}} \min(\#(p_m, e_{t+1:t+b}), \#(p_m, \hat{e}_{t+1:t+b}))}{\sum_{p_m \in e_{t+1:t+b}} \#(p_m, e_{t+1:t+b})} \quad (17)$$

其中,  $p_m$  是  $e_{t+1:t+b}$  中的一个  $m$ -gram 子序列,  $\#(p_m, e_{t+1:t+b})$  是  $p_m$  出现在  $e_{t+1:t+b}$  中的次数. 如公式 16 所示,  $R_{seq}$  奖励函数能够支持推荐模型生成与未来真实习题交互序列更一致的习题交互序列, 从而使推荐模型在给推荐下一道习题  $\hat{e}_{t+1}$  时可以考虑学生的认知在未来发生的突然变化.  $R_{DRER}$  为 DRER 模型的奖励函数变体, 以学生在  $t$  时刻及其之前的历史答题序列  $i_{1:t}$  和模型所预测的  $t+1$  到  $t+b$  时刻的习题交互序列  $\hat{e}_{t+1:t+b}$  作为输入, 如公式(18)所示:

$$R_{DRER}(i_{1:t}, \hat{e}_{t+1:t+b}) = \frac{1}{4} \left\{ \frac{1}{b} [R_2(e_t, \hat{e}_{t+1}) + \sum_{m=2}^b R_2(\hat{e}_{t+m-t+m})] + R_1(e_{1:t}, \hat{e}_{t+1}) + R_3(i_{1-N:t}) \right\} + \frac{1}{2} \quad (18)$$

其中,  $R_1$ 、 $R_2$  和  $R_3$  分别为 DRER 模型中的“复习与探索”、“难度平滑性”和“参与度”奖励子函数 (子函数细节可参考文献[1]中的公式(7)~(9)).

#### 4.5 模型训练

本文的目标是要找到一个向学生推荐习题的最优策略  $\pi$ , 使得学生在学习所推荐习题的过程中累积的推荐奖励值期望  $J(\theta)$  最大化,  $J(\theta)$  的定义如公式 (19) 所示:

$$J(\theta) = \mathbb{E}_{\pi} \left\{ \sum_{u=1}^{T-b} \sum_{t=1}^{T-b} [\ln(\pi(a_t | s_t; \theta))] \sum_{j=t}^{T-b} \gamma^{j-t} R(s_j, a_j) \right\} \quad (19)$$

其中,  $\theta$  是模型的相关参数,  $\gamma \in (0,1)$  为惩罚因子,  $a_t = \hat{e}_{t+1}$ ,  $\pi(a_t | s_t; \theta)$  代表的是在模型拟合的最优策略  $\pi$  下, 当学生状态  $s_t$  为时, 最有可能给学生推荐的习题  $\hat{e}_{t+1}$  的概率, 计算方法如公式(20)所示:

$$\pi(a_t | s_t; \theta) = \pi(\hat{e}_{t+1} | s_t; \theta) = \frac{\exp(x_{t+1}^T M h_t)}{\sum_{e \in E} \exp(x^T M h_t)} \quad (20)$$

其中,  $M \in \mathbb{R}^{d_x \times d_h}$  为可训练的参数矩阵,  $d_x$  为习题

向量的维度,  $d_h$  为学习状态向量的维度. 模型参数学习的步骤如算法 2 所示.

#### 算法 2. 模型参数学习方法.

输入: 策略  $\pi$ , 奖励函数  $R$ , 知识图谱  $g$ , 学习率  $lr$

输出: 训练好的模型参数  $\theta$

29. 随机初始化模型参数  $\theta$

30. 根据 TransE 算法获取习题的知识点向量

31. FOR  $u$  IN  $U$  DO

32. 根据公式 1、2 获取所有习题的向量

33. FOR  $t=1$  TO  $T-b$  DO

34.  $cumulative\_reward = 0$

35. FOR  $j=t$  TO  $T-b$  DO

36.  $cumulative\_reward += \gamma^{j-t} R(s_j, a_j)$

37. END FOR

38.  $\theta \leftarrow \theta + lr \frac{\nabla \pi(a_t | s_t; \theta)}{\pi(a_t | s_t; \theta)} cumulative\_reward$

39. END FOR

40. END FOR

## 5 实 验

本文在洛谷和 CodeForces 两个业界知名的 OJ 平台上进行实验, 通过对比 MC-DRER 与其他传统推荐算法、基于认知诊断的推荐方法、基于知识追踪的推荐方法和基于强化学习的推荐方法的性能, 验证了 MC-DRER 的有效性.

### 5.1 数据介绍与分析

#### 5.1.1 数据介绍

拥有高质量的数据集是建立有效的 OJ 习题推荐模型的基础, 本文收集了洛谷和 CodeForces 的数据作为项目实验的源数据. 两个数据集都包括了题目信息和用户提交记录, 其中题目信息中的每条数据代表一个题目的信息, 包含字段: 题目 id、标题、题干 (包括习题描述、输入格式和输出格式)、题目知识点标签; 用户提交记录中的每条数据包含字段: 提交 id、用户 id、题目 id、提交状态、提交时间戳. 洛谷和 CodeForces 数据集的基本情况见表 1, 分别从每个数据集中随机抽取 80% 的学生训练 MC-DRER, 抽取 10% 的学生验证 MC-DRER, 抽取 10% 的学生测试 MC-DRER.

表 1 数据集基本情况

数据集名称	用户数量	习题数量	知识点数量	提交记录数
洛谷	12804	5786	208	384800
CodeForces	26787	7008	36	1048575

### 5.1.2 数据分析

本节对洛谷和 CodeForces 数据集进行深入分析. 首先, 分析“习题描述词数”、“输入词数”、“输出词数”和“是否为英文题干”这四个习题特征与习题难度的关联性. 本文选取作答人数在 10 以上的习题 (习题的作答人数越多, 习题错误率就越能准确地反映习题的难度), 通过这些习题的特征与学生作答错误率的折线图来观察特征与难度的关系, 如图 6 所示, 可以看出, “习题描述词数”、“输入词数”与“输出词数”较大的习题, 其难度往往也会较大.

此外, 在洛谷数据集中, 英文习题的平均学生作答错误率为 53.88%, 中文习题的平均学生作答错误率为 28.65%. 可见习题的题干如果是用非母语书写的, 那么就会加大习题的难度 (CodeForces 数据集中的习题皆为英文习题, 故无法使用该数据集分析习题特征“是否为英文题干”和习题难度的关联性). 上述现象验证了 4.1 节所选取的用以描述 OJ 习题难度的四个特征的合理性. 当习题作答人数较少时, 这四个特征可以辅助习题推荐模型估计习题的难度.

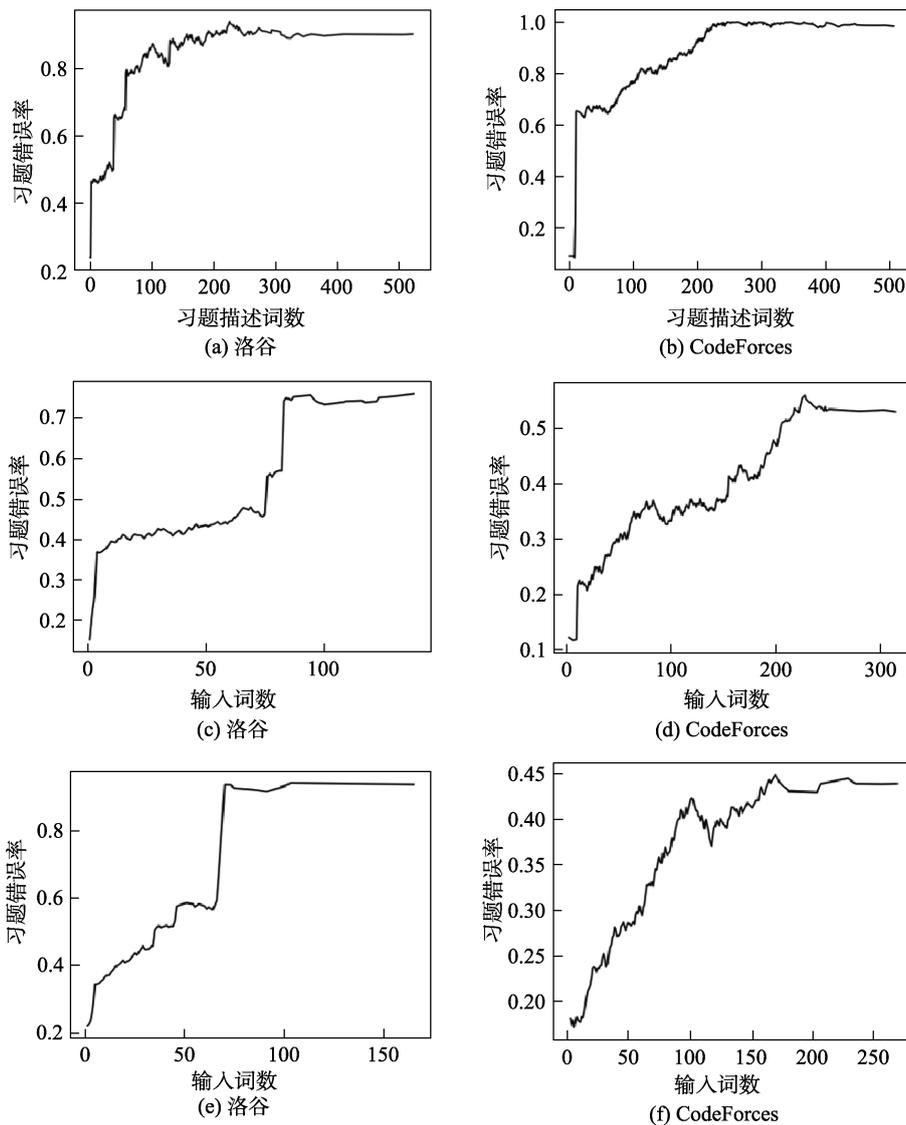


图 6 习题特征与作答错误率的关系图

然后, 通过洛谷和 CodeForces 数据集分析学生在学习程序设计过程中是否存在非连续性认知发展现象. 为了判断学生的认知发展在某个时间点是否突然发生了较大变化, 本文定义了两个概

念, 分别为“难度跳跃”和“知识点跳跃”. “难度跳跃”指的是学生在当前时刻所做习题的学生作答错误率与前一时刻所做习题的学生作答错误率之差大于 0.2 (本文假设, 如果连续两个习题的学

生作答错误率之差大于 0.2, 则难度变化较剧烈)。“知识点跳跃”指的是连续两个习题所涉及的知识点在知识图谱中没有关系, 即学生的学习兴趣发生了较大变化. 将各个学生的作答习题数与其在答题过程中发生“难度跳跃”、“知识点跳跃”的次数绘制成折线图, 如图 7 所示. 为了排除题库本身顺序给“难度跳跃”、“知识点跳跃”的次数造成的影响, 若学生的习题交互序列中的某道习题的 id 值与 1 的差等于序列中前一道习题的 id 值, 则该习

题不纳入本实验的统计范围内. 在洛谷数据集中, 学生大约每做 7 道习题就会发生一次“难度跳跃”, 每做 2 道习题就会发生一次“知识点跳跃”. 在 CodeForces 数据集中, 学生大约每做 6 道习题就会发生一次“难度跳跃”, 每做 4 道习题就会发生一次“知识点跳跃”. 通过上述分析可知, 非连续性认知发展在学生学习程序设计的过程中是一种普遍现象, 因此, 在设计习题推荐模型时, 有必要考虑学生的非连续性认知发展.

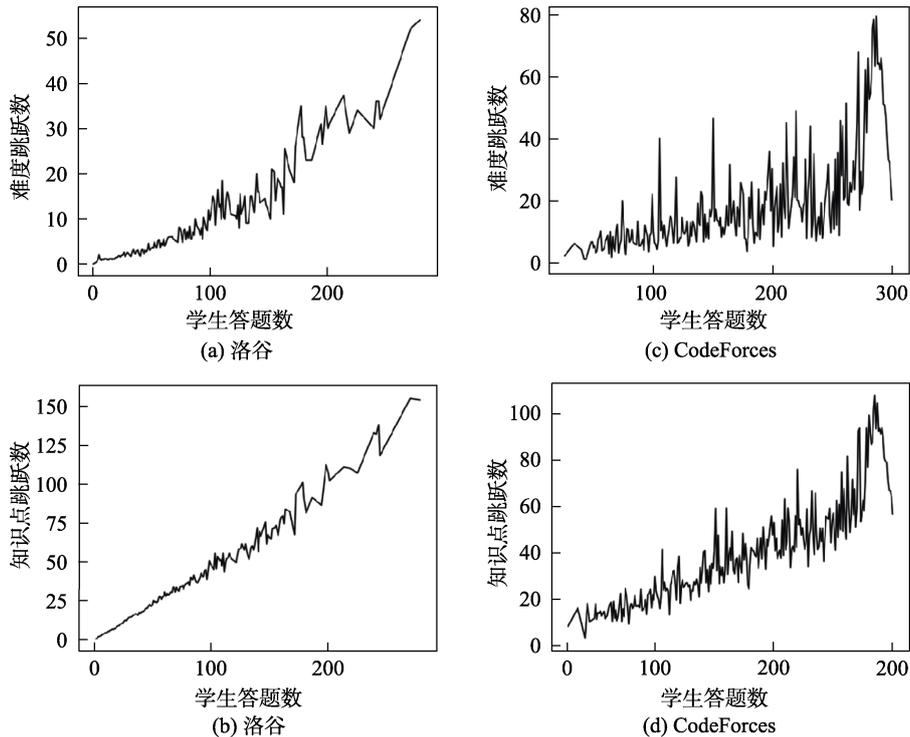


图 7 非连续性认知发展分析图

最后, 通过洛谷和 CodeForces 数据集分析学生在学习程序设计的过程中是否存在编程语言遗忘与学习的现象, 并验证编程语言熟悉程度与学习时间间隔、总体学习次数的关联性. OJ 数据集的用户提交记录中的提交状态可以分成三类, 分别为“Accepted”、“Wrong Answer”和“Compile Error”. “Accepted”表示学生作答正确. “Wrong Answer”表示学生作答错误, 其所提交程序的实际输出与测试用例的期望输出不一致, 出现这种情况的主要原因是学生未能够完全掌握习题所考查的知识点. “Compile Error”表示学生作答错误, 其所提交程序未能编译成功, 说明学生的编程语言熟悉程度不够高. 本文通过状态为“Compile Error”的提交记录来分析学生编程语言遗忘与学习的现象. 本文统计了状态为“Compile Error”的提交记录占有提交

记录的比例, 将该比例称为“总体编译错误比例”. 同时, 获取所有学生初学的前 10 条提交记录和暂停学习大于等于 7 天后继续开始学习的前 10 条提交记录, 分别统计状态为“Compile Error”的提交记录占这两种情况下的提交记录的比例, 将这两种比例称为“初学编译错误比例”和“间断学习编译错误比例”, 结果如表 2 所示.

表 2 提交记录分析表 (%)

数据集	总体编译错误比例	初学编译错误比例	间断学习编译错误比例
洛谷	10.31	23.59	18.37
CodeForces	9.72	20.68	16.09

通过表 2 可知, 在两个数据集中, “间断学习编译错误比例”远高于“总体编译错误比例”, 则说明

在学习时间间隔较长的情况下，学生会对编程语言产生遗忘，熟悉程度会下降，从而更有可能提交未能编译成功的程序。在日常学习的过程中，每当较长时间没有编写程序，学生有较大概率会遗忘编程语言内置的部分数据结构的操作方法，比如列表排序、出栈、查找字符串中的某个子串等，编程语言熟悉程度大幅度下降。因此，实验结果也符合日常学习的过程。此外，初学的前 10 条记录的编译错误率远高于其余的记录，也证明了学生可以通过增加总体学习次数来提升编程语言熟悉程度。通过上述分析可知，编程语言的遗忘与学习现象在学生学习程序设计的过程中普遍存在，会影响学生的答题错误率，并且与学习时间间隔和总体学习次数有关。因此，在设计 OJ 习题推荐模型时，需要考虑编程语言熟悉程度，并将学习时间间隔和总体学习次数加入建模的过程。

## 5.2 评价指标

对于下一交互习题预测任务，本文使用  $Recall@10$  (Recall Rate of Top-10 items) 和  $MRR@10$  (Mean Reciprocal Rank of Top-10 items) 来比较 OJ 习题推荐模型的性能。 $Recall@10$  和  $MRR@10$  的详细定义如下：

$$Recall@10 = \frac{1}{|U|} \sum_{u \in U} \frac{|R_u \cap T_u|}{|R_u|} \quad (21)$$

$$MRR@10 = \frac{1}{|U|} \sum_{u \in U} \frac{1}{rank_u} \quad (22)$$

其中， $|U|$  是学生的数量， $R_u$  为学生  $u$  的习题测试集， $T_u$  为推荐给学生的前 10 个习题所构成的集合， $rank_u$  为学生  $u$  的推荐列表的前 10 个习题中第 1 个正确推荐的习题所处的位置。

对于每个真实输出随机采样了 100 个负例，真实输出和负例一同排序。

## 5.3 实验参数设置

本文使用 Tensorflow 实现所提出方法，优化器选择 Adam，方法对应的超参数设置如表 3 所示。

表 3 超参数设置

参数名	CodeForces	洛谷
批处理大小 $bs$	128	128
学习率 $lr$	0.001	0.001
知识点向量维度 $d_k$	10	10
学习状态向量维度 $d_h$	128	128
最小效用阈值 $E$	14	8
时刻跨度 $b$	4	3
惩罚因子 $\gamma$	0.9	0.9

## 5.4 实验结果分析

### 5.4.1 习题推荐实验结果

为了评估 MC-DRER 模型的性能，本文选择传统推荐算法 PMF<sup>[59]</sup>、FM<sup>[60]</sup>、GRU4REC<sup>[61]</sup>、RippleNet<sup>[62]</sup>、KGCN-LS<sup>[63]</sup>、KSR<sup>[64]</sup>、KERL<sup>[65]</sup>，基于认知诊断的方法 IRT<sup>[18]</sup>、DINA<sup>[19]</sup>、PMF-CD<sup>[11]</sup>，基于知识追踪的方法 DKT<sup>[26]</sup>、DKVMN<sup>[42]</sup>、DKT-CF<sup>[13]</sup> 和基于强化学习的方法 DRER<sup>[1]</sup> 作为基准方法，实验结果如表 4 所示。

表 4 模型在 CodeForces 和洛谷数据集上的效果

模型	Recall@10		MRR@10	
	CodeForces	洛谷	CodeForces	洛谷
IRT	0.1982	0.1894	0.1243	0.1173
DINA	0.2012	0.1926	0.1423	0.1354
PMF	0.2031	0.1937	0.1598	0.1502
FM	0.2008	0.1906	0.1587	0.1496
PMF-CD	0.2057	0.1962	0.1624	0.1527
DKT	0.2173	0.2064	0.1721	0.1613
DKVMN	0.2208	0.2095	0.1787	0.1674
DKT-CF	0.2227	0.2114	0.1809	0.1710
GRU4REC	0.2156	0.2089	0.1708	0.1672
RippleNet	0.2547	0.2483	0.2103	0.2016
KGCN-LS	0.2793	0.2691	0.2321	0.2217
KSR	0.3254	0.3175	0.2716	0.2629
KERL	0.3323	0.3216	0.2879	0.2782
DRER	0.2804	0.2712	0.2309	0.2196
DRER-e	0.3142	0.3128	0.2601	0.2568
DRER-d	0.3259	0.3170	0.2729	0.2593
DRER-k	0.3165	0.2879	0.2648	0.2311
DRER-h	0.3032	0.2976	0.2527	0.2447
DRER-r	0.3191	0.3095	0.2649	0.2538
MC-DRER	<b>0.4160</b>	<b>0.4013</b>	<b>0.3595</b>	<b>0.3445</b>

从表 4 中可以发现，由于 MC-DRER 充分关注了认知心理学中会影响学生答题行为的因素，因此，性能要优于基准方法。同时，通过消融实验可以看出，针对 DRER 的每一点改进，都是有效的，其中，DRER-e、DRER-d、DRER-k、DRER-h 和 DRER-r 分别为使用了本文提出的习题编码方式、加入了知识状态提纯模块、引入了知识图谱、引入了修改后的高效用项集挖掘算法和引入了认知突变奖励函数的一些细节，对 DRER 分别做出改进后在洛谷数据集上进行测试，模型在  $Recall@10$  上分别提升了 0.0416、0.0458、0.0167、0.0264 和 0.0383，在

$MRR@10$  上分别提升了 0.0372、0.0397、0.0115、0.0251 和 0.0342. 不难发现, 知识图谱的加入没有对模型在性能上的提升有太大的帮助, 经过分析, 出现这种情况的原因是: 在 CodeForces 数据集的 7008 道习题中, 有 6684 道习题有知识点标签, 有知识点标签的习题占总习题的 95.38%, 而在洛谷数据集的 5786 道习题中, 只有 3827 道习题拥有知识点标签, 有知识点标签的习题占总习题的 66.14%, 没有知识点标签的习题无法通过知识图谱与知识点产生联系, 因而给模型挖掘习题与知识点、习题与习题之间的关联造成了困难, 致使模型性能得不到显著提升. 此外, 还对 DRER 分别做出改进后在 CodeForces 数据集上进行测试, 模型在  $Recall@10$  上分别提升了 0.0338、0.0455、0.0361、0.0228 和 0.0387, 在  $MRR@10$  上分别提升了 0.0292、0.042、0.0339、0.0218 和 0.034. 在 CodeForces 数据集上, 习题特征的改进效果并没有洛谷数据集显著, 造成这种结果的原因是 CodeForces 数据集中的所有习题都是由英文描述的, 因此, “是否为英文题干” 这个特征也就完全没有了意义, 从而干扰模型挖掘习题的难度, 降低了模型的性能.

#### 5.4.2 遗忘与学习机制对比实验结果

对于遗忘和学习行为, 在知识追踪领域, 已经有学者<sup>[42-47]</sup>认识到这两种行为对知识水平评估的重要性, 在设计模型时考虑了这两种行为, 本文以文献[42]和文献[46]所提出的遗忘与学习机制为例进行介绍和实验对比. Zhang 等人<sup>[42]</sup>为每一道习题在被正确解答和错误解答的两种情况下各分配了一个的遗忘和学习向量进而对学生的知识水平进行评估, 如公式(23)所示:

$$\mathbf{h}'_{t-1} = \mathbf{h}_{t-1} \odot (\mathbf{I} - \mathbf{f}_t) + \mathbf{z}_t \quad (23)$$

其中,  $\mathbf{f}_t$  为习题  $e_t$  的遗忘向量, 形状为  $d_h \times 1$ ,  $d_h$  为知识状态向量  $\mathbf{h}$  的维度;  $\mathbf{z}_t$  为习题  $e_t$  的学习向量, 形状为  $d_h \times 1$ ;  $\mathbf{I}$  是值全为 1 的向量, 形状为  $d_h \times 1$ ;  $\odot$  是哈达玛积.

Huang 等人<sup>[46]</sup>选用相同知识的学习次数和双曲线形式定义记忆因子, 相同知识的学习时间间隔和指数曲线形式定义遗忘因子, 从而建模学生的遗忘与学习行为, 如公式(24)所示:

$$\mathbf{h}'_{t-1} = [w_\alpha L_t + (1 - w_\alpha) F_t] \mathbf{h}_{t-1} \quad (24)$$

其中,

$$L_t = \frac{(D+1)KT_t}{KT_t + r}, \quad F_t = e^{-\frac{RT_t}{S}}$$

$D$  和  $r$  是可训练参数, 分别控制了知识水平增长的

幅度和系数;  $S$  是控制记忆程度的可训练参数;  $w_\alpha$  ( $0 \leq w_\alpha \leq 1$ ) 是平衡记忆与遗忘因子的可训练参数.

本文在 4.2 节中提出的遗忘与学习机制, 如公式(25)所示:

$$\mathbf{h}'_{t-1} = \mathbf{Z}_t^T \text{softmax}(\mathbf{W}_z) \cdot \mathbf{h}_{t-1} \quad (25)$$

其中,

$$\mathbf{Z}_t = [FK_t, FL_t, LK_t, LL_t],$$

$\mathbf{W}_z$  是平衡记忆与遗忘因子的可训练权重向量, 形状为  $4 \times 1$ .

本文分别将 4.2 节所提出的遗忘与学习机制(如公式(25)所示)和文献[42]、文献[46]所提出的遗忘与学习机制(如公式(23)、(24)所示)引入 DRER 模型, 并在洛谷和 CodeForces 数据集上进行实验, 从而比较三种遗忘与学习机制应用在 OJ 习题推荐任务上的性能. 实验结果如表 5 所示, 其中, DRER-eq23、DRER-eq24 和 DRER-eq25 分别表示引入了公式(23)、公式(24)和公式(25)的 DRER.

表 5 模型在不同遗忘与学习机制下的表现

模型	Recall@10		MRR@10	
	CodeForces	洛谷	CodeForces	洛谷
DRER	0.2804	0.2712	0.2309	0.2196
DRER-eq23	0.2908	0.2820	0.2416	0.2297
DRER-eq24	0.3027	0.2937	0.2520	0.2329
DRER-eq25	0.3129	0.3018	0.2603	0.2458

根据表 5 的实验结果可知, 本文所提出的遗忘与学习机制能够提升模型在 OJ 习题推荐任务上的性能, 并且本文提出的遗忘与学习机制优于文献[42]、文献[46]所提出的机制, 原因在于本文所提出的遗忘与学习机制考虑了学生编程语言熟悉程度, 即学生学习时间间隔与总体学习次数. 当学生总体学习次数越大, 学生的编程语言熟练度也越高. 当学习时间间隔越大, 学生越会对编程语言感到生疏, 编程语言熟练度也越低.

#### 5.4.3 习题特征对比实验结果

为了验证习题描述词数、输入格式词数、输出格式词数和是否为英文题干的有效性, 本节将这 4 个习题特征分别引入 DRER 进行实验. 实验结果如表 6 所示, 其中, DRER-desc、DRER-input、DRER-output 和 DRER-english 分别为引入了习题描述词数、输入格式词数、输出格式词数和是否为英文题干的 DRER. DRER-all\_features 为引入 4 个特征后的 DRER. 从表 6 的实验结果中可以观察到在 DRER

分别引入了 4 个特征后,模型的性能皆得到了提升,证明了这 4 个特征中的任何一个特征对于习题推荐任务都是有效的,并且同时引入 4 个特征后,效果更佳.

表 6 模型引入不同特征后的表现

模型	Recall@10		MRR@10	
	CodeForces	洛谷	CodeForces	洛谷
DRER	0.2804	0.2712	0.2309	0.2196
DRER-desc	0.2889	0.2787	0.2375	0.2267
DRER-input	0.2875	0.2769	0.2356	0.2245
DRER-output	0.2874	0.2775	0.2354	0.2253
DRER-english	—	0.2773	—	0.2249
DRER-all_features	0.3019	0.3022	0.2497	0.2488

#### 5.4.4 失误与猜对机制对比实验结果

为了验证学生作答正确率与提交作答正确率之差、习题描述词数、输入格式词数、输出格式词数、是否为英文题干和习题考查的知识点向量的有效性,本节将这 6 个要素分别加入失误与猜对机制,并引入到 DRER 进行实验.实验结果如表 7 所示,其中,DRER-difference、DRER-desc、DRER-input、DRER-output、DRER-english、DRER-knowledge 分别为加入了学生作答正确率与提交作答正确率之差、习题描述词数、输入格式词数、输出格式词数、是否为英文题干和习题考查的知识点向量的 DRER,DRER-all\_elements 为加入 6 个要素后的 DRER.从表 7 的实验结果中可以观察得到在失误与猜对机制分别加入了 6 个要素后,模型的性能皆得到了提升,证明了这 6 个特征中的任何一个特征对于习题推荐任务都是有效的,并且同时加入 6 个要素后,效果更佳.

表 7 模型引入不同要素后的表现

模型	Recall@10		MRR@10	
	CodeForces	洛谷	CodeForces	洛谷
DRER	0.2804	0.2712	0.2309	0.2196
DRER-difference'	0.2891	0.2806	0.2384	0.2267
DRER-desc'	0.2867	0.2765	0.2352	0.2245
DRER-input'	0.2852	0.2756	0.2349	0.2234
DRER-output'	0.2850	0.2762	0.2344	0.2239
DRER-english'	--	0.2753	--	0.2231
DRER-knowledge'	0.2877	0.2784	0.2375	0.2258
DRER-all_elements	0.3021	0.2965	0.2494	0.2404

#### 5.4.5 知识图谱驱动下的推荐结果合理性验证

为了直观地验证模型在知识图谱驱动下习题推

荐结果的合理性,本文从洛谷数据的测试集中随机抽取了一位学生的历史学习记录,分别输入到 DRER-k (引入程序设计领域知识图谱后的 DRER) 和 DRER 模型中.根据历史学习记录人工采样了相应习题作为习题推荐候选集,通过各个习题的相对推荐排序判断知识图谱驱动下的推荐结果是否合理.该学生的历史学习记录包含 4 道“递归”相关的习题(正确率为 100%)、3 道“广度优先搜索”相关的习题(正确率为 66.67%)、3 道“分治”相关的习题(正确率为 100%)、2 道“二叉堆”相关的习题(正确率为 100%)和 1 道“A\*”相关的习题(正确率为 100%).从历史学习记录中可以观察到该学生已经基本掌握了知识点“递归”、“广度优先搜索”、“分治”、“二叉堆”和“A\*”,根据程序设计教师的教学经验,教师期望模型接下来给他推荐与知识点“动态规划”、“Dijkstra”、“cdq 分治”、“斜堆”和“IDA\*”相关的习题.本文分别选取了与知识点“递归”、“广度优先搜索”、“分治”、“二叉堆”、“A\*”、“动态规划”、“Dijkstra”、“cdq 分治”、“斜堆”和“IDA\*”相关的 1 道习题加入习题推荐候选集.除此之外,本文还随机选取了两个知识点,分别为“后缀数组”和“栈”,从这两个知识点中再分别随机抽取 1 道习题加入习题推荐候选集,习题推荐排序的结果如表 8 所示.

表 8 习题推荐排序

知识点	DRER	DRER-k
递归	1	1
广度优先搜索	2	5
分治	3	3
二叉堆	4	7
A*	5	8
动态规划	6	2
Dijkstra	7	6
cdq 分治	8	4
斜堆	9	9
IDA*	10	10
栈	11	11
后缀数组	12	12

从表 8 中可知,引入知识图谱后,知识点“动态规划”、“Dijkstra”和“cdq 分治”相关的习题排序得到了提升,习题推荐结果更加合理,更加符合程序设计教师的期望.在复习旧知识点的同时,循序渐进,引出旧知识点关联的新知识点(如“动态规划”、“Dijkstra”和“cdq 分治”),并且不会引出

与旧知识点毫无关联的新知识点（如“栈”和“后缀数组”）。因为在程序设计领域知识图谱中，有如下三元组：（递归，next，动态规划）、（广度优先搜索，include，Dijkstra）、（分治，include，cdq 分治）、（斜堆，include，二叉堆）、（A\*，include，IDA\*），当 TransE 模型训练时可以学习到这些知识点之间的关系，从而帮助学生走出知识点舒适区，为学生推荐真正需要的习题。

#### 5.4.6 在线实验分析

在线场景下，本文关注于下一交互习题预测任务，为学生逐步地推荐习题。本文在模拟环境下进行实验，使用学生模拟器和系统的交互行为验证 MC-DRER 模型，通过每一时刻的推荐奖励值来实现自适应的推荐。在线实验着重验证本文提出的认知突变奖励函数的有效性。

本文需要构造一个学生模拟器来创建模拟环境，该学生模拟器的功能是输入一个习题，能够精确地预测学生的答题结果，即对与错。本文选择 SAINT 模型<sup>[66]</sup>作为学生模拟器。接着，本文分别将 CodeForces 和洛谷数据集中的学生按 50%/50% 的比例划分成两个子集，使用第一个子集训练 SAINT 模型，第二个子集训练 MC-DRER 模型。在训练模拟器的过程中，选择每个学生最后 10% 的学习记录进行模型验证，结果显示 SAINT 模型在 CodeForces 和洛谷数据集上的准确率分别为 87.5% 和 83.9%，准确率较高。因此，SAINT 模型适用于本文的 OJ 习题推荐在线实验场景。

本文选择随机贪心策略和 MC-DRER 的简化变体作为基准模型与 MC-DRER 模型进行在线场景下的对比实验。模型的具体描述如下：

- RG：随机贪心（Random Greedy，RG）策略，在每一步均随机挑选习题进行推荐。
- MC-DRER-r1：该算法仅考虑了  $R_{DRER}$  奖励函数（公式（18））。
- MC-DRER-r2：该算法仅考虑了  $R_{seq}$  奖励函数（公式（16））。
- MC-DRER：MC-DRER 算法不仅考虑了  $R_{DRER}$  奖励函数（公式（18）），而且还考虑了本文提出的认知突变奖励函数  $R_{seq}$ （公式（16））。

本节实验让每一个算法分别产生 20 次序列推荐的结果，计算它们的累计奖励作为在线实验的评估指标。在线习题推荐的实验结果如图 8 所示。

从图 8 中可以看出，MC-DRER 在在线场景下的性能最佳，其次是 MC-DRER 的两种简化变体，

最后是 RG 算法。实验结果表明：本文提出的认知突变奖励函数有助于习题推荐，并且 MC-DRER 可以同时优化多种学习目标，能够帮助模型学习最优推荐策略。此外，MC-DRER-r2 的性能优于 MC-DRER-r1，说明认知突变奖励函数在在线习题推荐的过程中具有更加重要的作用。

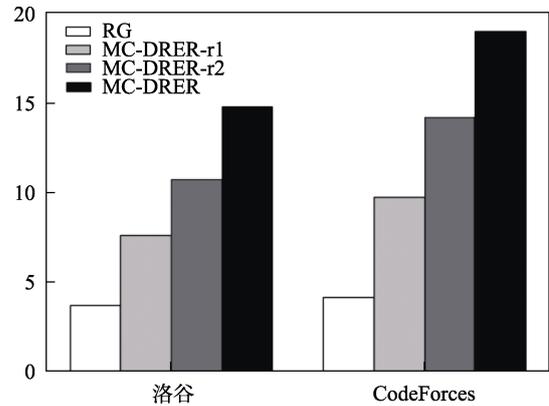


图 8 在线推荐实验结果

#### 5.4.7 重要超参数影响分析

本节着重讨论本文所提出的模型涉及到的两个重要超参数对习题推荐性能的影响，分别为高效用项集挖掘算法中的最小效用阈值  $E$  和认知突变奖励函数中的时刻跨度  $b$ 。

(1) 最小效用阈值。通过 5.4.1 节的实验结果可知，引入高效用项集挖掘算法能够提升模型在习题推荐任务中的性能，其中，最小效用阈值是高效用项集挖掘算法中的重要超参数。本节研究不同的最小效用阈值对模型性能的影响，该超参数的取值集合为 {2, 4, 6, 8, 10, 12, 14, 16, 18}。图 9 为不同的最小效用阈值下模型在 CodeForces 和洛谷数据集上 MRR@10 指标的变化曲线，在实验时其它超参数设置不变。从图 9 中可以看出，最小效用阈值对模型效果影响较大，CodeForces、洛谷数据集分别在 14 和 8 处取得最佳表现，阈值过小或者过大都会降低模型的性能。因为阈值过小时，知识图谱中会添加过多的 next-algo 关系，干扰模型挖掘知识点之间的隐式关系。当阈值过大时，next-algo 关系过少，使得经过高效用项集挖掘算法处理过的知识图谱与原先的知识图谱几乎没有差异，从而降低了模型的性能。此外，通过图 9 可以发现，CodeForces 数据集的最优最小效用阈值要大于洛谷数据集，原因可能为：CodeForces 的提交记录数大于洛谷，并且知识点数量小于洛谷。提交记录数越大或者知识点数量越小，某两个知识点的效用值也就可能越高，

同时也应该调高最小效用阈值来挖掘知识点之间普遍存在的隐式关系。

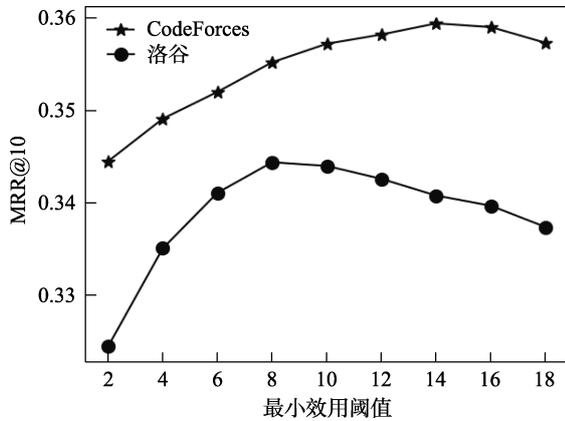


图 9 最小效用阈值对模型效果的影响

(2) 时刻跨度. 通过 5.4.1 节和 5.4.6 节的实验结果可知, 引入认知突变奖励函数能够提升模型的习题推荐效果, 其中, 时刻跨度是认知突变奖励函数的重要超参数. 本节研究不同的时刻跨度对模型性能的影响, 该超参数的取值集合为 {2, 3, 4, 5, 6, 7, 8, 9}. 图 10 为不同的时刻跨度下模型在洛谷和 CodeForces 数据集上 MRR@10 指标的变化曲线. 从图 10 中可以看出, 当时刻跨度分别为 4 和 3 时, 模型在 CodeForces 和洛谷数据集上的性能最优. 因为时刻跨度过小时, MC-DRER 会与只参考了学生过去的学习资源交互记录的习题推荐模型几乎没有差异而降低了推荐效果, 而当时刻跨度过大时, 模型的重点就放到了预测学生未来的长期习题交互序列上, 而忽视了下一交互习题。

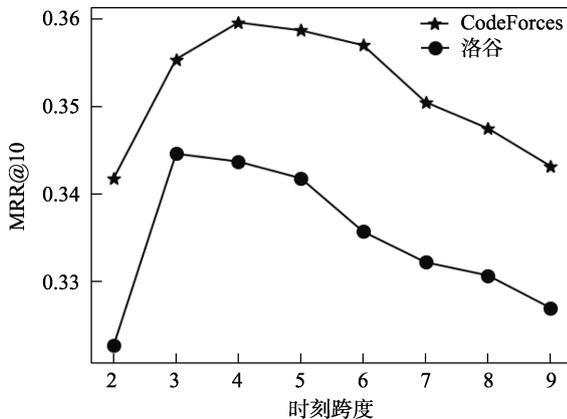


图 10 时刻跨度对模型效果的影响

## 6 总 结

本文充分关注了认知心理学中会影响学生答题

行为的因素, 提出了基于认知心理学和强化学习的个性化 OJ 习题推荐模型 MC-DRER。

MC-DRER 首次考虑了工作记忆对知识水平评估的影响, 结合工作记忆的相关工作<sup>[31-33]</sup>并针对 OJ 习题的特点, 提炼了习题题干中会对工作记忆造成挑战从而加大习题难度的 4 个要素作为习题特征的一部分来考虑学生的工作记忆对其答题行为的影响, 分别为习题描述词数、输入格式词数、输出格式词数和是否为英文题干. 通过实验证明了这 4 个要素与习题难度的关联性, 并验证了这 4 个要素的加入能提升习题推荐模型的准确性。

然后, 针对 OJ 习题的特点, 本文首次在长时记忆的遗忘与学习机制中考虑了学生编程语言熟悉程度, 结合认知心理学中遗忘曲线<sup>[34,35]</sup>和学习曲线<sup>[36]</sup>描述的人类大脑对长时记忆遗忘与学习的规律, 在相同知识的学习时间间隔和学习次数的基础上, 再选取了 2 个影响编程语言遗忘与学习的要素来建模学生的知识状态, 分别为学习时间间隔、总体学习次数. 通过实验证明了这 2 个要素、编程语言的熟悉程度和学生答题错误率四者之间的相关性, 并验证了本文所提出的遗忘与学习机制的优越性。

而后, 本文首次探索并选取习题会造成学生失误与猜对的要素来拟合学生因失误与猜对行为导致的知识掌握程度变化, 所选取的 6 个要素分别为习题的学生作答正确率与提交作答正确率之差、习题描述词数、输入格式词数、输出格式词数、是否为英文题干以及习题考查的知识点. 本文通过实验证明了本文所提出的失误与猜对机制能提升习题推荐模型的准确性。

接着, 本文首次提出能够考虑知识点间复杂多样的关系的习题推荐方法, 通过构造程序设计领域知识图谱并使用知识图谱嵌入技术来考虑外部信息对学生答题行为的影响. 通过实验证明了知识图谱能够提升习题推荐的准确性与合理性。

最后, 本文首次考虑了学生认知发展的非连续性, 通过引入认知突变奖励函数和隐式后序关系来考虑非连续性认知发展对学生答题行为的影响. 本文通过实验证明了非连续性认知发展在学生程序设计的过程中是一种普遍现象, 验证了两种策略的加入能提升习题推荐模型的准确性。

因为考虑了这些因素对学生答题行为的影响, 所以 MC-DRER 在 OJ 习题推荐任务上的性能要优于现有方法。

在未来的工作中, 会进一步将学生在答题时所提交的代码作为习题特征的一部分引入到模型中,

或使用算法分析代码从而为没有知识点标签的习题标注标签,使没有知识点标签的习题能够与知识图谱中的知识点产生联系,进而提升模型在 OJ 习题推荐上的性能。

### 参 考 文 献

- [1] Huang Z, Liu Q, Zhai C, et al. Exploring multi-objective exercise recommendations in online education systems//Proceedings of the 28th ACM International Conference on Information and Knowledge Management. Beijing, China, 2019: 1261-1270
- [2] Huang Zhen-Ya. Data mining techniques and applications for personalized learning[Ph.D. dissertation]. University of Science and Technology of China, Anhui, 2020(in Chinese)  
(黄振亚. 面向个性化学习的数据挖掘方法与应用研究[博士学位论文]. 中国科学技术大学, 安徽, 2020)
- [3] Walker A, Recker MM, Lawless K, et al. Collaborative information filtering: A review and an educational application. *International Journal of Artificial Intelligence in Education*, 2004, 14(1): 3-28
- [4] Hsu MH. A personalized English learning recommender system for ESL students. *Expert Systems with Applications*, 2008, 34(1): 683-688
- [5] Toledo RY, Mota YC, Martínez L. A recommender system for programming online judges using fuzzy information modeling. *Informatics*, 2018, 5(2): 17
- [6] Toledo RY, Martinez L. A recommendation approach for programming online judges supported by data preprocessing techniques. *Applied Intelligence*, 2017, 47(2): 277-290
- [7] Toledo RY, Mota YC. An e-learning collaborative filtering approach to suggest problems to solve in programming online judges. *International Journal of Distance Education Technologies (IJDET)*, 2014, 12(2): 51-65
- [8] Mao Hong-Yue. A study on knowledge recommender algorithm based on time transition[M.S. dissertation]. Tianjin University, Tianjin, 2017(in Chinese)  
(毛洪玥. 基于时间迁移的知识推荐算法研究[硕士学位论文]. 天津大学, 天津, 2017)
- [9] Gong J, Wang S, Wang J, et al. Attentional graph convolutional networks for knowledge concept recommendation in moocs in a heterogeneous view//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Xian, China, 2020: 79-88
- [10] Shao E, Guo S, Pardos Z A. Degree planning with PLAN-BERT: Multi-semester recommendation using future courses of interest//Proceedings of the AAAI Conference on Artificial Intelligence. Vancouver, Canada, 2021: 14920-14929
- [11] Zhu Tian-Yu, Huang Zhen-Ya, Chen En-Hong, et al. Cognitive diagnosis based personalized question recommendation. *Chinese Journal of Computers*, 2017, 41(1): 176-191(in Chinese)  
(朱天宇, 黄振亚, 陈恩红, 等. 基于认知诊断的个性化试题推荐方法. *计算机学报*, 2017, 41(1): 176-191)
- [12] Kang W, Zhang L, Li B, et al. Personalized exercise recommendation via implicit skills//Proceedings of the ACM Turing Celebration Conference-China. Chengdu, China, 2019: 1-6
- [13] Ma Xiao-Rui, Xu Yuan, Zhu Qun-Xiong. Personalized exercises recommendation method based on deep knowledge tracing. *Journal of Chinese Computer Systems*, 2020, 41(5): 990-995(in Chinese)  
(马晓睿, 徐圆, 朱群雄. 一种结合深度知识追踪的个性化习题推荐方法. *小型微型计算机系统*, 2020, 41(5):990-995)
- [14] Ai F, Chen Y, Guo Y, et al. Concept-aware deep knowledge tracing and exercise recommendation in an online learning system//Proceedings of the 12th International Conference on Educational Data Mining. Montral, Canada, 2019: 240-245
- [15] Zhao J, Bhatt S, Thille C, et al. Interpretable personalized knowledge tracing and next learning activity recommendation//Proceedings of the 7th ACM Conference on Learning @ Scale. New York, USA, 2020: 325-328
- [16] Huo Y, Wong D F, Ni L M, et al. Knowledge modeling via contextualized representations for LSTM-based personalized exercise recommendation. *Information Sciences*, 2020, 523: 266-278
- [17] Leighton J P, Gierl M J. *Cognitive diagnostic assessment for education: Theory and applications*. New York, USA: Cambridge University Press, 2007
- [18] Fan X. Item response theory and classical test theory: An empirical comparison of their item/person statistics. *Educational and psychological measurement*, 1998, 58(3): 357-381
- [19] De La Torre J. DINA model and parameter estimation: A didactic. *Journal of educational and behavioral statistics*, 2009, 34(1): 115-130
- [20] Reckase M D. *Multidimensional item response theory models*//Bonifay W. *Multidimensional item response theory*. New York, USA: Springer, 2009: 79-112
- [21] Cheng S, Liu Q, Chen E, et al. DIRT: Deep learning enhanced item response theory for cognitive diagnosis//Proceedings of the 28th ACM International Conference on Information and Knowledge Management. Beijing, China, 2019: 2397-2400
- [22] Wang F, Liu Q, Chen E, et al. Neural cognitive diagnosis for intelligent education systems//Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA, 2020, 34(04): 6153-6161
- [23] Corbett AT, Anderson JR. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-adapted Interaction*, 1994, 4(4): 253-278
- [24] Zhang K, Yao Y. A three learning states Bayesian knowledge tracing model. *Knowledge-Based Systems*, 2018, 148: 189-201
- [25] Agarwal D, Baker RS, Muraleedharan A. Dynamic knowledge tracing through data driven recency weights//Proceedings of the 13th International Conference on Educational Data Mining. Ifrane, Morocco, 2020: 725-729
- [26] Piech C, Bassen J, Huang J, et al. Deep knowledge tracing//Proceedings of the Advances in Neural Information Processing Systems. Quebec, Canada, 2015: 505-513
- [27] Zhang L, Xiong X, Zhao S, et al. Incorporating rich features into deep knowledge tracing//Proceedings of the 4th ACM Conference on Learning @ Scale. Massachusetts, USA, 2017: 169-172
- [28] Minn S, Yu Y, Desmarais MC, et al. Deep knowledge tracing and dynamic student classification for knowledge tracing//Proceedings of the 2018 IEEE International conference on data mining (ICDM). Singapore, 2018: 1182-1187
- [29] Liu D, Zhang Y, Zhang J, et al. Multiple features fusion attention mechanism enhanced deep knowledge tracing for student per-

- formance prediction. *IEEE Access*, 2020, 8: 194894-194903
- [30] Baddeley A. Working memory. *Science*, 1992, 255(5044): 556-559
- [31] Gathercole SE, Pickering SJ. Assessment of working memory in six-and seven-year-old children. *Journal of Educational Psychology*, 2000, 92(2): 377
- [32] Swanson HL, Sachse-Lee C. Mathematical problem solving and working memory in children with learning disabilities: Both executive and phonological processes are important. *Journal of Experimental Child Psychology*, 2001, 79(3): 294-321
- [33] Geary DC, Hoard MK, Byrd-Craven J, et al. Strategy choices in simple and complex addition: Contributions of working memory and counting knowledge for children with mathematical disability. *Journal of Experimental Child Psychology*, 2004, 88(2): 121-151
- [34] Ebbinghaus H. Memory: A contribution to experimental psychology. *Annals of Neurosciences*, 2013, 20(4): 155
- [35] Averell L, Heathcote A. The form of the forgetting curve and the fate of memories. *Journal of Mathematical Psychology*, 2011, 55(1): 25-35
- [36] Anzanello M J, Fogliatto F S. Learning curve models and applications: Literature review and research directions. *International Journal of Industrial Ergonomics*, 2011, 41(5): 573-583
- [37] Anderson J R. *Cognitive psychology and its implications*. UK: Macmillan, 2005
- [38] Sternberg RJ, Okagaki L. Continuity and discontinuity in intellectual development are not a matter of 'either-or'. *Human Development*, 1989, 32(3-4): 158-166
- [39] Thatcher RW. Maturation of the human frontal lobes: Physiological evidence for staging. *Developmental Neuropsychology*, 1991, 7(3): 397-419
- [40] Van Dijk M, Van Geert P. Wobbles, humps and sudden jumps: A case study of continuity, discontinuity and variability in early language development. *Infant and Child Development: An International Journal of Research and Practice*, 2007, 16(1): 7-33
- [41] Ahmed CF, Tanbeer SK, Jeong BS, et al. Efficient tree structures for high utility pattern mining in incremental databases. *IEEE Transactions on Knowledge and Data Engineering*, 2009, 21(12): 1708-1721
- [42] Zhang J, Shi X, King I, et al. Dynamic key-value memory networks for knowledge tracing//*Proceedings of the 26th international conference on World Wide Web*. Perth, Australia, 2017: 765-774
- [43] Mohammad K, Robert L and Michael M. How deep is knowledge tracing//*Proceedings of the 9th International Conference on Educational Data Mining*. North Carolina, USA, 2016: 94-101
- [44] Qiu Y, Qi Y, Lu H, et al. Does time matter? modeling the effect of time with bayesian knowledge tracing//*Proceedings of the 4th International Conference on Educational Data Mining*. Eindhoven, The Netherlands, 2011: 139-148
- [45] Pelánek R. Modeling students' memory for application in adaptive educational systems//*Proceedings of 8th International Conference on Educational Data Mining*. Madrid, Spain, 2015: 480-483
- [46] Huang Z, Liu Q, Chen Y, et al. Learning or forgetting? a dynamic approach for tracking the knowledge proficiency of students. *ACM Transactions on Information Systems*, 2020, 38(2): 1-33
- [47] Shen S, Liu Q, Chen E, et al. Learning process-consistent knowledge tracing//*Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Singapore, 2021: 1452-1460
- [48] Zhang J, Hao B, Chen B, et al. Hierarchical reinforcement learning for course recommendation in MOOCs//*Proceedings of the AAAI Conference on Artificial Intelligence*. Hawaii, USA, 2019: 435-442.
- [49] Liu Q, Tong S, Liu C, et al. Exploiting cognitive structure for adaptive learning//*Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Alaska, USA, 2019: 627-635
- [50] Zhu H, Tian F, Wu K, et al. A multi-constraint learning path recommendation algorithm based on knowledge map. *Knowledge-Based Systems*, 2018, 143: 102-114
- [51] Chen P, Lu Y, Zheng V W, et al. Prerequisite-driven deep knowledge tracing//*Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)*. Singapore, 2018: 39-48
- [52] Yang Y, Shen J, Qu Y, et al. GIKT: A graph-based interaction model for knowledge tracing//*Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Ghent, Belgium, 2020: 299-315
- [53] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data//*Proceedings of the Advances in Neural Information Processing Systems*. Lake Tahoe, USA, 2013: 2787-2795
- [54] Gao W, Liu Q, Huang Z, et al. Rcd: Relation map driven cognitive diagnosis for intelligent education systems//*Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Montreal, Canada, 2021: 501-510
- [55] Zhao Z, Yang Y, Li C, et al. GuessUNeed: Recommending courses via neural attention network and course prerequisite relation embeddings. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2020, 16(4): 1-17
- [56] Koshkareva M, Khlopotov M, Chernysheva A. The development of learning outcomes and prerequisite knowledge recommendation system//*Proceedings of the 13th International Conference on Education Technology and Computers*. Wuhan, China, 2021: 1-6
- [57] Wang Sheng. The research and application of data mining algorithms based on college entrance examination of knowledge points[M.S. dissertation]. North China University of Technology, Beijing, 2012(in Chinese)  
(王升. 高考知识点数据挖掘算法研究与应用[硕士学位论文]. 北方工业大学, 北京, 2012)
- [58] Li Jin, Chen Jian-Bin, He Yan-Qin. Association analysis of mathematic point in college entrance examination based on apriori algorithm. *Value Engineering*, 2012, 31(29): 211-212(in Chinese)  
(李锦, 陈建斌, 贺艳琴. 基于 Apriori 算法的高考数学知识点关联分析, *价值工程*, 2012, 31(29): 211-212)
- [59] Thai-Nghe N, Drumond L, Krohn-Grimberghe A, et al. Recommender system for predicting student performance. *Procedia Computer Science*, 2010, 1(2): 2811-2819
- [60] Rendle S. Factorization machines//*Proceedings of the 2010 IEEE*

- International Conference on Data Mining. Sydney, Australia, 2010: 995-1000
- [61] Hidasi B, Karatzoglou A, Baltrunas L, et al. Session-based recommendations with recurrent neural networks//Proceedings of 4th International Conference on Learning Representations. San Juan, Puerto Rico, 2016:1-10
- [62] Wang H, Zhang F, Wang J, et al. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems//Proceedings of the 27th ACM International Conference on Information and Knowledge Management. Torino, Italy, 2018: 417-426
- [63] Wang H, Zhang F, Zhang M, et al. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Alaska, USA, 2019: 968-977
- [64] Huang J, Zhao W X, Dou H, et al. Improving sequential recommendation with knowledge-enhanced memory networks//Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. Michigan, USA, 2018: 505-514
- [65] Wang P, Fan Y, Xia L, et al. KERL: A knowledge-guided reinforcement learning model for sequential recommendation//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Xi'an, China, 2020: 209-218
- [66] Choi Y, Lee Y, Cho J, et al. Towards an appropriate query, key, and value computation for knowledge tracing//Proceedings of the 7th ACM Conference on Learning @ Scale. New York, USA, 2020: 341-344



**JIN Tian-Cheng**, Ph. D. candidate. His research interests include educational data mining and recommender system.

**DOU Liang**, Ph. D., associate professor. Her research interests include software methods, AI for education.

**XIAO Chun-Yun**, M. S., lecturer. Her research interests include programming.

**ZHANG Wei**, Ph. D., associate researcher. His research interests include data mining.

**ZHOU Ai-Min**, Ph. D., researcher. His research interests include evolutionary computation and optimization, machine learning, and AI for education.

## Background

Personalized exercise recommendation is a research hotspot of EDM (Educational Data Mining), EDM is a kind of technology which combines the theories of pedagogy, computer science, psychology and statistics to solve the problems in teaching practice. Current exercise recommendation studies often depend on traditional recommendation algorithms, cognitive diagnosis models, knowledge tracing models or reinforcement learning technology, but most existing methods ignore students' cognitive process for information. To solve the problem above, this paper proposes a personalized OJ exercise recommendation method with memory and cognition merging based on cognitive psy-

chology and reinforcement learning technology, the method is able to consider the influence of students' working memory, long term memory, external information and discontinuity of students' cognitive development on their problem answering action. Finally, we conduct extensive experiments on two real-world datasets named Luogu and CodeForces, experimental results show that our approach can achieve higher performance than the state-of-the-art methods.

This work is supported by Youth Program of National Natural Science Foundation of China (No. 61907015) and the Science and Technology Commission of Shanghai Municipality Grant (No. 20511102502).