

北京交通大学

硕士专业学位论文

基于知识追踪的智能导学算法设计

Design of Intelligent Tutoring Algorithm Based on Knowledge  
Tracing

作者：艾方哲

导师：陈一帅


北京交通大学

2019年6月

## 学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。学校可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：

签字日期：2019年5月30日

导师签名：

签字日期：2019年5月30日

学校代码：10004

密级：公开

# 北京交通大学

## 硕士专业学位论文

基于知识追踪的智能导学算法设计

Design of Intelligent Tutoring Algorithm Based on Knowledge Tracing

作者姓名：艾方哲

学 号：17125001

导师姓名：陈一帅

职 称：副教授

工程硕士专业领域：信息网络

学位级别：硕士

北京交通大学

2019年6月

## 致谢

本论文的研究工作是在我的导师陈一帅老师的悉心指导下完成的。陈一帅老师严谨的科学态度，一丝不苟和精益求精的学术精神深深地影响我并激励我不断进取，探究科学真理。陈一帅老师学识渊博，视野广阔，不仅传授我知识，更传授了科研的方法，对我以后的工作和学习有着很大的影响。在此衷心感谢两年来陈一帅老师对我的悉心指导和关怀。

衷心感谢郭宇春老师、赵永祥老师和实验室里的所有老师在我研究生学习阶段对我的无私帮助和关怀。特别感谢郭宇春老师和赵永祥老师对我论文的审阅并提出宝贵的修改意见，在老师们的帮助下，我不断完善论文内容并最终成功完成毕业论文。同时，老师们严谨的科学态度与方法是我不断学习的方向，在此向各位老师表示诚挚的谢意。

衷心感谢北京世纪好未来教育科技有限公司的付国为老师对我的支持与帮助。感谢他创新的想法、对科研的数据支持、热情的鼓励，对我完成论文提供了宝贵的指导意见。在此向付国为老师表示衷心的感谢。

衷心感谢唐伟康师兄、陈勣师兄、李俊峰师兄、冯梦菲、王珍珠等同学在实验室工作和撰写论文期间对我的研究工作给予的热心帮助。在此向他们表示我的感谢之意。

最后，特别感谢一直无微不至的关心、支持我的父母，正是他们对我不断的鼓励与默默地付出，才使得我不断努力克服科研途中的困难，并顺利地完成学业，成为社会有用之才。

## 摘要

智能教育系统能够建立以学习者为中心的教育环境,提升学生学习效率,已被列入国家新一代人工智能发展规划。智能导学系统能够通过学生习题作答情况追踪学生各个知识概念的掌握程度,自适应地给学生推荐习题以加强学生对知识的掌握水平,提高学生的学习效率,对其的研究具有重要的理论价值和实际意义。

目前,制约智能导学系统性能的瓶颈有两个:1)知识追踪模型不能准确追踪学生各个知识概念的掌握程度,预测学生习题作答结果。原因在于当前的知识追踪模型没有设计合理的模型结构来有效使用题目的概念特征(知识点标签),导致模型性能不佳。2)习题推荐算法的设计依靠人工制定规则,效率不高,而基于启发式的习题推荐算法,只关注学生短期内的成绩提升,难以找到让学生能力稳步提升的习题。

针对上述问题,本文基于一个大规模、真实的在线教育系统的学生答题数据,测量观察了与学生作答结果相关的有效特征,设计了特征的有效表征方式和新的利用知识概念结构设计知识追踪模型,创造性地将深度强化学习应用于习题推荐中,设计了习题推荐策略算法,并基于实际数据进行了评估。具体贡献如下:

- (1) 为了提升知识追踪模型的性能,创新性地题目概念的等级结构引入神经网络的设计,设计了新的深度学习知识追踪模型,改进了模型追踪性能。首先,针对题目的多级知识概念特征,基于动态键值记忆网络 DKVMN,提出了概念敏感的知识追踪模型 DKVMN-CA。实验证明:该模型的神经网络结构能够有效地利用题目的多级知识概念特征进行知识追踪,明显提升了知识追踪模型的性能,相比 DKVMN 模型, AUC 值提升了 1.2%。其次,修改模型,在模型中加入题目难度、关卡特征、做题时间等特征,进一步提升了知识追踪模型的性能, AUC 值提升了 1.9%。
- (2) 以改进的知识追踪模型 DKVMN-CA 为学生模拟器,创新性地深度强化学习引入习题推荐算法中,优化习题推荐策略。该习题推荐策略能够根据学生作答历史,考虑学生的长期成绩提升进行习题推荐,以最大化学生在完成题目推荐序列后的各个知识概念掌握程度。实验证明:该算法相对于启发式习题推荐策略,能够找到使学生成绩提高的题目以持续地提升学生知识水平,解决了启发式推荐算法经过一定次数的习题推荐后,再无法找到合适题目以提升学生的成绩的问题。据我们所知,这是目前首次将深度增强学习应用于数学习题推荐,为未来的习题推荐方法提供了新的参考。

图 17 幅,表 2 个,参考文献 46 篇。

**关键词:** 知识追踪; 习题推荐; 分类预测; 深度强化学习

## ABSTRACT

The intelligent education system can establish a learner-centered education environment and improve students' learning efficiency. It has been included in the national new generation artificial intelligence development plan. The intelligent guidance system is able to track the mastery of each knowledge concept of students through student exercises, and adaptively recommend relevant exercises to students to enhance students' mastery of knowledge and improve students' learning efficiency. The research on it has important theoretical and practical significance.

At present, there are two bottlenecks that restrict the performance of the intelligent guidance system: 1) The knowledge tracing model cannot accurately track the knowledge status of students and predict the results of student exercises. The reason is that the current knowledge tracing model does not have a reasonable model structure to effectively use the exercise's concept features(knowledge point labels), resulting in poor model performance. 2) The design of the exercise recommendation algorithm relies on manual rules is not efficient. The heuristic-based exercise recommendation algorithm only focuses on the students' short-term performance improvement, and it is difficult to find exercises that make students' ability to improve steadily.

In view of the above problems, this paper is based on a large-scale, real online education system's exercise database and students practice data, measures and observes the effective features related to the students' answering results, designs the effective representation of features and the knowledge tracing model which uses the new knowledge concept structure. This paper creatively applies deep reinforcement learning to exercise recommendation, designs exercise recommendation policy algorithm, and evaluated the policy based on real data. The specific contributions are as follows:

- (1) In order to improve the performance of the knowledge tracing model, the hierarchical structure of the exercise concept is innovatively introduced into the design of the neural network, and a new deep learning knowledge tracing model is designed to improve the model tracing performance. Firstly, based on the multi-level knowledge concept feature of the exercise and the dynamic key-value memory network DKVMN, a concept-aware knowledge tracing model DKVMN-CA is proposed. The experiment proves that the neural network structure of the model can effectively utilize the multi-level knowledge concept

features of the exercise for knowledge tracing, which significantly improves the performance of the knowledge tracing model and has 1.2% of AUC higher than DKVMN. Secondly, the model is modified by adding features such as difficulty, stage features, and the duration of finishing exercise, which further improves the performance of the knowledge tracing model and has 1.9% of AUC higher than DKVMN.

- (2) With the improved knowledge tracing model DKVMN-CA as the student simulator, innovatively introduce deep reinforcement learning into the exercise recommendation algorithm to optimize the exercise recommendation policy. The exercise recommendation policy can be based on the student's history of doing exercises, considering the long-term performance improvement of the students to recommend exercises, in order to maximize the degree of knowledge of the students after completing the exercise recommendation sequence. The experiment proves that the algorithm can find the exercises that improve the students' grades in order to continuously improve the students' knowledge level, and solve the traditional heuristic recommendation algorithm problem that after a certain number of exercises, the exercise can not be found to raise the student's grades. To the best of our knowledge, this is the first time that deep reinforcement learning has been applied to the math exercise recommendation, providing a new reference for future exercise recommendation methods.

Figure 17, table 2, reference 46.

**KEYWORDS:** Knowledge tracing; exercise recommendation; classification prediction; deep reinforcement learning;

## 目录

摘要 .....	III
ABSTRACT .....	IV
1 引言 .....	1
1.1 研究背景和意义 .....	1
1.2 国内外研究现状 .....	1
1.2.1 知识追踪研究现状及其问题 .....	2
1.2.2 导学推荐策略研究现状及其问题 .....	4
1.3 研究内容 .....	6
1.3.1 知识追踪模型 .....	6
1.3.2 习题推荐系统 .....	7
1.4 本论文的主要贡献 .....	8
1.5 本论文的组织结构 .....	9
2 技术背景 .....	10
2.1 IPS 智能练习系统 .....	10
2.2 机器学习 .....	11
2.2.1 强化学习 .....	12
2.2.2 神经网络 .....	14
2.2.3 循环神经网络 .....	15
2.2.4 深度强化学习 .....	17
2.2.5 交叉熵损失函数 .....	18
2.2.6 策略梯度算法 .....	19
2.2.7 特征表达 .....	20
2.3 模型评估 .....	21
2.3.1 评估方法 .....	21
2.3.2 性能度量 .....	22
2.4 开发平台 .....	24
2.4.1 Scikit-learn 算法库 .....	24
2.4.2 Anaconda 集成环境 .....	25
2.4.3 Rllab 库 .....	25



2.4.4 TensorFlow 框架.....	25
2.7 本章小结.....	26
3 深度知识追踪模型.....	28
3.1 基本思路.....	28
3.2 特征统计与表达.....	28
3.2.1 数据集介绍.....	28
3.2.2 知识概念.....	29
3.2.3 题目关卡.....	29
3.2.4 做题时间.....	30
3.2.5 题目难度.....	31
3.3 模型构建.....	32
3.3.1 知识概念记忆矩阵.....	33
3.3.2 知识概念权重.....	34
3.3.3 作答结果预测.....	35
3.3.4 记忆矩阵更新.....	35
3.4 性能评估.....	36
3.4.1 数据预处理.....	36
3.4.2 实验细节.....	37
3.4.3 知识概念结构的增益.....	37
3.4.4 其他习题特征的增益.....	38
3.5 本章小结.....	38
4 习题推荐系统.....	40
4.1 基本思路.....	40
4.2 习题推荐模型.....	40
4.3 策略优化.....	41
4.4 知识增长过程评估.....	42
4.5 习题推荐评估.....	44
4.6 本章小结.....	46
5 总结及展望.....	48
5.1 总结.....	48
5.2 未来工作展望.....	49
参考文献.....	50

---

作者简历及攻读硕士学位期间取得的研究成果.....	53
独创性声明.....	54
学位论文数据集.....	55

## 缩略词表

英文缩写	英文全称	中文全称
RL	Reinforcement Learning	强化学习
POMDP	Partially Observable Markov Decision Process	部分可观察马尔可夫决策过程
TRPO	Trust region policy optimization	信赖域策略优化
TPR	True Positive Ratio	真正率
FPR	False Positive Ratio	假正率
CDF	Cumulative Distribution Function	累积分布函数
ROC	Receiver Operating Characteristic Curve	受试者工作特征曲线
AUC	Area Under Receiver Operating Characteristic Curve	ROC 曲线下与坐标轴围成的面积
DKVMN	Dynamic Key Value Memory Network	动态键值记忆网络
DKVMN-CA	Concept-Aware Dynamic Key Value Memory Network	概念敏感的动态键值记忆网络
BKT	Bayesian Knowledge Tracing	贝叶斯知识追踪
DKT	Deep Knowledge Tracing	深度知识追踪

# 1 引言

## 1.1 研究背景和意义

基于互联网的在线教育正越来越多地应用到教育实践中，比如慕课、学而思智能教育系统（IPS: Intelligent Practice System）等。这些教育系统不仅通过网络的方式给学生传授课程，也有针对课程的习题提供给学生进行练习以巩固学习的内容。

智能教育对于提升学生的学习效率具有重要意义，已经引起社会和学术界的广泛重视。2017年国务院发布《新一代人工智能发展规划》明确指出发展智能教育，推动人工智能在教学、管理、资源建设等全流程应用，建立以学习者为中心的教育环境，提供精准推送的教育服务，实现日常教育和终身教育定制化。但是，目前因为教育资源有限，教师没有精力和时间为每一个学生专门定制练习方案，只能按最典型同学的接受能力和学习方法，给所有学生布置同样的习题。这导致为了每个学生达到理想的教学效果，大量低效的习题给学生带来了沉重的负担，降低了学习效率。解决该问题的方法在于通过智能导学系统实现个性化教育，个性化教育能够基于学生的知识状态进行个性化教学内容推荐，综合考虑了学生的知识，能力，行为特征和当前学习环节的时间约束等因素。从而节省学生学习时间，达到更快地提升学生能力的目标。

本文的研究围绕智能教育，设计和实现了智能导学算法，具有重要的理论价值与实际意义。它充分地利用了在线教育系统的教育大数据资源，改进知识追踪的性能，提升了学生模型的预测准确度。采用先进的深度强化学习算法训练习题推荐策略，对学生进行个性化导学。有效的个性化导学对于家长，减轻了低效的课外辅导的经济压力，对于学生，因材施教，高效学习能大幅度的减轻学生的学习负担，达到理想的学习效果，具有重大的社会经济价值。

## 1.2 国内外研究现状

智能导学系统现在有大量的研究工作。下面对这些工作进行介绍，并指出它们的问题。这些问题制约了智能教育系统的发展，对它们的研究和改进对提升智能教

育水平具有重要的理论价值和实际意义。

### 1.2.1 知识追踪研究现状及其问题

基于教育大数据，通过知识追踪的方法建立学生模型能够较为准确地进行作答结果预测，且智能导学系统中的学生模型能够准确预测习题作答结果是十分重要的<sup>[1]</sup>。知识追踪是根据学生的历史学习行为对学生的知识水平建模，以便我们能准确地预测学生对于各个知识概念的掌握程度，以及学生在未来学习行为的表现。准确可靠的知识追踪意味着我们可以根据学生的自身的知识状态，给他们推荐合适的练习题目，比如，推荐给学生薄弱知识概念关联的题目，而过于困难或者过于简单的题目不应该被推荐，从而可以给学生进行高效的个性化教学。

当前知识追踪的研究及其问题主要在以下几个方面：

1. 贝叶斯知识追踪用二元变量表示知识掌握程度，且认为知识概念之间没有相关性，与实际不符。

贝叶斯知识追踪（BKT: Bayesian Knowledge Tracing）是一种很流行的用于构建学生学习的时序模型。BKT 模型中提出了一组用于表示学生知识状态的二元隐变量，每一个隐变量表示针对某一个知识概念学生掌握还是没掌握。通过学生在某些知识概念的题目的作答情况（正确还是错误），建立隐马尔科夫模型，来更新各个知识概念二元隐变量的概率，从而预测某一知识概念的题目是否能够正确作答<sup>[2]</sup>。但是在 BKT 模型假定某一知识概念一旦掌握就不会被遗忘。基于贝叶斯知识追踪模型，后来又有相关的拓展，比如考虑概念之间关系的模型：北京大学的张铭教授提出了 Multi-Grained-BKT 和 Historical-BKT 模型，能够反映一门课程中各知识概念的层级结构和相关关系，描述学生知识生长过程中的知识层级和先后顺序，从而提高模型的准确度<sup>[3]</sup>。还有考虑学生在未掌握知识的情况下猜对答案或者在已掌握知识的情况由于失误错误作答的概率<sup>[4]</sup>，对于每个学生的先验知识估计<sup>[5]</sup>，题目的难度估计等<sup>[6]</sup>。即使这些拓展能够提升 BKT 的表现，但是 BKT 模型依然存在问题：1) 学生的知识状态用二元变量表示并不是很实际，无法表现知识掌握程度。2) 模型假定知识概念之间是没有相关性的，而这与实际不符合的。

2. PFA, LFA 等动态概率模型依赖题目的知识概念标记才能进行知识追踪，准确率不高。

PFA (Performance Factors Analysis)<sup>[7]</sup>与 LFA (Learning Factors Analysis)<sup>[8]</sup>等动态的概率模型在知识追踪方面也有与 BKT 想媲美的表现，能够根据学生的题目作答情况，来更新学生对一个知识概念的掌握程度，并且 PFA 与 BKT 的集成模型进一步提升了单个模型的表现<sup>[9]</sup>。但是这些模型都十分依赖于题目的知识概念标记。

将项目反映理论 (IRT: Item Response Theory) 与知识追踪模型结合, 实现了不同学生个体学习过程建模与高准确率的预测<sup>[10,11]</sup>。其中, IRT、PFA 都是线性模型。最近, 研究者们又提出非线性模型, 以提高模型表示能力, 获得更好跟踪效果。

- (1) 支持多概念习题的非线性模型: 美国普林斯顿大学的 Andrew S. Lan 等研究者提出了一种新型非线性学生答题模型 (BLAh: Boolean Logic Analysis)。它用布尔逻辑描述一个题中多个概念之间的相互关系, 得到学生的答题模型<sup>[12]</sup>。
- (2) 基于技巧向量的模型: 康奈尔大学的 Siddharth Reddy 等人基于大量学生在线学习数据, 得到了学生和课程材料的隐式技巧向量表征。基于该向量表征, 就可以在向量空间上进行学生和课程材料之间的匹配和推荐<sup>[13]</sup>。
3. 深度知识追踪由于没有设计合理的网络结构利用题目知识概念特征, 模型性能仍有待提高。

最近, 深度学习被引入学生知识追踪领域, 相比于贝叶斯知识追踪取得了较好效果<sup>[14]</sup>, 这方面工作有:

- (1) 深度知识跟踪模型 (DKT: Deep Knowledge Tracing): 斯坦福大学的 Chris Piech 等人将深度学习方法应用于知识追踪问题, 提出了 DKT。他们采用的是循环神经网络 (RNN: Recurrent Neural Network) 中的长短期记忆模型 (LSTM: Long Short-Term Memory), 取得了较好的模型效果<sup>[15]</sup>。但是由于模型采用隐藏层来表示所有知识概念掌握程度, 因此, 无法追踪预测某一个特定的概念的掌握情况。随后, 香港科技大学的 Chun-Kit Yeung 等研究者针对 DKT 模型的两个问题: 1) 模型不准确: 有时学生实际上表现很好, 但模型预测他的表现并不好。2) 模型预测的学生表现很不稳定, 忽高忽低, 而实际上学生对一个知识的掌握程度是比较稳定的, 在模型的损失函数中引入了正则化, 改进了模型性能<sup>[16]</sup>。在深度知识追踪模型的基础上, 文献<sup>[17]</sup>提出通过双向 LSTM 网络提取题目文本特征, 并将文本特征加入预测模型, 并且加入注意力机制提升了模型预测效果。文献<sup>[18,19]</sup>通过向 RNN 加入更多特征扩展来提升知识追踪效果, 如完成时间, 作答次数, 是否请求提示等。
- (2) 动态键值记忆网络模型 (DKVMN): 香港中文大学的 Irwin King 教授等研究者针对 BKT、DKT 的问题, 提出了可以发现题目概念权重关系, 并且对学生掌握不同的知识概念的情况进行追踪的网络结构, 以更准确地预测学生答题准确率与各个概念掌握程度。他们在模拟数据集以及真实数据集上进行测试, 取得了比 BKT 和 DKT 更好的预测效果<sup>[20]</sup>。文献<sup>[21]</sup>通过多任务学习, 进一步改进 DKVMN 的性能: 将知识追踪与请求提示预测进

行联合训练，进行多任务预测，从而提升了模型的性能。

随着深度学习的不断发展，基于神经网络的知识追踪的模型性能不断提升，并且高于贝叶斯等的知识追踪方法。本文就采用深度神经网络，并结合实际真实的在线教育数据，改进知识追踪模型，进一步提升知识追踪的效果，并以知识追踪建立学生模拟器或学生环境，应用于训练习题推荐策略。

## 1.2.2 导学推荐策略研究现状及其问题

导学推荐策略主要是用于根据学生的知识状态给学生推荐合适的学习内容以提升学生的成绩。用于智能教育的推荐算法主要有下面三种：1) 基于学生知识状态的启发式推荐算法。2) 基于 Bandit 的推荐算法。3) 基于深度增强学习的推荐算法。下面将对这三种算法及其问题做具体阐述：

### 1. 基于启发式的习题推荐策略，不一定是最优的习题推荐策略，在多次习题推荐之后学生不一定能达到最好的成绩。

启发式推荐算法基于学生知识追踪模型输出的、对学生当前知识和能力状态的估计，进行课程材料推送。最近有如下工作：

- (1) 美国哈佛大学的 Yigal Rosen 等研究者扩展了传统 BKT 模型，给不同难度习题不同模型参数，支持一个题多个知识概念，也考虑知识概念的先后顺序，然后基于模型评估了两种启发式推荐算法：补救和连续学习。前者注意根据学生做错的情况进行补救，而后者注重知识概念的连贯性。最后发现补救策略与学习收益的大幅增加相关<sup>[22]</sup>。
- (2) 德国哈斯帕拉特研究所的 Ralf Teusner<sup>[23]</sup>等研究者提出，应根据发现的学生弱点，提供补救练习。为此，他们设计了一种考虑知识概念之间关联和学生知识掌握程度模型的练习推荐算法。该算法综合考虑题目难度、知识概念组成、学生做题时间，根据一个设定好的评分函数进行系统的排序。他们实现的具体推荐策略是：a) 移走太难的、学生还没有掌握基础概念的系统；b) 选择让学生学习收益最大的。但是他们的算法是启发式的，没有系统探索最优策略<sup>[23]</sup>。
- (3) 卡内基梅隆大学的 Irene-Angelica Chounta 参考文献等人提出通过学生知识追踪模型预测的学生做题正确概率来推荐。如果正确概率为 0 或 100% 都不推荐，而是推荐那些概率为 50% 的。该方法有一定道理，即：如果预测学生正确概念为 50%，这就意味着学生有一定基础，但因为还缺乏某些知识或技能，所以还不能完全做对，因此就要向该学生推荐该习题，让学生通过做习题和看讲解，学到新知识，增长新能力。这一算法的问题是：

这个 50% 的判断门限的设定是启发式的，并不一定最优<sup>[24]</sup>。

- (4) Google 提出 Expectimax 搜索算法<sup>[15]</sup>，将对学生的习题推荐过程建模为马尔可夫决策过程，每一步的习题推荐都会遍历所有的备选题目，选择完成该题目练习后使学生获得最高的奖赏期望值的题目进行推荐。但是这仍然是启发式的推荐策略，没有考虑推荐习题对学生成绩提升的长期影响，没有探究整条习题推荐序列对学生能力提升的影响。

## 2. 基于 Bandit 的算法在短期内能够有效地提升学生成绩，但是对于多次的习题推荐并不一定能够持续地提升学生成绩。

MAB (Multi-Armed Bandits)<sup>[25]</sup>和 Contextual Bandits<sup>[26]</sup>是新闻推荐常用的一种算法，它能够根据历史推荐的效果（回报 Reward），决定下一步推荐什么新闻。最近它也被用于个性化教育领域中。研究工作有：

- (1) MAB 算法：文献<sup>[27]</sup>提出基于学生的当前的知识状态，由专家划定最近发展区 ZPD (Zone of Proximal Development)，然后通过 MAB 算法选择最合适的题目推荐给学生去做。这种算法能够通过探索方式发现学生的学习特点，但是并不是高效的，因为每个学生都需要这种独立的探索过程来确定导学策略，而且专家对于 ZPD 的划定也浪费了大量的人力。斯坦福大学的 Tong Mu 等人采用 MAB 算法，给学生推荐习题。他们的目标是：在有限时间内，最大化学生获得的技巧。研究场景为：由易到难地完成一个学习任务，比如：加法（其中包括各种子技巧，如单位数相加，进位，等）。因此，它首先构造子技巧之间的知识图谱。然后基于该图谱，根据学生做题情况，在合理的子技巧区间内用 MAB 算法探索发现最适合学生的学习内容。他们采用的 MAB 算法回报是学生知识增长的速度。它的推荐单位是子技巧，所以，对一个子技巧，会反复让学生练习<sup>[28]</sup>。在他们最近的一个工作中，他们又加入了遗忘模型，将已经学过但随着时间流逝学生可能忘记的内容加入 MAB 选项中，进行复习<sup>[29]</sup>。
- (2) Contextual Bandits 算法：美国莱斯大学的 Andrew S. Lan 用基于 Logistic 回归的 Contextual Bandits 算法，研究了物理课程的教学单元推荐问题。他们的研究场景是首先由学生进行三次测试，然后将测试的结果作为 Contextual Bandits 算法的输入，智能选择一个教学单元（如：做练习、观看教学视频等），最后再做一次测验，看提分效果<sup>[30]</sup>。斯坦福大学的 Y. Alex Kolchinski 等研究者研究了类似的问题。他们设计了一个实验，从学生的自然语言答题内容中提取特征（利用深度学习的 Glove 词嵌入），训练两个线性回归模型，分别预测不同反馈下学生的学习增益，最后选择有最大增益的反馈给学生。整个问题被作为一个 Contextual bandits 问题，即根据



输入（学生的回答），用机器学习模型获得最佳动作（在这里是给用户推荐什么内容。当然也可以是提醒学生休息，做另一个练习等），令收益最大（即提分效果最大）<sup>[31]</sup>。

### 3. 基于深度增强学习的推荐算法刚刚被应用于单词记忆等简单学习任务的导学，但是还未被用于复杂学习任务的导学。

文献<sup>[32]</sup>提出将深度强化学习应用于导学系统，文献<sup>[33]</sup>将教学过程建模为 POMDP 过程，美国伍斯特理工学院的 Jacob Whitehill 等人研究了基于图片的单词教学，用 POMDP 决定教师的下一步动作：教学、提问，还是测试。为了求解这个 POMDP 问题，他们采用了策略梯度优化算法<sup>[34]</sup>。伯克利大学的 Siddharth Reddy 等研究人员将基于深度学习的增强学习算法应用于单词复习问题的研究。他们基于学生遗忘模型，采用基于循环神经网络 RNN 的增强学习 TRPO 算法，求解这个 POMDP 问题，决定要推送的复习单词<sup>[35]</sup>。文献<sup>[36]</sup>在具有挑战性的真实世界教育游戏应用程序中，开发一种用于自适应辅导任务的强化学习智能体，它使用离线的重要性抽样的方法，来选择表征与设置超参数，而没有采用基于真实学生昂贵的在线实验。该方法不用遍历搜索，而是通过取样用户轨迹实现算法的优化。

随着强化学习与深度强化学习不断发展，基于强化学习的推荐策略越来越多地应用于教育导学方面，其推荐策略考虑学生长期的能力提升，更有利于持续提升学生能力。本文将针对新的教育应用场景，数学习题推荐，探索使用深度强化学习完成习题推荐策略。

## 1.3 研究内容

目前的研究工作，尚存两个重要问题需要解决：1) 知识追踪模型预测学生作答结果准确率不高；2) 习题推荐策略不能持续地提升学生的能力。本文即针对上述两个问题，进行研究。

本文提出一种基于知识追踪的智能导学方法，进行个性化题目推荐。具体在以下两个方面展开工作：1) 改进知识追踪模型，提升知识追踪模型性能；2) 设计习题推荐系统，实现个性化导学。下面将对这两个方面的研究内容做具体阐述。

### 1.3.1 知识追踪模型

因为基于神经网络的知识追踪相比于传统的贝叶斯知识追踪模型具有更好的性能<sup>[14]</sup>，所以，我们通过设计合理的神经网络结构来提升预测习题作答结果的准确性。我们基于在线教育的学生习题作答数据集（包括题目信息，学生作答结果详

情等)提取有效的题目特征与学生学习行为特征,改进现有的知识追踪模型,提升知识追踪模型的性能。

具体来说,基于学而思 IPS 智能练习系统真实的学生习题作答数据集,数据集包括了题目信息,如题目难度,题目知识概念,关卡等,还有学生的做题行为特征如习题作答时间,作答结果等。测量并观察学生的习题作答结果与多个题目特征,做题时间的相关性,特别针对题目的多级知识概念特征,基于动态键值记忆网络(DKVMN: Dynamic Key Value Memory Network)进行模型修改,以合理地利用多级知识概念特征进行知识追踪,又在此基础上加入关卡,难度,作答时间特征以进一步提升模型的预测准确率,并分析不同的特征对于模型性能的影响程度。

### 1.3.2 习题推荐系统

我们设计合理的奖赏函数,通过深度强化学习优化习题推荐策略,实现个性化习题推荐。深度强化学习最近在训练智能体玩游戏,击败人类世界冠军以及控制机器人方面取得了显著成功<sup>[37]</sup>。我们认为,习题推荐可以视为智能体观察学生的历史作答情况,选择合适的习题推荐动作以提升学生的知识掌握状态,当实现其教育目标时候获得奖赏的过程。因此,我们将习题推荐建模为部分可观测马尔可夫决策过程(POMDP: Partially Observable Markov Decision Process),利用知识追踪建立学生模拟器,通过深度增强学习算法解决 POMDP 问题,实现个性化习题推荐策略的优化。

具体研究内容如下:

- (1) 基于前面获得的深度追踪模型,我们设计实现了一个基于强化学习的学生智能导学算法的实验和优化框架。深度强化学习数据采集十分复杂:智能体的训练需要与环境进行大量的交互,通过智能体与真实学生之间进行交互来获得大量的样本是不可行的,并且强化学习奖赏函数难以设置,效果不稳定等原因导致深度强化学习应用难以大量落地。因此,我们参考 Reddy 提出的采样方法<sup>[35]</sup>,使用改进的知识追踪模型作为学生模拟器,以智能体与学生模拟器进行交互来获得足够的样本。基于该框架,我们能够根据推荐题目的作答结果,自动更新学生的知识状态,从而能够根据学生做题历史进行多次的自适应题目推荐。
- (2) 基于该框架,结合实际习题推荐应用场景,将习题推荐建模为 POMDP 过程,设计合理的奖赏函数,采用置信域策略优化算法(TRPO: Trust Region Policy Optimization)解决 POMDP 问题,优化习题推荐策略。使习题推荐策略能够自适应地根据学生历史习题作答情况进行题目推荐。实验

评估多轮，通过设计对照实验，对比基于强化学习优化的习题推荐策略与启发式习题推荐策略在提升学生成绩方面的差别，可视化习题推荐序列与学生知识状态变化情况，分析采用强化学习优化的习题推荐策略的合理性与有效性。

## 1.4 本论文的主要贡献

本文的工作和贡献如下：

- (1) 为了提升知识追踪模型的性能，创新性地将题目概念的等级结构引入神经网络的设计，设计了新的深度学习知识追踪模型，改进了模型追踪性能。我们采用真实的在线教育系统的学生做题行为数据，通过测量分析学生做题行为特征，习题难度，多级知识概念特征等，与学生作答结果之间的相关性，并采用合理的特征表示方法进行特征表示。基于动态键值记忆网络，针对题目的多级知识概念特征进行模型修改，以有效地利用多级知识概念特征，我们称之为概念敏感的动态键值记忆网络（DKVMN-CA: Concept-Aware Dynamic Key Value Memory Network）。经过实验，加入多级知识概念特征的 DKVMN-CA 模型相比于 DKVMN 在平均 AUC 上提升 1.2%，明显高于 DKVMN 对于 DKT 的 0.1% 的提升。且基于多级知识概念进行模型修改，能有效地利用知识概念特征，相比于直接进行向量拼接的方法 AUC 高 1%。同时，其他特征加入进一步提升了 DKVMN-CA 模型的表现，加入关卡，做题时间特征，DKVMN-CA 模型 AUC 最高 0.739，相比于 DKT 模型最高 AUC 高出 2.7%，相比于目前知识追踪效果最好的 DKVMN 模型最高 AUC 提升 1.9%。
- (2) 创新性将深度增强学习引入习题推荐算法中，提出基于深度强化学习的习题推荐策略。首先将习题推荐建模为部分可观测马尔可夫决策过程，习题推荐策略能够根据学生的做题历史来推荐习题。通过知识追踪 DKVMN-CA 建立学生模拟器，通过学生模拟器，可以明确地表示学生在各个知识概念的知识状态，且准确地预测某一学生正确作答某一道题目的概率，并将其应用于深度强化学习的智能体训练与策略评估中去。使用强化学习算法置信域策略优化算法来获得习题推荐策略。设计对比实验，将强化学习习题推荐策略与启发式习题推荐策略进行对比，将两个相同的学生模拟器分别通过两个策略的 50 次的习题推荐，发现通过强化学习训练获得策略能更持续且有效地提升学生的成绩，知识水平相比于启发式 Expectimax<sup>[15]</sup> 算法高 5%，且差距随着推荐次数的增加，进一步扩大。并且我们通过可视

化真实习题推荐与学生知识状态的变化过程，证明了深度增强学习在智能习题推荐中的有效性和可行性。据我们所知，这是目前首次将深度增强学习应用于数学习题推荐的具体应用场景。

## 1.5 本论文的组织结构

本文的组织结构如下：

第二章主要介绍了本论文相关的技术背景，包括所使用的数据，研究所采用的机器学习算法与评估方法，模型的开发平台等。

第三章详细介绍了我们通过观察测量，发现与学生习题作答结果相关的不同特征，对特征进行了合理的特征表达，并详细介绍了基于多级知识概念特征，我们改进的知识追踪模型 DKVMN-CA。并通过对比实验，评估了我们提出的模型性能。

第四章详细介绍了习题推荐模型，将习题推荐建模为部分可观测马尔可夫决策过程，以改进的知识追踪模型 DKVMN-CA 建立学生模拟器，采用强化学习算法对习题推荐策略进行优化，并设计与启发式推荐算法的对比实验，可视化习题推荐路径与学生知识状态变化情况，评估习题推荐策略的性能。

第五章总结全文的工作与本论文的贡献点，并对智能教育中的习题推荐在未来可深度研究改进的工作提出看法。

## 2 技术背景

本章介绍论文中所应用的在线教育系统和主要技术。首先介绍 IPS 智能练习系统，即好未来（TAL）学而思的在线教育系统及其数据；然后介绍了机器学习方法的原理，包括神经网络，强化学习等，以及对模型进行评估的方法。最后介绍了实现机器算法的科学计算库与开发平台。

### 2.1 IPS 智能练习系统

北京世纪好未来教育科技有限公司（NYSE: TAL）是一个以智慧教育和开放平台为主体，探索未来教育新模式的科技教育公司，前身为学而思。学而思 IPS 是学而思配合现有的课堂教学内容，研发的一套“智能练习系统”。每节课有多个学习关卡，每个关卡都有丰富的学习试题和视频，提高学生的学习效率，帮助学生养成良好的学习习惯。

IPS 系统主要有 6 个关卡类型：

- 1) 预习关卡：学生在家看完预习视频并做完 3 道预习题。
- 2) 课前诊断：学生在老师正式授课之前在教室中所做的练习题。
- 3) 课后诊断：整堂课学习结束，对学生课堂所学内容进行测试。
- 4) 作业：老师给学生所布置的家庭作业练习题。
- 5) 复习巩固：学生在完成作业之后，对所学内容的复习习题。
- 6) 阶段复习：在学生完成一段时间的学习后，对这段时间所学内容进行复习巩固的练习题。

IPS 是一个自我导学的学习系统，学生可以根据自身的实际情况选择任一关卡进行练习，在做一关卡的习题时能选择离开此关卡进入别的关卡进行学习。学生也可以按照老师的要求进行学习。学生通过 IPS 进行练习，IPS 系统会详细地记录学生行为信息，包括做题时间，题目 ID，题目知识概念，作答结果（正确与否），题目难度，关卡类型等。

IPS 是由专家来设计每一关卡的具体题目，每个题目都有专家所给定的三级知识概念结构，如图 2-1 所示，三级知识概念结构是分层的树形结构。一个一级知识概念下有多个二级知识概念，同样地，一个二级知识概念有多个三级知识概念。比如，某道题目的一级知识概念是“数论”，其二级知识概念为“质数与合数”，三级知识概念为“分解质因数”；再比如，某道题目一级知识概念为“数论”，二级知识概念为“完全平方数”，三级知识概念为“平方数判定”。

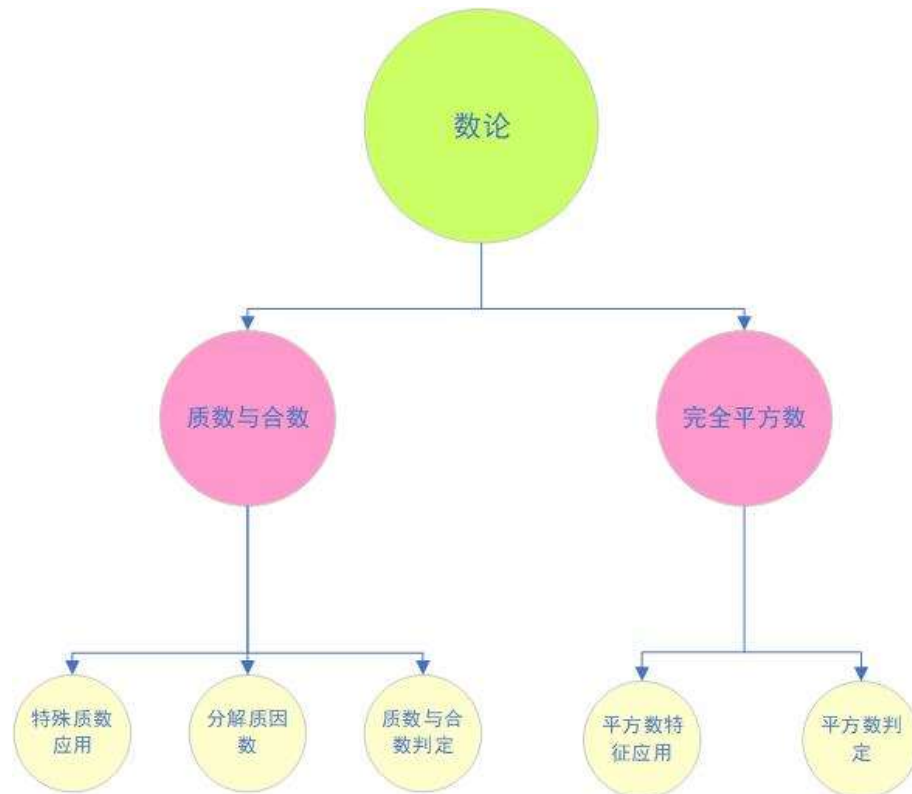


图 2-1 三级知识概念结构图

Figure 2-1 Three-level Knowledge Point Structure

本文中所研究的智能导学系统的数据来源即为学而思 IPS 系统的教育数据。

## 2.2 机器学习

机器学习是计算机科学中的一个重要领域，也是人工智能的子领域。但它与传统的计算方法不同。在传统计算中，算法是计算机用于计算或解决问题的明确编程指令集。机器学习算法用于建立数学模型的数据称为“训练数据”，机器学习算法是让计算机在训练数据上进行训练，并使用统计分析的方法以输出落在特定范围内的值。因此，机器学习有助于计算机根据样本数据构建模型，从而根据数据输入实现自动化决策过程。它是利用现有的数据预测未来的数据。机器学习算法用于各种应用，在人脸识别、垃圾邮件过滤、网络入侵者的检测和计算机视觉等不可行甚至不可能编写算法来执行任务的领域很重要。机器学习与计算统计密切相关，计算统计侧重于使用计算机进行预测。数学优化研究为机器学习领域提供了方法，理论和应用领域。数据挖掘是机器学习中的一个研究领域，侧重于通过无监督学习进行

探索性数据分析。同时，在自然语言处理与语音识别领域等，机器学习也有广泛的应用。

两种最广泛采用的机器学习方法是监督学习和无监督学习。监督学习基于由人类标记的示例输入和输出数据来训练模型，让模型能够根据新输入的数据进行合理的预测，这类学习方法主要以分类与回归任务为主，如，车流量预测，心脏病发作的预测。无监督学习，其为算法提供没有标记数据以允许其在其输入数据内找到内在的数据结构或者隐藏的模式，如聚类任务是一种最常用的无监督学习技术，包括 K-均值算法，密度聚类，层次聚类等，其在对象识别，市场调查等方面有着广泛的应用。

强化学习被认为是与监督学习和无监督学习一起的三种机器学习范式，它与监督学习不同之处在于，不需要呈现正确的输入与输出对，而是通过最大化奖赏收益，以与环境不断交互的方式来发现在不同状态下的最优的决策动作，从而得到策略。强化学习与深度神经网络相结合而发展出的深度强化学习已经取得突破性进展，其在推荐系统，自动驾驶，金融，对话系统等领域得到应用。

下面我们将针对智能导学算法所涉及的几种机器学习算法进行介绍，具体如下：

## 2.2.1 强化学习

强化学习主要由智能体（Agent）、环境（Environment）、状态（State）、动作（Action）、奖励（Reward）等基本元素组成。其中智能体作为强化学习的本体，作为学习者或者决策者。而环境为强化学习智能体以外的一切，主要由表示状态的数据集合构成，各种状态数据组成了状态集合。动作则是智能体根据当前的策略可以做出的动作，动作集则是智能体可以做出的所有动作组成的集合。奖赏是智能体在某个状态做出对应的动作后所获得的反馈信号，包括正反馈与负反馈，这些反馈均可以通过设计奖赏函数来进行表示。

基本的强化学习可以建模为一个马尔可夫决策过程，且在许多工作中，假定智能体是完全可观察到当前的环境状态，否则智能体具有部分可观察性，需要相关函数对当前的观察解释为状态。如图 2-2（截取自维基百科）所示，强化学习智能体在离散的时间戳下与环境进行交互，在每个时间戳  $t$ ，智能体观察到现象  $o_t$ ，其中  $o_t$  通常包含了  $t-1$  时刻所做动作的奖赏  $r_t$ 。然后它从动作集里面根据当前策略选择一个动作  $a_t$ ，应用到当前环境中，因此环境进入了新的状态  $s_{t+1}$ ，并且智能体由状态转移概率函数  $P_a(s, s') = \Pr(s_{t+1} = s' | s_t = s, a_t = a)$  获得新的奖赏  $r_{t+1}$ 。强化学习的目的就是让智能体通过与环境不断的交互尽可能多地获得奖赏值来优化策略。智能体可以根据历史观察作为参数的函数来选择动作。

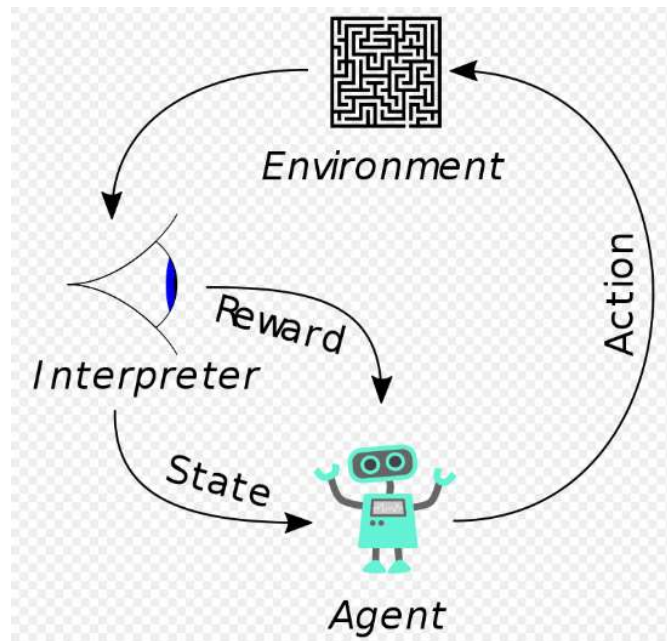


图 2-2 强化学习

Figure 2-2 Reinforcement Learning

在相同的状态下，当智能体完成动作所获得奖赏与另外一个智能体的获得的奖赏有差距时候，为了使动作更接近最优，智能体必须考虑该动作后多个动作序列的长期的收益，即最大化未来奖赏，即使当前的单步奖赏是负的。因此，通过智能体与环境不断地进行交互，逐渐完善智能体的动作策略，从而让智能体获得最优的决策能力。也正是因为智能体的奖赏是考虑长期收益，强化学习特别适合短期奖赏与长期奖赏进行折中考虑的问题。它已经成功地应用于解决不同的问题，涉及机器人控制，电梯调度，人机对话等领域。

强化学习智能体想要最大化当前动作奖赏需要考虑两个方面：一是需要知道动作集中每个动作带来的奖赏，二是实行奖赏最大的动作。若每个动作对应的奖赏都是一个确定的数值，那么采用遍历各个动作的方法，便可以得到最优的动作。但是，在实际情况中，当智能体在某个状态下，执行某个动作的奖赏值是符合某个概率分布的，因此，仅通过一次尝试就确定动作的奖赏期望值是不够确切的。在强化学习中通过“探索”，“利用”策略来选择合适的动作，“探索”即随机地从动作集合中选择动作，以获得动作的奖赏期望，而“利用”则是根据已经获得在各个动作的奖赏值，选择奖赏最大的动作。对于强化学习智能体来说，想要动作累积奖赏最大，则必须在探索与利用之间达到比较好的折中。

在多步强化学习任务，如果马尔可夫决策过程的四个要素：状态，奖赏函数，状态转移概率，动作，均已知，这样的情形称为“模型已知”，即机器已经对环境



建立了数学模型，从而能够在内部模拟出环境进行策略优化。智能体在这种模型已知的环境下进行的策略学习称为“有模型学习”，而在许多现实的强化学习任务中，环境的状态转移概率、奖赏函数并不知道，同时也无法知道环境中的状态情况。在这种不明确环境要素的情形下的学习，则称为“免模型学习”<sup>[38]</sup>。在免模型情形下，因为模型要素未知，智能体只能通过在环境中执行选择的动作，来观察转移的状态和得到的奖赏。一种直接的策略评估替代方法是让强化学习智能体与环境进行有限次数的交互采样，每一次采样都是智能体发出动作然后收到奖赏。基于采样轨迹得到动作的奖赏期望值。

## 2.2.2 神经网络

人工神经网络是受构成动物大脑的生物神经网络启发的计算系统。神经网络本身不是算法，而是许多不同机器学习算法的框架，它们协同工作并处理复杂的数据输入。这样的系统通过示例样本来“学习”近似某个函数  $f$ ，具体地是学习网络参数进而完成分类预测等任务，通常不用任何特定任务规则编程。例如，在图像识别中，他们可以通过分析已经手动标记为“猫”或“没有猫”的示例图像并使用结果来识别其他图像中的猫来学习识别包含猫的图像。神经网络在没有任何关于猫的先验知识的情况下进行学习，例如，猫有毛皮，尾巴，胡须和猫般的面孔。相反，神经网络会自动从它处理的学习样本中生成识别特征。

神经网络是基于称为人工神经元的连接单元或节点的集合进行构建，这些人工神经元模拟生物大脑中的神经元。每个连接，如生物大脑中的突触，可以将信号从一个人工神经元传递到另一个人工神经元。接收信号的人工神经元可以处理它，然后发信号通知与之相连的其他人工神经元。

如图 2-3 所示，在常见的神经网络实现中，人工神经元之间的连接处的信号是实数，且通常具有随着学习进行而调整的权重进行加权，每个人工神经元的输出通过其输入的非线性函数来计算。通常，人工神经元聚集成层。不同的层可以对其输入执行不同类型的转换。神经网络的第一层为输入层，训练数据就是通过该层输入到神经网络中去。神经网络的最后一层为输出层，以输出最后的结果。在输入层与输出层之间的层，学习算法必须决定如何使用这些层来产生想要的输出，但是训练数据并没有说每个单独的层应该做什么。相反，学习算法必须决定如何使用这些层来最好地实现  $f$  的近似。因为训练数据并没有给出这些层中每一层所需的输出，所以这些层被称为隐藏层。数据可能在多次遍历各层之后从第一层（输入层）传播到最后一层（输出层），并且通过反向传播算法来学习模型的参数。

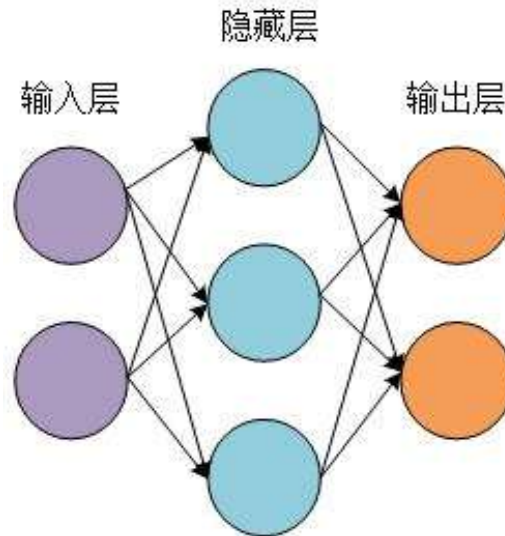


图 2-3 神经网络结构图

Figure 2-3 Neural Network Structure

神经网络模型根据任务的不同进化出不同结构的神经网络，如玻尔兹曼机，循环神经网络，卷积神经网络，自组织特征映射神经网络等。神经网络已经用于各种任务，包括计算机视觉，语音识别，机器翻译，社交网络过滤，游戏板和视频游戏以及医学诊断。

### 2.2.3 循环神经网络

循环神经网络是人工神经网络的一种，其中神经单元之间的连接沿时间序列形成有向图，这样就能允许它展示时间动态行为。与前馈神经网络不同，RNN 可以使用其内部状态（存储器）来处理输入序列，这使它能够在自然语言处理，语音识别等有连续时间特征的任务中有广泛的应用。

具体地如图 2-4（截取自维基百科）所示，序列信息  $x$  被保存在循环网络的隐藏层  $h$ ，并且  $h$  层保存了多个时间点的输入信息，以用于当前时间点的输入信息处理。循环神经网络能够发现由很多时刻分隔的事件之间的相关性，并且这些相关性被称为“长期依赖性”，因为时间下游的事件取决于之前发生的一个或多个事件，并且是其函数。正如人类记忆在身体内无形地循环，影响我们的行为而不能完全表示出来一样，历史输入信息在循环神经网络的隐藏层中循环。

在  $t$  时刻的隐藏层  $h$  的状态为：

$$h_t = \phi(Ux_t + Vh_{t-1}) \quad (2-1)$$

在  $t$  时刻的输入是  $x_t$ ，由权重矩阵  $U$  加权，然后与权重矩阵  $V$  加权的  $t-1$  时刻的隐藏层  $h_{t-1}$  的和得到  $t$  时刻的隐藏层状态  $h_t$ 。权重矩阵如同滤波器般分别决定了当前

输入与过去的隐藏层状态的重要性。通过反向传播算法来调节这些权重，减小神经网络的误差。加权的输入与隐藏层状态通过非线性激活函数，如 Sigmoid 或者 Tanh 函数计算得到隐藏层向量。进而由隐向量得到输出：

$$o_t = Wh_t \tag{2-2}$$

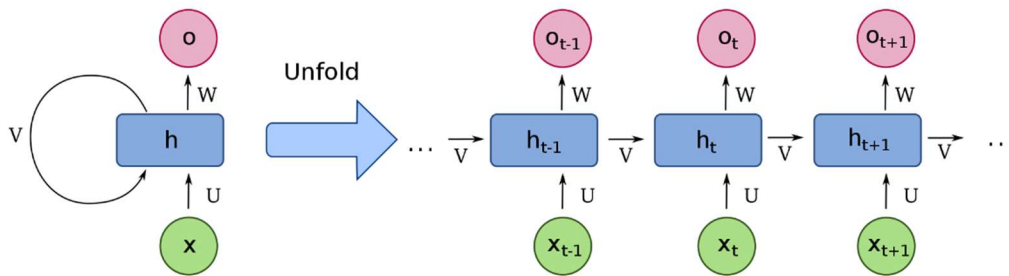


图 2-4 循环神经网络结构图

Figure 2-4 Recurrent Neural Network Structure

给定一系列字母，循环网络将使用第一个字符来帮助确定其对第二个字符的预测，比如初始 q 可能会导致它推断下一个字母将是 u，而初始 t 可能会导致它推断下一个字母将是 h。

长短期记忆网络（LSTM）是一种被广泛地用于深度学习的循环神经网络，一个标准的 LSTM 神经网络单元是由一个细胞，输入门，输出门和一个遗忘门构成。细胞记录了任意时刻的信息，三个门控制了流入和流出细胞的信息。具体地如图 2-5（截取自维基百科）所示，其中  $c_t$  表示 t 时刻的细胞状态， $h_t$  表示 t 时刻的隐藏层， $x_t$  表示 t 时刻的输入。

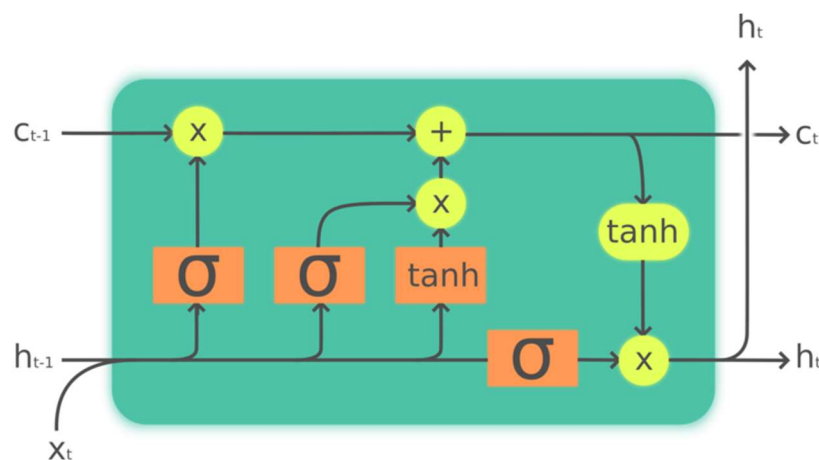


图 2-5 LSTM 神经网络结构图

Figure 2-5 LSTM Neural Network Structure

遗忘门:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2-3)$$

输入门:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2-4)$$

输出门:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (2-5)$$

其中函数 $\sigma$ 均是 Sigmoid 函数,  $h_{t-1}$ 为用于输出的隐藏层向量。对于细胞状态的更新:

$$c_t = f_t \odot c_{t-1} + i_t \odot \text{Tanh}(W_c x_t + U_c h_{t-1} + b_c) \quad (2-6)$$

其中,  $\odot$ 为矩阵哈达马乘积, 公式表示, 当前细胞状态向量是由前一时刻的细胞状态通过遗忘门遗忘一部分后, 加上经过输入门控制的当前输入的新信息来进行更新的。神经网络隐藏层 $h_t$ 通过输出门和当前的细胞状态进行更新:

$$h_t = o_t \odot \text{Tanh}(c_t) \quad (2-7)$$

同样地, LSTM 神经网络所涉及的变量  $W$ ,  $U$ ,  $b$  等均是采用反向传播算法进行调优。

LSTM 神经网络非常适合基于时间序列数据进行分类, 处理和预测。因为在时间序列中的事件之间存在长期依赖, 而依赖事件之间可能存在未知持续时间的滞后, 而 RNN 在训练时候就会因此出现梯度消失问题, 而 LSTM 细胞与门结构可以有效地解决在训练传统 RNN 时可能遇到的爆炸和消失的梯度问题, 从而更好地解决长期依赖问题。基于 LSTM 神经网络的变形网络 GRU<sup>[39]</sup>更简洁且有效地提升了模型的性能。

## 2.2.4 深度强化学习

在强化学习中, 智能体在某个状态  $s$  下做出动作  $a$  来获得奖赏  $r$ 。深度强化学习将神经网络与强化学习结合起来, 神经网络是将状态-动作对映射到奖励的智能体。与所有神经网络一样, 它们使用参数来近似将输入与输出相关联的函数, 并且它们通过梯度下降的方法调整那些权重来找到正确的参数或权重。

在强化学习中, 卷积神经网络<sup>[40]</sup>可用于识别智能体的状态, 例如在游戏中马里奥所在的屏幕的位置, 或无人机前的地形。也就是说, 他们执行他们典型的图像识别任务。但是卷积网络在强化学习中比在监督学习中对图像有不同的解释。在监督学习中, 神经网络将标签应用于图像, 也就是说, 它将名称与像素匹配。实际上, 它会根据概率对最适合图像的标签进行排名, 选出对应的标签。在强化学习中, 给

定表示状态的图像，卷积神经网络可以对可能在该状态下执行的动作进行排序，以选出奖赏最大的动作；例如，图 2-6，马里奥游戏中，智能体可能预测马里奥向右跑将得到 5 分，跳跃得 7 分，向左跑没有得分。因此，为了最大的得分，智能体会做出跳跃的决策。

## 卷积智能体

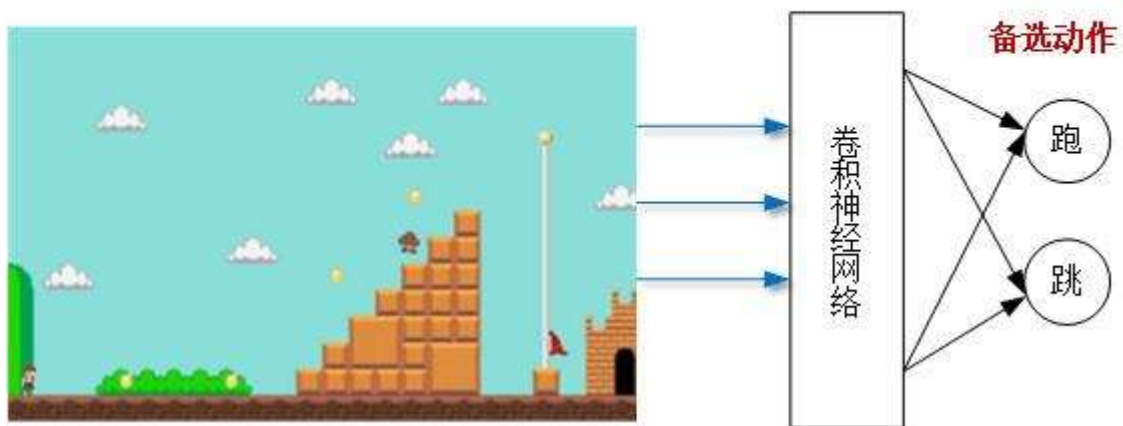


图 2-6 卷积智能体的深度强化学习

Figure 2-6 Deep Reinforcement Learning of Convolutional Agent

说明了一个卷积神经网络智能体的决策作用，它是将提取图像状态并将其映射到最优的动作。同样也可以采用其他的神经网络作为智能体进行决策，如在文献<sup>[37]</sup>中采用 GRU 神经网络作为智能体，它能够有效地提取时间序列特征进行决策，给学生推荐合适的单词复习以加速学生的学习过程。

在强化学习开始时，可以随机地初始化神经网络参数。通过智能体与环境不断地进行交互，使用来自环境的反馈，神经网络可以使用其期望奖励和实际奖励之间的差异来调整其权重并改进其对状态-动作对的合理性，以选择在不同状态下的最优动作。深度强化学习巧妙地结合了深度学习的优点，即具有较强的感知能力，可以对状态进行高维度向量的描述，和强化学习的优势，即其具有决策能力。将两者结合起来，实现了优势互补，为复杂系统的感知决策问题提供了解决思路，并且在自然语言处理，机器人控制，游戏 AI，自动驾驶与智能医疗等领域进行了应用。

### 2.2.5 交叉熵损失函数

神经网络中分类任务常使用交叉熵损失函数作为模型的优化目标。交叉熵常用于衡量不同分布之间的差异。假设样本类别有两个概率分布  $p, q$ ，其中  $p$  为样本类别的真实分布， $q$  为模型所预测出来样本类别的非真实分布。若按照真实分布  $p$  来衡量识别一个样本所需要的编码长度的期望为：

$$H(p) = \sum_i p(i) \log\left(\frac{1}{p(i)}\right) \quad (2-8)$$

而按照模型所预测的样本类别分布  $q$  来表示来自真实分布  $p$  的平均编码长度，则应该是：

$$H(p, q) = \sum_i p(i) \log\left(\frac{1}{q(i)}\right) \quad (2-9)$$

$H(p, q)$  称为交叉熵，衡量了  $q$  与  $p$  分布的差距，也就是衡量了机器学习算法模型在样本的分类的表现。特别地，针对二分类问题，单个样本交叉熵损失函数为：

$$\text{Loss} = -[y \log y' + (1 - y) \log(1 - y')] \quad (2-10)$$

其中  $y$  为样本的真实类别， $y'$  为模型的预测概率。

机器学习算法模型的目标就是让这两个分布的差距尽可能的小，让模型预测的分类标签尽可能与真实标签一致。

## 2.2.6 策略梯度算法

基于策略梯度的强化学习算法，是对强化学习策略进行优化的方法，目的是让智能体在不同的状态下选择最优的动作，合理地做出决策。基于策略梯度的强化学习算法将直接基于长期奖赏期望进行优化策略优化，采用的是机器学习领域经常使用的基于梯度的优化方法。策略梯度优化算法一直受到广泛的关注，作为一种经典的优化算法，策略梯度算法在智能体学习策略过程中有着相对稳定的表现，同时也可以连续和离散的行动空间进行计算来优化策略。

具体地，强化学习的目标是最大化长期奖赏期望，于是目标可以表示为：

$$\pi^* = \operatorname{argmax}_{\pi} E_{\tau \sim \pi(\tau)} [r(\tau)] \quad (2-11)$$

其中， $\tau$  表示智能体使用策略与环境进行交互得到的一条轨迹， $r(\tau)$  表示这条轨迹的总体回报，强化学习的目标是最大化这个值函数。我们将策略的值函数表示为：

$$J(\theta) = E_{\tau \sim \pi(\tau)} [r(\tau)] \quad (2-12)$$

采用的方法是对奖赏值函数进行求导，求出值函数关于策略参数的梯度  $\nabla_{\theta} J(\theta)$ ，并使策略参数沿着梯度上升的方向更新，从而使值函数增大，策略的奖赏期望增大，也就可以提升策略了。所以总结起来策略梯度分为两步：

- (1) 计算  $\nabla_{\theta} J(\theta)$
- (2)  $\theta' = \theta + \alpha \nabla_{\theta} J(\theta)$

基于策略梯度的算法有很多改进扩展的算法，其中包括了 Actor Critic 算法<sup>[41]</sup>；使策略单调提升的算法，其中包括置信域策略优化（TRPO）<sup>[42]</sup>和近端策略优化

(PPO) 等。

## 2.2.7 特征表达

### (1) 独热编码

独热编码是针对离散特征进行编码的方法，很多离散特征的取值并没有数值大小的意义，比如“红色”，“蓝色”等类别特征，为了合理地表示这些特征，并将其送入机器学习模型进行训练，需要将这些类别特征进行编码，而独热编码就是针对这种类别特征常用的机器学习数据预处理方法之一。独热编码的编码方式是根据类别值的个数  $N$ ，建立  $N$  维的 0 向量，向量的每一位表示一个类别特征值，针对某一类别特征值，在其对应的  $N$  维的 0 向量中的位置 1。比如颜色有“红”，“黄”，“蓝”三种，那么，红色独热编码表示为“001”，黄色表示为“010”，蓝色表示为“100”。

我们的类别特征，包括关卡特征，题目难度特征，均采用独热编码的方式，对特征值进行编码。

### (2) 特征组合

特征组合将两个或者更多的特征编码成一个的特征，以表示这些特征同时出现的方法，是对特征空间中的非线性规律进行编码的合成特征，模型通过组合特征获得的预测能力将会很大程度上超过任一特征单独的预测。多个数字特征和类别特征都可以通过组合特征进行表示，对于类别特征，我们使用独热编码对其进行编码。在神经网络中，进行特征组合，引入了非线性，并且降低了权重不平衡造成的不良影响，从而能够提升模型的性能。下面的等式表示了如何进行两个特征的组合特征。

举个例子，假设对纬度和经度划分区间进行分箱表示，获得单独的的长度为 5 独热编码特征矢量。例如，指定的纬度和经度可以表示如下：

$$\text{binned\_latitude} = [0, 0, 0, 1, 0]$$

$$\text{binned\_longitude} = [0, 1, 0, 0, 0]$$

对这两个特征矢量进行特征组合，即  $\text{binned\_latitude} * \text{binned\_longitude}$ ，那么该特征为是一个 25 元素独热矢量，其中包含 24 个 0 与 1 个 1。该组合中的单个 1 表示纬度与经度的特定连接。

因此，学生的做题结果，是由（题目，是否正确）组成，“题目”与“是否正确”都是类别特征，其中作答结果可以表示成长度为 2 的独热编码，而题目的独热编码长度根据题目的数量  $n$  确定。进行特征组合后，组合特征长度为  $2*n$ ，包括  $2*n-1$  个 0 与 1 个 1，具体表示某道题目是否正确作答。因此，我们通过他们的组

合特征来表示这一做题结果这一特征。

### (3) 连续特征离散化

某些特征具有连续的特征数值，我们称之为连续特征。特征的连续值在不同的数值区间的重要性是不一样的，所以希望连续特征在不同的区间有不同的权重，实现的方法就是对特征进行划分区间，每个区间为一个新的特征。常用做法，就是先对特征进行排序，然后再按照等频离散化为  $N$  个区间，每个区间为一个类别特征。

我们的习题数据中的时间特征为连续数值特征，采用这种方法对该连续特征进行离散化。

### (4) 离散特征 Embedding

当离散特征值较多时候，即独热编码的向量维度过高，数据便会变得过于稀疏，神经网络的训练效果就会下降，因此，需要采取方法将解决这个问题。

Embedding 旨在将网络中的节点表示成低维、实值、稠密的向量形式，使得得到的向量形式可以在向量空间中具有表示以及推理的能力，同时可轻松方便的作为机器学习模型的输入，而对于我们的类别特征，也可以采用这种方法避免独热编码维度过高的问题。

首先，针对某一特征，建立这个特征的 Embedding 矩阵，矩阵的每一列向量表示了该特征的一个特征值，且矩阵的第二维度与该特征的特征值的数目相同。因此，通过某一特征的独热编码与该 Embedding 矩阵的转置做点积，便可得到该特征值所对应的特征向量，避免了独热编码维度过高，从而解决数据稀疏问题。其中 Embedding 矩阵作为变量在神经网络训练中通过反向传播算法不断更新以更好地表示不同类别特征值。

我们的离散化后的特征，关卡，题目难度，离散化的时间，均采用离散特征 Embedding 的方法，由独热编码转换为向量表示。

## 2.3 模型评估

### 2.3.1 评估方法

对于机器学习模型，我们将分类错误的样本的数量占样本总数的比例称为“错误率”，更一般地，我们将机器学习算法模型的实际预测输出与样本的真实输出的差异称为“误差”。对于训练模型来说，采用的数据集分为“训练集”，“验证集”，“测试集”。“训练集”是用于机器学习算法模型拟合的样本，模型通过损失函数，采用梯度下降等优化方法调节模型的参数。“验证集”则是用于调节模型的超参数，并初步评估超参数设置对于模型的影响，常从训练集中划分。“测试集”不参与模



型的训练，只用于评估模型的泛化能力，即模型对于新样本的预测准确性。模型在训练数据集上的误差称为“经验误差”，在测试数据集上的误差称为“泛化误差”。一个性能优秀的模型，应该是有着比较小的“泛化误差”。如果模型在训练数据集上表现出良好的性能，但是在新的样本上泛化误差很大，则出现了过拟合现象，这说明模型性能并不理想。

交叉验证是一种常用的模型评估方法，它先将数据集分成  $k$  个大小相似的互斥子集，即  $D = D_1 \cup D_2 \cup \dots \cup D_k$ ,  $D_i \cap D_j = \emptyset (i \neq j)$ ，每个子集的数据分布相似。每次用  $k-1$  个子集的并集作为训练集，剩下的一个子集作为测试集。采用这种划分方法将会获得  $k$  组的训练集与测试集组合。然后就进行  $k$  轮的训练与测试，最终返回这  $k$  轮的测试结果，将这  $k$  轮的测试结果求取平均值，得到模型的性能表现。这种评估模型的方法称为“ $K$  折交叉验证”，常见的  $k$  的取值有 5,10 等。

### 2.3.2 性能度量

对机器学习模型的泛化性能评估不仅需要有效科学的方法，还要具体的数值对模型的泛化能力进行评价。在预测任务中，给定样本数据集  $S = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)\}$ ，其中  $y_i$  是样本  $x_i$  的真实标记。要想评估模型  $f$  的性能，就需要将模型的预测结果  $f(x)$  与真实标记  $y$  进行对比。

机器学习模型的任务主要分为回归与分类，针对回归任务，常用的度量方法是“均方误差”：

$$E(f; S) = \frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i)^2 \quad (2-13)$$

对于分类任务而言，以常见的二分类任务为例，可将标记好的样本根据其真实类别与模型预测类别的组合分为真正例（TP），假正例（FP），真反例（TN），假反例（FN）。其中真正例表示样本真实标签是正例且模型也预测为正例的样本，假正例表示样本真实标签是反例但是模型预测为正例的样本，真反例为样本真实标签是反例且模型预测也为反例的样本，假反例为样本真实标签是正例但是模型预测为反例的样本。通过混淆矩阵可以明确地表示，如表 x 所示。

表 2-1 混淆矩阵

Table 2-1 Confusion Matrix

真实情况	预测结果	
	正例	反例
正例	真正例（TP）	假反例（FN）
反例	假正例（FP）	真反例（TN）

我们定义精确率为 P:

$$P = \frac{TP}{TP+FP} \quad (2-14)$$

定义召回率为 R:

$$R = \frac{TP}{TP+FN} \quad (2-15)$$

精确率与召回率为矛盾的度量, 精确率高则召回率低, 否则, 精确率低则召回率高。在一些应用中, 我们对精确率与召回率的要求不同, 比如在推荐系统中, 为了减少无用的推荐对用户造成的负面影响, 我们希望推荐系统的精确率高, 就是希望系统所推荐的用户尽可能都点击, 对于重要机械零件的筛选分类中, 我们希望提升出货的质量, 将尽可能地将所有的坏零件回收, 这就要求模型的召回率高。为了表达我们对精确率与召回率的不同重视程度, 设置了 F1 值作为模型性能度量的一个标准:

$$F1 = \frac{(1+\beta^2)*P*R}{(\beta^2*P)+R} \quad (2-16)$$

其中  $\beta > 0$  表示了模型对于精确率与召回率的权衡。  $\beta > 1$  模型更加重视召回率,  $\beta < 1$  模型更加重视精确率。

很多机器学习模型为样本输出一个概率值, 如神经网络, 逻辑斯特回归等模型, 然后设置阈值, 若输出的概率值大于阈值, 那么分为正类, 否则分为反类。常见的是设置阈值为 0.5, 输出概率大于 0.5, 分为正类, 否则为反类。这个阈值的设定直接决定了模型的泛化能力。因此, 我们根据模型输出的预测概率结果对各个样本进行排序, 概率越大, 排名越靠前, 然后设置一个节点, 在排名在该节点之前的为正例, 之后的为反例。在实际的任务中, 我们根据任务对分类精确率与召回率的需求不同, 设置不同的节点位置, 节点越靠前, 则模型更加重视精确率, 节点越靠后, 模型更加重视召回率。模型的这种排名能力, 体现了模型在不同任务下的泛化能力的好坏。ROC 曲线就是采用这种原理来度量模型的性能的。ROC 全称为“受试者工作特性曲线”, 其横轴是“真正例率”(TPR):

$$TPR = \frac{TP}{TP+FN} \quad (2-17)$$

纵轴是“假正例率”(FPR):

$$FPR = \frac{FP}{TN+FP} \quad (2-18)$$

如图 2-7 所示, 对根据模型输出概率排序的样本来说, 将节点设置到最大模型的分裂结果对应坐标(0,0), 表示模型将所有的样本都预测为反例。三种不同颜色表示了三个不同模型的 ROC 曲线, 理想的模型的 ROC 能够达到 TPR=1 而 FPR=0, 但是这种理想模型很难达到, 为了比较不同模型的性能, 我们采用 ROC 曲线下面的面积 AUC 来判断不同模型的优劣。AUC 越大, 则模型的性能越好, 反之越差。

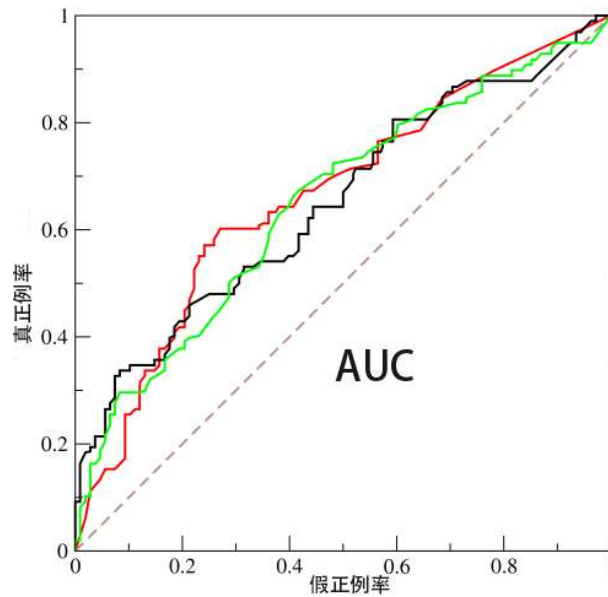


图 2-7 AUC 与 ROC 曲线

Figure 2-7 AUC and ROC Curve

## 2.4 开发平台

本节主要介绍本论文实验中所用的开发平台,包括机器学习算法库 Scikit-learn,集成多个科学计算包的 python 发行版本 Anaconda,强化学习算法库 Rllab,机器学习框架 TensorFlow。

### 2.4.1 Scikit-learn 算法库

Scikit-learn 是一个用于 Python 编程语言的免费机器学习库,支持 Linux, macOS, Windows 等操作系统。它具有各种用于分类,回归和聚类的机器学习算法,包括支持向量机,随机森林,k 均值聚类和 DBSCAN 聚类等,同时对于最新的机器学习算法也有对应的更新,如 Xgboost 算法等。已经训练好的模型,可以进行保存,方便下次的调用。其算法封装的函数使用简单,能够方便开发人员快速地开发应用。

在数据预处理方面,Scikit-learn 同时也提供了方便的数据的预处理函数,如划分训练与测试数据集,数据降维,数据归一化处理,特征提取与转换等,开发者可以根据自身的数据情况选择合适的函数进行数据预处理。在模型评估方面,它具体实现了交叉验证,F1 值,ROC 曲线,模型 AUC 值等常见的模型评估方法与度量方法,以便开发者对于模型进行评估改进。在本论文的研究中,模型的评估与数据预处理均调用 Scikit-learn 库来实现,且它可以与 Python 的数值科学库 NumPy 和 SciPy 联合使用。

## 2.4.2 Anaconda 集成环境

Anaconda 是用于科学计算（数据科学，机器学习应用程序，大规模数据处理，预测分析等）的 Python 和 R 编程语言的免费开源发行版，旨在简化科学计算包的管理和部署。包由包管理系统 conda 管理，包括包的安装，卸载，升级等。Anaconda 发行版包含了 1400 多个适用于 Windows, Linux 和 MacOS 的流行数据科学包。

同时 Anaconda 也包含了 Jupyter notebook 和 spyder。Jupyter notebook 是基于 web 的交互式计算环境，可以编辑易于人们阅读的文档，用于展示数据分析的过程。spyder 是一个使用 Python 语言、跨平台的、科学运算集成开发环境。在本论文研究中，系统的开发是在 Anaconda 集成环境下进行的，所涉及的 pandas, numpy 等科学计算包均有 conda 进行管理。

## 2.4.3 Rllab 库

Rllab 是一个用于开发和评估强化学习算法的框架，里面集成了强化学习的基本要素的定义，包括了环境，奖赏，动作等，也实现了基本的和前沿的强化学习算法。因此，基于它我们可以更好地进行强化学习的研究，我们可以根据自己的问题自定义强化学习中的各个要素，采用合适的强化学习算法解决自己的问题。

OpenAI Gym 是一个开发和比较强化学习算法的工具包，其中包括了很多环境要素和强化学习中的在线计分板，Rllab 实现了强化学习算法的实现，这些实现与环境策略的布局无关，Rllab 和 Gym 完全兼容，开发者可以采用 Rllab 的强化学习算法在 Gym 的环境中方便地进行开发。本论文的题目推荐策略所采用的强化学习算法信赖域策略优化（TRPO）就是基于 Rllab 实现的。

## 2.4.4 TensorFlow 框架

TensorFlow 是一个采用数据流图，用于数值计算的开源软件库，被广泛应用于各类机器学习算法的编程实现。数据流图用“结点”和“线”的有向图来描述数学计算，节点在图中表示数学操作，但也可以表示数据输入的起点/输出的终点，或者是读取/写入持久变量的终点。图中的线则表示在节点间相互联系的多维数据数组，即张量（tensor），代表着“节点”之间的输入/输出关系。一旦输入端的所有张量准备好，节点将被分配到各种计算设备完成异步并行地执行运算。它架构灵活且可以在多种平台上展开计算，例如台式计算机中的一个或多个 CPU（或 GPU），服务器，移动设备等等，并且具有真正的可移植性，可以在不修改代码的情况下，将

模型移植到云端或者在多个 CPU 上规模化运算。TensorFlow 支持多种语言编程，如 C++，Python 等。TensorFlow 最初由 Google 大脑小组（隶属于 Google 机器智能研究机构）的研究员和工程师们开发出来，用于机器学习和神经网络方面的研究，但这个系统的通用性使其也可广泛用于其他计算领域。<sup>[43]</sup>本论文研究中的知识追踪模型便是基于 TensorFlow 框架进行搭建和训练的。

## 2.7 本章小结

本章主要介绍了在本论文研究中的技术背景，主要包括采用的数据集，以及开发所采用的算法工具包以及模型的开发平台。具体包括：

- (1) IPS 智能练习系统的学生习题作答数据集。
- (2) 研究中所采用的机器学习算法原理与模型的评估、度量方法。
- (3) 模型的开发平台与使用的算法工具包。

## 3 深度知识追踪模型

本章介绍我们提出的深度知识追踪模型。我们首先介绍了我们提出的深度知识追踪模型的基本思路，然后对模型所采用的题目特征、学生行为特征与学生作答结果进行了相关性分析，然后基于这些特征进行模型的改进，将题目概念的等级结构引入神经网络的设计，设计了新的深度学习知识追踪模型，最后设计与 Baseline 模型的对照实验，评估了我们改进的知识追踪模型的性能。

### 3.1 基本思路

为了解决知识追踪模型预测学生作答结果准确率不高的问题，我们基于 DKVMN 网络<sup>[20]</sup>，修改了模型的知识概念矩阵、题目与各个知识概念相关性权重的计算方法以合理地利用题目的知识概念特征，然后利用知识概念矩阵与题目难度、关卡特征对作答结果进行预测。模型的知识概念矩阵会根据学生的实际作答结果进行更新以追踪学生知识状态，进而可以对新的题目进行作答结果预测。在对模型的知识概念矩阵更新方面，我们提出采用习题作答结果与习题作答时间特征联合更新知识概念矩阵的方法，以获得学生在各个知识概念掌握水平上更加准确的表征，从而提升预测学生作答结果的准确率。

### 3.2 特征统计与表达

在本节中，我们首先介绍了我们的数据集，然后基于该数据集观察并分析题目特征，学生行为特征与学生作答结果的相关性，并采用科学的方法对特征进行表达，使特征能够适用于深度知识追踪模型中。

#### 3.2.1 数据集介绍

我们采用的数据集为第二章介绍的 IPS 智能练习系统的五年级数学习题答题数据集，包括 24806 名学生 390340 条做题记录。每一条习题作答记录包括做题时间，题目 ID，题目知识概念，作答结果（正确与否），题目难度，关卡类型等，每道题目都有三级知识概念结构，所有的题目共涉及 13 个一级知识概念，47 个二级知识概念，123 个三级知识概念。

### 3.2.2 知识概念

我们针对现有数学题目所涉及的高频出现的前 100 个知识概念进行了分析，如图 3-1 所示：

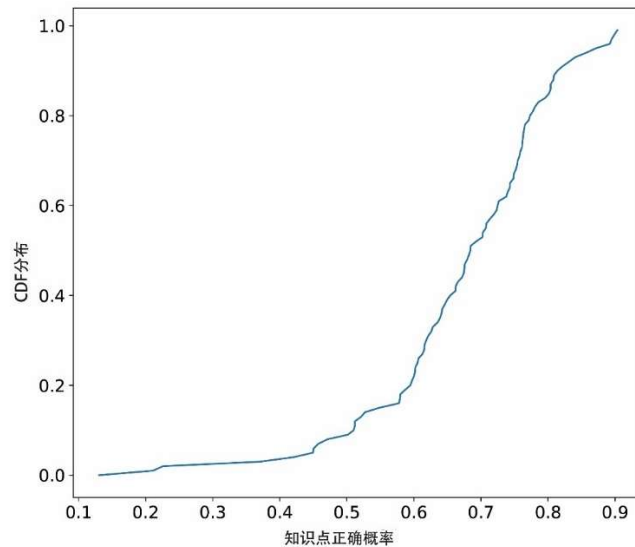


图 3-1 知识状态与作答结果相关性

Figure 3-1 Correlation between Knowledge Status and Answering Results

我们统计了 100 个知识概念题目正确率的累积分布函数 (CDF: Cumulative Distribution Function)，从图中可以看出，有近 80% 的知识概念正确率较为均匀地分布在 0.6 到 0.8 之间，在正确率为 0.8 到 0.9 之间的知识概念为相对简单的知识概念，意味着大部分人都能正确作答该知识概念所涉及的题目，占约 10%。而正确率小于 0.5 的知识概念为相对难的知识概念，也占了近 10%。因此，不同的知识概念表现出不同的难度特征，这是由知识概念本身之间有着较大的差异造成的。学生的知识状态就是由各个不同的知识概念的掌握状态决定。学生在涉及某个或者某几个知识概念的题目的做题情况，能反映学生在这些知识概念的掌握情况，也同样是根据这些题目的作答结果来更新学生对各个知识概念的掌握状态，进而预测学生正确作答某一道题目的概率。因此根据题目涉及的知识概念对准确预测学生作答结果与学生的知识状态更新十分重要。

题目的多级概念特征通过互不相同的数字编号 (1,2,3...) 进行特征表达，送入知识追踪模型从而对特征 Embedding 矩阵进行过滤以计算题目的知识概念权重向量。

### 3.2.3 题目关卡

我们数据所涉及的题目关卡包括预习，课前测，课后测，作业，复习巩固，阶段回顾等 6 个关卡，因为每个关卡的题目设置目的不同，学生在每个关卡的学习阶段与知识掌握状态不同，学生在各个关卡的题目正确率也有差异。

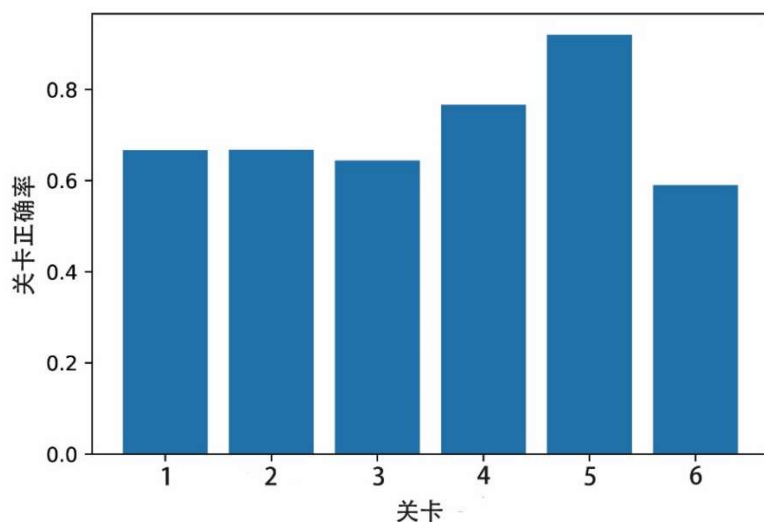


图 3-2 各个关卡正确率

Figure 3-2 Correct Rate of Each Gate

如图 3-2 所示，我们统计了 6 个不同关卡 390340 题次的学生作答结果情况，其中 1, 2, 3 关卡的正确率均在 0.65 左右，在作业阶段的关卡 4，正确率提升很大，在 0.77，这说明，学生经过前面的课堂学习过程，知识状态提升，题目作答情况较好，复习巩固在 0.9 左右，进一步说明前面的学习效果。阶段回顾所涉及的知识概念较多，很多知识概念是比较长时间以前学生所学的内容，在这个关卡题目正确率大幅下降，在 0.59，可能是学生在经过一段时间后知识状态由于记忆衰减，减弱，没有及时复习等原因，导致题目错误率上升。

不同的关卡的题目设置都有其考察的目的与难度。因此，我们可以看出，学生所做题目的关卡特征，能进一步表示由于时间，学习阶段等因素导致的学生的知识状态的变化，这些改变是无法直接通过题目信息来进行表达，而关卡却可以记录这种变化，从而对做题结果进行更好地预测，对知识状态进行更准确地更新。

题目的关卡特征属于类别特征，我们采用第二章 2.2.7 节所提到的独热编码与离散特征 Embedding 的方法对题目的关卡特征进行编码，与其他特征向量拼接以对习题作答结果进行预测。

### 3.2.4 做题时间



我们统计了每道题目所涉及的所有作答记录中作答时间的中位数，如果一个学生完成某道题目的时间超过中位数，我们将其置 1，否则置-1。因此，一个学生的做题序列对应着一条由 1 和-1 组成的时间表示序列。我们计算每一个学生的作答结果与该学生题目序列的时间表示序列的皮尔森相关系数。这里我们选择了做题路径最长的 2000 名学生的做题路径进行了分析，并绘制了相关系数的 CDF 图。

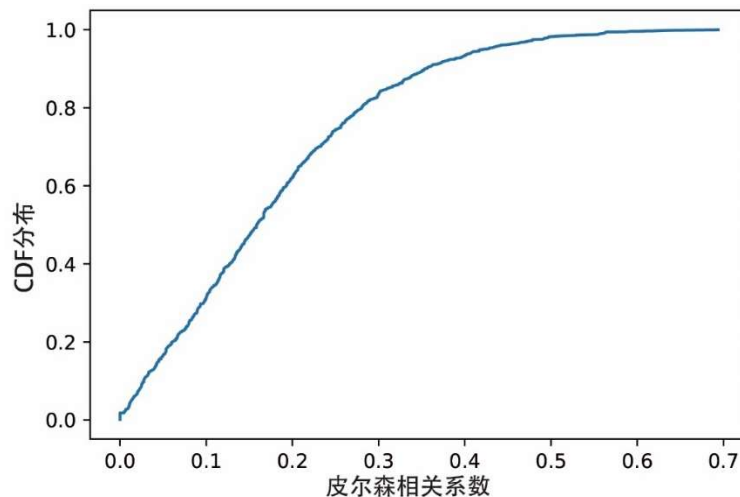


图 3-3 皮尔森相关系数累积概率分布

Figure 3-3 Cumulative Distribution Function of Pearson Correlation Coefficient

从图 3-3 可以看出，大部分学生（接近 90%）的作答结果与做题时间有比较小的相关性，相关性小于 0.4，且分布相对均匀。很少的一部分学生的作答结果与时间具有很强的相关性。因此，做题时间对最后学生的做题结果影响不大，做题时间不参与作答结果的直接预测中去。

习题作答时间为连续数值特征，我们采用第二章 2.2.7 节提到的连续特征离散化的方法，对时间特征离散化，将做题时间分布均匀划分为 10 个区间，进而得到 10 个时间间隔，将每个时间间隔作为一个类别特征。我们根据学生的做题时间确定该时间位于哪个时间间隔，进而得到时间的类别特征，这就是将时间特征离散化。然后对每个离散化的时间特征通过独热编码与离散特征 Embedding 的方法，对其编码以加入到模型知识概念矩阵更新中。

### 3.2.5 题目难度

针对题目难度特征，我们发现不同的难度的题目其正确率分布在不同的关卡下是不同的。

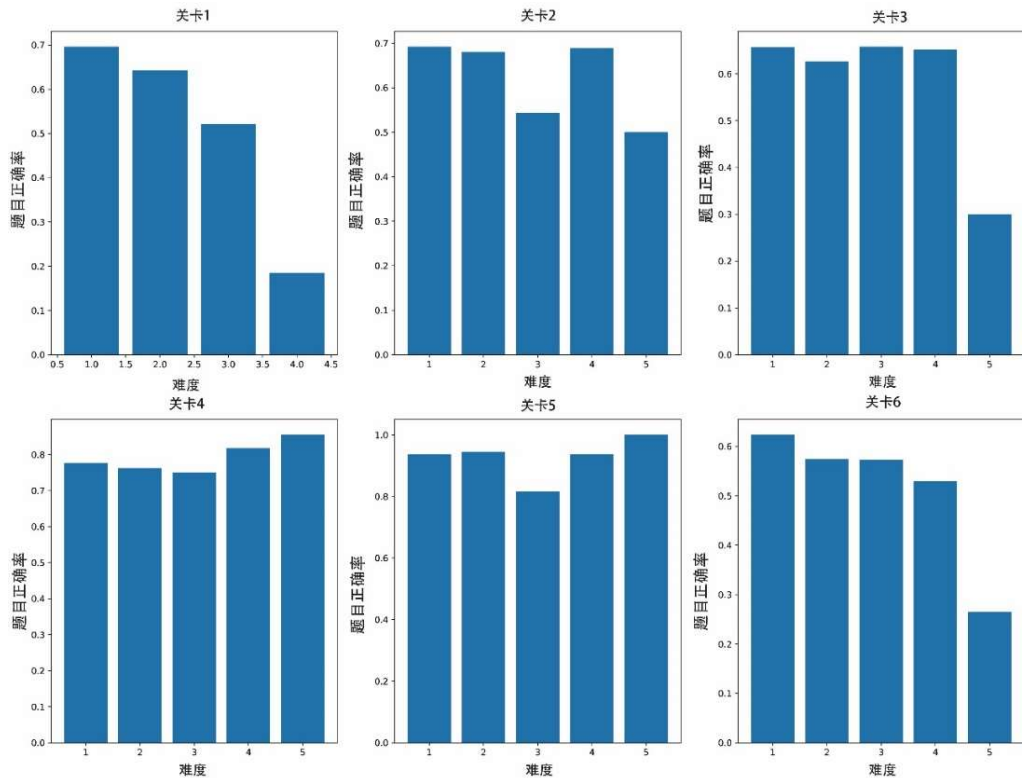


图 3-4 关卡-难度题目正确率

Figure 3-4 Gate-Difficulty Correct Rates of Exercises

图 3-4 是 6 个关卡 5 个不同难度题目的作答情况。从图中可以看出，由于不同关卡下，专家出题的目的不同，不同难度题目侧重考察方向不同，学生的知识状态不同，因此，不同关卡下，不同难度题目的正确概率有着明显的差异。比如难度为 1,2 的题目正确率在前 5 个关卡始终保持较高的水平，前且随着学生学习过程的进行，3 难度的题目的在前 5 个关卡中的正确率逐渐增加，这个符合学生学习知识状态提升的过程。同时，在关卡 6，阶段复习关卡，各个难度的题目正确率均下降，且越难的题目下降越多，5 难度的题目出现错误记录大于正确记录的现象。

不同难度题目的正确概率分布模式方面，关卡 4 和关卡 5 更加接近，由于关卡 1 没有难度为 5 的题目，关卡 1 和关卡 6 在难度为 1, 2, 3, 4 的题目更加接近。

因此，难度特征结合关卡特征，是预测学生做题结果的重要参考。题目的难度特征属于类别特征，我们采用第二章 2.2.7 节所提到的独热编码与离散特征 Embedding 的方法对题目的难度特征进行编码，与其他的特征向量拼接以对习题作答结果进行预测。

### 3.3 模型构建

这一节我们将详细介绍我们的深度知识追踪模型，包括知识概念权重的计算方法，模型的预测，知识概念矩阵的更新。

### 3.3.1 知识概念记忆矩阵

我们基于动态键值记忆网络去设计了知识概念记忆矩阵。知识概念记忆矩阵是用于表示学生各个概念的掌握状态。矩阵的维度是根据真实的题目的概念列表进行定义，设记忆矩阵的维度为  $N$ ，则说明记忆矩阵涉及  $N$  个不同的知识概念的掌握状态，每个知识状态由  $M$  维的向量进行表示，即为图 3-5 中的矩阵  $M_t^v$ ，与其对应地有矩阵  $M_t^k$ ，为概念的 Embedding 矩阵，用以  $N$  个概念的  $K$  维向量表示。比如，在“数论”一级知识概念下有 7 个二级知识概念，15 个三级知识概念，则设置  $N=23$ 。而在动态键值记忆网络中， $N$  是模型可调参数，表示题目隐概念的个数。记忆矩阵是随着做题的进行而不断写入与擦除，以追踪学生实时的知识状态。我们的模型是基于动态键值记忆网络，结合实际的题目的多级知识概念进行改进的，我们称这种模型为概念敏感的动态键值记忆网络。模型的结构如图 3-5 所示。下面章节将对模型的细节进行介绍。

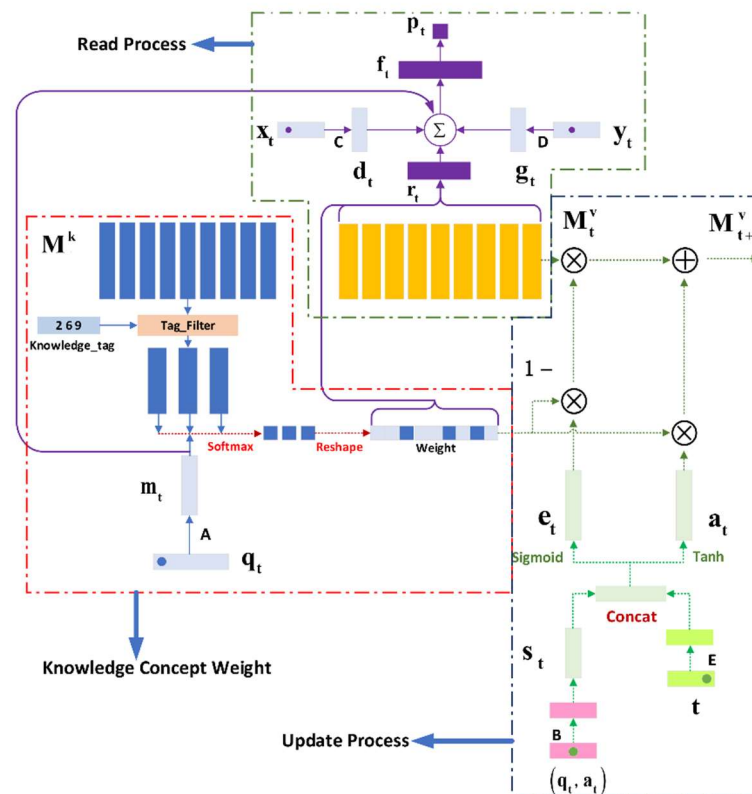


图 3-5 知识追踪模型

Figure 3-5 Knowledge Tracing Model

### 3.3.2 知识概念权重

我们首先计算题目的知识概念权重 (KCW: Knowledge Concept Weight), 知识概念权重是用于计算加权的各个知识概念的掌握程度, 进而对题目的作答结果进行预测。并且, 在学生完成了题目作答之后, 根据学生的实际作答结果, 利用知识概念权重去更新学生的知识概念记忆矩阵。

如图 3-5 所示, 首先, 习题的独热编码通过习题的 Embedding 矩阵, 获得其 Embedding 的向量表示。然后我们按照算法 1 的过程, 计算习题的知识概念权重。

---

#### Algorithm 1 Knowledge Concept Weight Calculation

---

**Require:**

$\mathbf{q}_t$ : embedding of the exercise arrived at time  $t$

$K_t$ : knowledge concept list of  $q_t$

$\mathbf{M}_t^k$ : the concept embedding matrix at time  $t$

**Ensure:**

*Weight*: Knowledge concept weight of the exercise arrived at time  $t$

```

/* Calculate KCW */
1:  $R \leftarrow []$ 
2: for each  $k \in K_t$  do
3:    $corr \leftarrow \mathbf{q}_t \cdot \mathbf{M}_t^k[k]^T$ 
4:    $R.append(corr)$ 
5: end for
6:  $R_s \leftarrow Softmax(R)$ 
   /* Reshape the weight vector to make its length equal to the number of concepts */
7:  $Weight \leftarrow [0, \dots, 0]$ 
8:  $i \leftarrow 0$ 
9: for  $i < 3$  do
10:   $Weight[K_t[i]] \leftarrow R_s[i]$ 
11:   $i \leftarrow i + 1$ 
12: end for
13: return Weight

```

如算法 1 所示, 在第一行, 我们先初始化一个权重列表  $R$ , 因为每个题目共涉及三级知识概念, 因此  $R$  的长度为 3。然后根据题目涉及的知识概念标号列表  $K_t$ , 从知识概念矩阵中提取对应的知识概念 Embedding。在第二行, 我们计算题目 Embedding 与各个提取出来的知识概念 Embedding 的点积, 并将结果保存在  $R$  列表。之后在第六行, 我们对  $R$  列表中的数值做 Softmax 计算, 其计算公式为:

$$Softmax(z_i) = e^{z_i} / \sum_{j=1}^N e^{z_j} \quad (3-1)$$

得到知识概念权重 KCW。然后, 我们初始化一个长度为总的题目概念数量的 0 向量  $Weight$  (7 行), 并对题目涉及的三级知识概念计算出的 KCW, 将其每个值赋值到  $Weight$  向量中该知识概念标号所对应的位置上去 (8 行)。

总的来说, 相比于动态键值记忆网络, 知识概念权重的计算方法合理地利用了题目知识概念的标签特征, 只计算题目与其三级知识概念的关系权重, 而不是和所有隐概念之间的关系权重, 题目和其他的概念之间的权重, 我们设置为 0。

### 3.3.3 作答结果预测

我们通过计算得到的题目知识概念权重  $KCW$ ，加权该题目所涉及的三个知识概念的知识向量，并将这些知识向量进行求和来表征学生在该题目涉及知识概念的总体掌握状态，进而根据这个总体掌握状态来预测学生在这个题目的作答结果。具体地，我们通过算法 1 得到的权重为  $w$ ，则学生在该题目涉及知识概念的总体掌握状态为：

$$r_t = \sum_{i=1}^N w(i) M_t^y \quad (3-2)$$

如图 3-5 所示，在预测阶段，相比于动态键值记忆网络，我们不仅考虑了学生的知识状态，还考虑了题目的难度，关卡等特征。具体地，对于当前的题目  $q_t$ ，我们通过题目 Embedding 矩阵获得题目的向量表示  $m_t$ ，类似的方法，我们获得题目难度向量  $d$  与关卡特征向量  $g$ ，然后将它们进行拼接，通过以  $\tanh$  为激活函数的全连接层得到一个总体向量：

$$f_t = \text{Tanh}(W_0^T [r_t, d_t, g_t, m_t]) \quad (3-3)$$

这个总体向量包含了学生知识状态与题目特征。其中， $\text{Tanh}(z_i) = (e^{z_i} - e^{-z_i}) / (e^{z_i} + e^{-z_i})$ 。

最后  $f_t$  通过一个以 Sigmoid 为激活函数的全连接层输出学生正确作答该题目的概率：

$$p = \text{Sigmoid}(W_1^T f_t) \quad (3-4)$$

其中， $\text{Sigmoid}(z_i) = 1 / (1 + e^{-z_i})$ 。

### 3.3.4 记忆矩阵更新

在我们观察到学生的做题结果后，我们通过题目计算得到的知识概念权重  $KCW$ ，去更新表示学生各个概念掌握程度的知识概念记忆矩阵  $M_t^y$ 。文献<sup>[1]</sup>提出学生的答题时间和其知识概念的掌握程度有关。因此，相比于动态键值记忆网络，我们在更新知识概念记忆矩阵的时候考虑了学生的做题时间，而前者忽略了。具体地，我们将连续数值的变量时间进行离散化后，通过离散特征 Embedding 的方法，将其转换为向量  $t$ 。作答结果（题目，作答正确与否）我们参考 DKVMN 网络 [20] 的方法通过第二章提到的组合特征进行编码并通过 Embedding 矩阵  $B$  表示为向量  $s_t$ ，然后将  $t$  与  $s_t$  进行拼接表示知识的更新向量，来更新知识概念记忆矩阵。

具体地，更新过程包括记忆擦除过程与记忆增添过程。我们分别通过擦除权重  $E$  与添加权重  $D$  来获得记忆擦除向量  $e$  与记忆增添向量  $a$ ：

$$e = \text{Sigmoid}(E^T [s_t, t]) \quad (3-5)$$

$$\mathbf{a} = \text{Tanh}(\mathbf{D}^T[\mathbf{s}_t, \mathbf{t}]) \quad (3-6)$$

之后我们便得到更新后的知识概念记忆矩阵 $\mathbf{M}_{t+1}^y$ :

$$\mathbf{M}_{t+1}^y(\mathbf{i}) = \mathbf{M}_t^y(\mathbf{i})[1 - w(\mathbf{i})\mathbf{e}][1 + w(\mathbf{i})\mathbf{a}] \quad (3-7)$$

整个知识追踪模型的变量, 如  $\mathbf{E}$ ,  $\mathbf{D}$ , Embedding 矩阵  $\mathbf{A}$ ,  $\mathbf{B}$ , 还有知识概念矩阵 $\mathbf{M}^k$ 等均是通过对模型训练获得。模型的训练是最小化预测结果  $\mathbf{p}$  与真实回答结果  $\mathbf{y}$  的交叉熵损失函数:

$$L = -\sum_t((y_t \log p_t) + (1 - y_t) \log(1 - p_t)) \quad (3-8)$$

模型采用动量梯度下降法来进行优化, 以高效地训练模型。

总之, 我们所设计的深度知识追踪模型基于实际应用, 通过测量观察, 分析出对于学生模型较为相关的模型特征, 考虑课程的概念列表, 题目与三级知识概念的映射关系, 并加入了题目特征, 学生的学习行为特征, 来提高知识追踪的表现。

### 3.4 性能评估

这一节主要通过实验对比分析了深度知识追踪模型的表现, 包括基于 LSTM 神经网络的知识追踪模型, 基于动态键值记忆网络 DKVMN 的知识追踪模型, 我们设计了基于多级知识概念的 DKVMN-CA 模型, 以证明多级知识概念特征的有效性, 且设计合理的模型结构才能充分应用多级知识概念特征, 以达到明显提升模型性能的目的。在 DKVMN-CA 的基础上, 我们设计了多组对比实验, 加入难度, 关卡, 做题时间等特征, 测试其他特征对于知识追踪模型性能的影响。

#### 3.4.1 数据预处理

我们进行实验的数据集采用学而思 IPS 系统的学生做题行为数据集。数据集中详细记录了学生完成习题的详细信息, 包括题目 ID, 题目作答正误情况 (1, 0), 完成题目的时间, 题目的多级知识概念, 题目关卡, 难度, 关卡类型等信息。我们过滤掉了数据字段缺失的数据并且采用了五年级数学的做题数据集作为实验数据集。

结合我们的具体应用场景, 为一级知识概念的专题复习, 因此, 我们对数据进行了筛选过滤, 提取每个同学在该一级知识概念下的做题路径, 这里我们选择的一级知识概念为“数论”。因此, 每个同学的做题路径均为“数论”一级知识概念下各个二级与三级知识概念所涉及的题目。为了保证每条习题路径尽可能长且有足够的习题路径数据, 我们参考文献<sup>[44]</sup>的方法, 对每条习题路径的长度进行了限制, 保证每条路径的长度都大于或者等于 5, 不满足长度限制的习题路径将被删掉。经

过数据预处理后，用于实验的数据为 7124 名学生 44128 条做题记录。

### 3.4.2 实验细节

模型经过训练，去学习概念矩阵 $M^k$ 与知识概念记忆矩阵 $M^v$ 的初始值， $M^k$ 的每个位置表示一个概念的 Embedding 向量，其在测试集上是不变的。与此同时， $M^v$ 的每个位置对应一个概念的初始状态，表示该概念的初始难度。

针对实验中的每一个模型，我们均实验 50 次，并且，每一次实验都是随机地将数据按照 7:3 的比例进行训练集与测试集的切分，训练集用于模型参数的训练与超参数调参，测试集测评估模型的性能。变量都是采用 0 均值，方差为 0.1 的高斯分布随机初始化。我们使用梯度裁剪与动量随机梯度下降来训练 DKT 模型，DKVMN 模型与我们的 DKVMN-CA 等模型<sup>[45]</sup>，我们将动量设置为 0.9，梯度裁剪阈值设置为 50。在 DKT 中加入多级知识概念特征，我们参考文献<sup>[13]</sup>的方法，在题目与作答结果组合特征 Embedding 后面拼接知识概念 Embedding 作为 LSTM 的输入。因为输入的序列的长度不相同，我们将所有的题目序列都用一个无效的标记填充到长度为 200。我们采用<sup>[15]</sup>的方法，以 AUC 作为评价模型性能的指标。我们评估了每个模型在其 50 次实验中，测试集的模型 AUC 的平均值，最大值，和方差。

表 3-1 不同模型评估

Table 3-1 Evaluation of Different Model

模型	AUC 平均值	AUC 最大值	AUC 方差
DKT	0.711	0.712	1.86e-05
DKVMN	0.712	0.720	2.05e-05
DKT-KC	0.703	0.715	2.53e-05
DKVMN-KC	0.714	0.724	1.85e-05
DKVMN-CA	0.724	0.731	2.14e-05
DKVMN-CA+Stage	<b>0.728</b>	0.736	<b>1.48e-05</b>
DKVMN-CA+Duration	0.725	0.737	1.75e-05
DKVMN-CA+Difficulty	0.726	0.736	2.44e-05
DKVMN-CA+Stage, Duration	0.726	<b>0.739</b>	2.43e-05

### 3.4.3 知识概念结构的增益

我们首先评估了模型的知识概念结构对于深度知识追踪模型的影响。在没有其他的特征，如做题时间，关卡等，加入的情况下，我们对比了 DKVMN 模型与 DKVMN-CA 模型的性能，其结果如表 3-1 所示，表中“DKVMN”与“DKVMN-CA”这两行分别表示这两个模型的性能，基于知识概念结构的 DKVMN-CA 的平均 AUC 值为 0.724，是明显高于目前最好模型 DKVMN 的 0.712，1.2%的 AUC 提升明显高于 DKVMN 模型相比于 DKT(AUC=0.711)的 0.01 的模型提升。这说明，我们的知识概念结构对于知识追踪来说很有效。

除了 DKVMN-CA 涉及知识概念的特征，还有比较常见的用知识概念去提升 DKVMN 表现的方法。因为知识概念是离散特征，我们可以采用离散特征独热编码，后离散特征 Embedding 的方法，将离散特征表示成向量，然后将向量采用和难度关卡特征类似的方法加入模型进行预测。我们也同样做了这个实验，在表中见“DKVMN-KC”一行，其模型 AUC 为 0.714，相对于 DKVMN (AUC=0.711)，只有很小的提升。在 DKT 中加入多级知识概念特征，其模型表现见“DKT-KC”一行，模型的效果没有提升，反而略微有些下降，平均 AUC 由 0.711 降到 0.703。因此，很有必要针对知识概念这个特征设计合理的模型结构来充分发挥知识概念对深度知识追踪的提升。

#### 3.4.4 其他习题特征的增益

我们之后设计实验评估了其他特征，包括习题难度，关卡，做题时间等，对于知识追踪模型的影响。实验结果如表 3-1 所示。可以看到“DKVMN-CA+Difficulty”，“DKVMN-CA+Stage”，和“DKVMN-CA+Duration”这三行。这三个模型的 AUC 相比于 DKVMN-CA 均有提升，比如，“DKVMN-CA+Stage”的模型平均 AUC 为 0.728，相比于 DKVMN-CA 提升 AUC 0.04，且比目前最好模型 DKVMN (AUC=0.712) 有着更加明显的性能提升。在加入多个特征的“DKVMN-CA+Stage+Difficulty”模型，其最好的表现 AUC 达 0.739，对应地高出 DKVMN 模型 1.9%，高出 DKT 模型 2.7%。因此这些在在线教育系统中常出现的学生行为特征与习题特征等采用合理的特征表达方式与模型结构，均对提升模型效果有一定帮助。

### 3.5 本章小结

本章主要通过分析 IPS 日志中学生习题作答结果与题目特征、学生学习行为特征的关系，改进动态键值记忆网络，使其能够有效地支持知识概念特征、题目难



度、关卡、做题时间等特征，进一步提升知识追踪模型的性能，为下一步的习题推荐系统提供了性能良好的学生模型。具体工作如下：

- (1) 观察并分析了习题日志中习题知识概念，题目难度，关卡等特征与学生习题作答结果之间的相关性，发现这些特征均与学生作答有一定的相关性，可以用于知识追踪。
- (2) 针对新的特征加入，采用合适的特征表示的方法对特征进行处理，对离散特征如题目难度，关卡等，通过离散特征 **Embedding** 的方法向量化，而对于时间这种连续数值特征，将连续数值离散化后，再通过离散特征 **Embedding** 的方法进行向量化表示，从而避免了数据稀疏的问题。针对题目知识概念，我们有针对性地设计了知识概念记忆矩阵结构，并改进了知识概念权重的计算方法，在预测作答结构时候考虑了关卡，题目难度特征，在记忆矩阵更新时候，将作答时间特征加入更新过程来提升模型的性能。
- (3) 设计多个实验，通过分析比较各个模型在测试集上 AUC 值，证明了我们基于知识概念的模型 DKVMN-CA 能够有效地利用知识概念特征，从而明显地提升了模型的性能，相比于 DKVMN 算法 (baseline, AUC=0.712)，我们的模型 AUC 为 0.724，有 1.2% 的性能提升。进而基于 DKVMN-CA，我们加入了新的特征，如题目难度，关卡，做题时间等，均不同程度地提升了模型的性能，相比于 DKVMN，最大提升 AUC 1.9%。

## 4 习题推荐系统

本章介绍我们提出的基于深度强化学习的习题推荐系统。该系统采用第三章介绍的 DKVMN-CA 模型作为学生模拟器，通过深度强化学习训练得到优化的习题推荐策略，下面首先介绍习题推荐系统的基本思路，然后主要针对习题推荐建模与策略优化两个方面详细介绍我们的设计。

### 4.1 基本思路

为了解决习题推荐策略不能持续地提升学生的能力使学生达到最好的成绩的问题，我们将习题推荐建模为 POMDP 过程，采用第三章介绍的 DKVMN-CA 知识追踪模型作为学生模拟器，通过强化学习算法优化习题推荐策略，因为强化学习策略在推荐习题时候考虑推荐习题长期对学生的成绩提升的影响，使习题推荐策略能够持续提升学生的成绩。因此，习题推荐系统能够根据学生的原生做题历史进行习题推荐，持续提升学生成绩并最大化学生的知识能力提升。

### 4.2 习题推荐模型

自适应地根据学生的知识状态进行习题推荐来使学生更好地掌握知识，本质上是一个自适应的导学系统，而自适应导学系统需要一个学生模型来模拟学生去准确地预测习题作答结果<sup>[1]</sup>。因此，综合考虑模型的运算速度与模型的表现，我们采用第三章提出的 DKVMN-CA 来做我们的学生模拟器。

我们参考文献<sup>[37]</sup>，我们把推荐过程建模为一个 POMDP 过程。形式上，POMDP 是一个 7 元组， $(S, A, T, R, \Omega, O, \gamma)$ ， $S$  表示状态集合，对于我们的推荐系统来说，就是学生的隐知识状态，并且动作集合  $A$  是待推荐的习题集合。在时间  $t$ ，强化学习中的智能体无法观察到学生的隐知识状态  $s_t$ ，但是可以观察到学生的作答情况  $o_t \in O$ ，它是根据隐知识状态  $s_t$  的条件概率  $p(o_t | s_t)$  得到。因此，在时间  $t$ ，智能体需要在学生  $t$  时刻之前的做题历史观察  $h_t$ ，来决定推荐哪个题目最适合当前学生的知识状态。学生在完成了推荐的题目  $a_t$  后，学生的隐知识状态通过状态转移方程  $p(s_{t+1} | s_t, a_t)$  由  $s_t$  转到  $s_{t+1}$ 。

我们定义智能体推荐的题目  $a_t$  的单步奖赏为：

$$r_t = \frac{1}{K} \sum_{i=1}^K P_{t+1}(q_i) \quad (4-1)$$

其中， $K$  是一级知识概念为“数论”的习题的数目， $P_{t+1}(q)$  表示学生在练习完

题目 $a_t$ 后，状态转移到 $s_{t+1}$ ，在该隐知识状态下正确作答 $q$ 题目的概率，因此奖赏函数便是在当前隐知识状态下，正确回答“数论”知识概念下各个题目概率的平均值，这个平均值衡量了学生“数论”这一大知识概念的掌握情况。我们命名这个平均值为该知识概念的预测知识状态。

对于习题的推荐的策略 $\pi$ ，其推荐的题目的奖赏期望 $R$ 我们可以表示为：

$$R = \mathbb{E}_{\tau}[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t)] \quad (4-2)$$

其中，路径 $\tau = (s_1, o_1, a_1, s_2, o_2, a_2 \dots)$ 是由策略 $\pi$ 的路径分布决定的： $p(s_1)p(o_1|s_1)\pi(a_1|h_1)p(s_2|s_1, a_1)p(o_2|s_2)\pi(a_2|h_2)\dots$ ， $\gamma$ 为奖赏折扣因子，取值在0到1之间。对于智能体，其只能通过历史观测值进行决策，对于公式4-2所对应的动作值函数 $Q^{\pi}$ ，在时间 $t$ 的推荐习题 $a_t$ 的长期累积奖赏为：

$$Q^{\pi}(h_t, a_t) = \mathbb{E}_{s_t|h_t}[r_t(s_t, a_t)] + \mathbb{E}_{\tau>t|h_t, a_t}[\sum_{i=1}^{\infty} \gamma^i r(s_{t+i}, a_{t+i})] \quad (4-3)$$

其中 $\tau > t = (s_{t+1}, o_{t+1}, a_{t+1} \dots)$ 是在 $t$ 时刻后的未来的题目推荐路径。因此，策略推荐能够最大化以上奖赏函数值的题目 $q'$ ， $q' = \max_a Q^{\pi}(h_t, a_t)$ 。我们基于GRU神经网络利用置信域策略优化算法解决这个 POMDP 问题<sup>[46]</sup>，也就是通过神经网络表示策略 $\pi$ ，实现 $\pi(h_t) = a_t$ ， $a_t$ 即为当前观察值 $h_t$ 下，长期累积奖赏最大的习题。我们采用文献<sup>[15]</sup>的随机技巧来降低送入策略网络的独热编码的维度，以让模型能够支持大规模的习题推荐。算法通过 RLLAB 强化学习算法库来实现。

### 4.3 策略优化

这节主要介绍如何通过 DKVMN-CA 学生模拟器和强化学习进行策略优化。

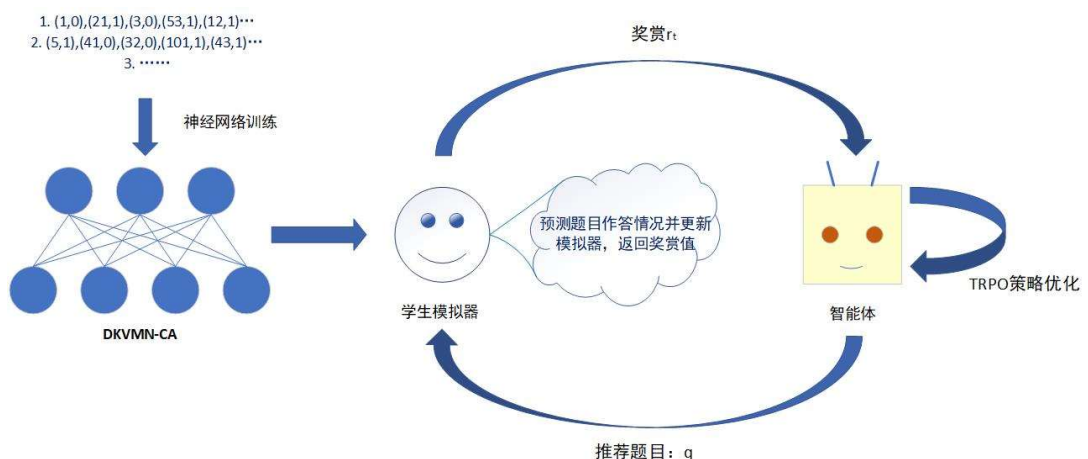


图 4-1 策略优化

Figure 4-1 Policy Optimization

我们通过 TRPO 强化算法对策略进行了 100 轮的迭代优化，并且每一轮结束后都采用随机知识状态的学生模型对策略进行评估，其具体方法是求取推荐习题路径上动作（推荐的习题）奖赏值的平均值，这个奖赏值就是“数论”知识概念的预测知识状态。习题推荐策略的优化过程如图 4-1 所示，采用训练好的知识追踪模型 DKVMN-CA 作为学生模拟器，其与智能体不断进行交互采样，具体地，智能体根据当前策略给学生模拟器推荐题目，学生模拟器预测题目作答情况，并根据作答情况更新其知识状态，同时返回公式 4-1 所定义的单步奖赏值 $r_t$ 。基于这些交互样本智能体通过 TRPO 算法对策略进行优化。

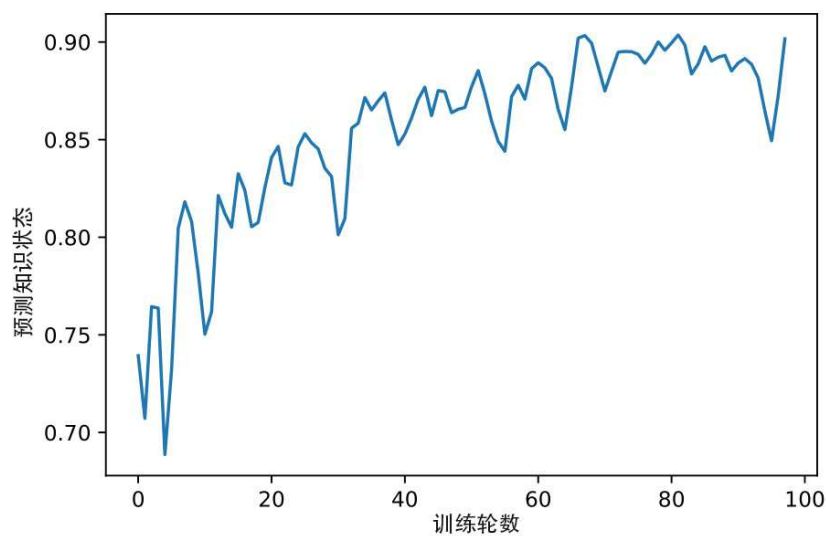


图 4-2 策略性能评估

Figure 4-2 Policy Performance Evaluation

我们通过图 4-2，可以看到通过 TRPO 算法进行策略优化，可以使策略的推荐奖赏随着训练轮数的增加而增加，策略的性能能够达到一个良好且稳定的状态，由于每轮学生模拟器均随机初始化，模拟不同能力的学生，因此，每轮的策略性能在学生能力的表现上来看，稍有差别，这也是从图中看到策略性能曲线出现波动的原因。下面我们将通过和启发式的 baseline 推荐策略进行对比，来进一步说明基于强化学习的推荐策略能够更有效地加快学生的学习进程，提高学习效果。

#### 4.4 知识增长过程评估

我们采用了 Google 提出的 Expectimax<sup>[15]</sup>启发式习题推荐策略作为 Baseline 算法，与通过强化学习 TRPO 优化的策略进行对比。Expectimax 算法将对学生的习题推荐过程建模为马尔可夫决策过程，其中，状态  $S$  为学生的知识状态，奖赏  $R$

为学生的预测知识状态，动作则是推荐的题目，状态转移概率  $P$  则是成功作答推荐的题目的概率。每一步的习题推荐都会遍历所有的备选题目，选择完成该题目练习后使学生获得最高的奖赏期望值的题目进行推荐，这是一种启发式的推荐算法，它的推荐仅仅考虑最大化下一步的收益，具体的实现见算法 2。在算法 2 中，我们使用 Expectimax 算法来给学生连续推荐了 50 道题目，并且算法返回了学生模拟器每一次完成推荐题目后的预测知识状态。

---

**Algorithm 2** Expectimax Algorithm
 

---

**Require:****S:** Student model**Q:** Exercise set**H:** History of student exercises**Ensure:**

Recommendation rewards

```

/* Init student model with the history of his exercises */
1:  $S.Init(H)$ 
   /* Recommended times */
2:  $Step \leftarrow 50$ 
3:  $t \leftarrow 0; rewards \leftarrow []$ 
4: for  $t < Step$  do
5:    $SC \leftarrow Copy(S)$ 
6:    $tr \leftarrow 0$ 
   /* Find the exercise whose expect reward is highest*/
7:   for  $q \in Q$  do
8:      $p \leftarrow SC.Predict(q)$ 
9:      $er \leftarrow p * SC.Update(q, 1).reward + (1 - p) * SC.Update(q, 0).reward$ 
10:    if  $er > tr$  then
11:       $target \leftarrow q$ 
12:       $tr \leftarrow er$ 
13:    end if
14:  end for
   /* Update student model */
15:   $n \leftarrow Random(0, 1)$ 
16:   $p \leftarrow S.Predict(target)$ 
17:  if  $n < p$  then
18:     $S.Update(target, 1)$ 
19:  else
20:     $S.Update(target, 0)$ 
21:  end if
22:   $r \leftarrow S.reward$ 
23:   $rewards.append(r)$ 
24:   $t \leftarrow t + 1$ 
25: end for
26: return rewards

```

为了比较两种推荐策略，我们首先随机地选择了 15 个学生，对于每一个学生，我们都做基于上面两种推荐策略的习题推荐实验。在每个实验中，我们先通过学生的做题历史将学生模拟器进行初始化，初始化后为当前学生的知识状态。然后我们使用当前的推荐策略连续给学生推荐 50 道题目，在这 50 步的推荐过程中，我们记录了这 15 个学生在每一步的预测知识状态的平均值。结果如图 4-3 所示，通过

强化学习进行优化的策略我们记为“RL Policy”，在经过 50 道题目的练习后，RL 策略导学的学生在预测知识状态值明显高于 Expectimax 策略，高出 5%，且随着推荐序列长度的增加，这个差距也在扩大。在 10 个题目推荐后，Expectimax 策略几乎无法再提升学生的知识状态，这意味着，策略无法找到合适的题目进行推荐，使其能力得到提升，相比之下，RL 策略因为每一步的推荐均考虑长期的奖赏，它总能发现适合题目给学生练习，进而提升学生的成绩。

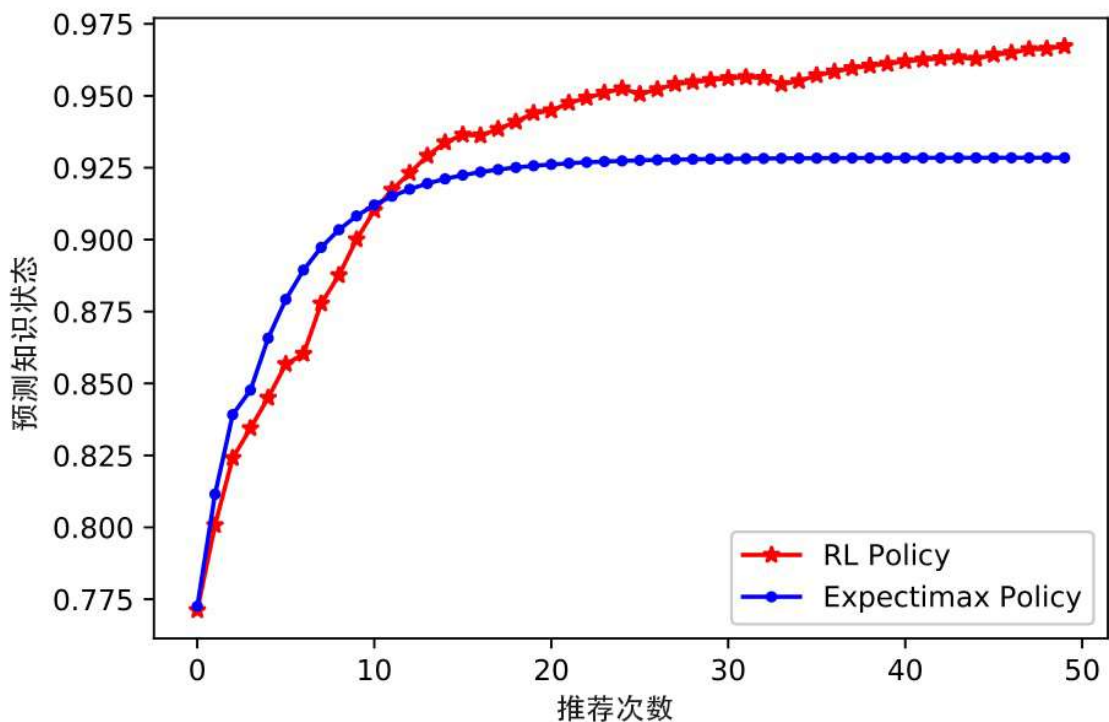


图 4-3 平均预测知识状态变化

Figure 4-3 Average Predicted Knowledge Status Change

## 4.5 习题推荐评估

我们设计了另外实验去评估真实的 RL 推荐策略的习题推荐效果，并通过可视化的方法，展示学生在各个二级知识概念的预测知识状态的变化情况。我们随机选择了历史做过 5 道题目的学生，并利用其做题历史初始化了学生模拟器。然后通过 RL 推荐策略，给该学生推荐了 5 道题目，图 4-4 展示了学生所做 10 道题目的 ID，涉及的二级知识概念，和作答结果。比如，第一个作答记录为 (88, 5, 0)，这表示了策略给学生推荐了 88 题，该题目涉及第 5 个二级知识概念，学生在该题目的作答情况是错误作答。从图中我们可以看到，这 10 个题涉及 6 个知识概

念，我们也在图中绘制了学生在这 6 个知识概念的预测知识状态的变化情况。比如，当学生做错了 88 题，88 题所涉及的第五个知识概念的预测知识状态相对来说就比较低，颜色较深，而 923 题，学生做对了，题目所涉及的第四个知识概念的预测知识状态相对较高，颜色较浅。学生做题历史未涉及到的知识概念，这些知识概念的预测知识状态我们用黑色进行标识。

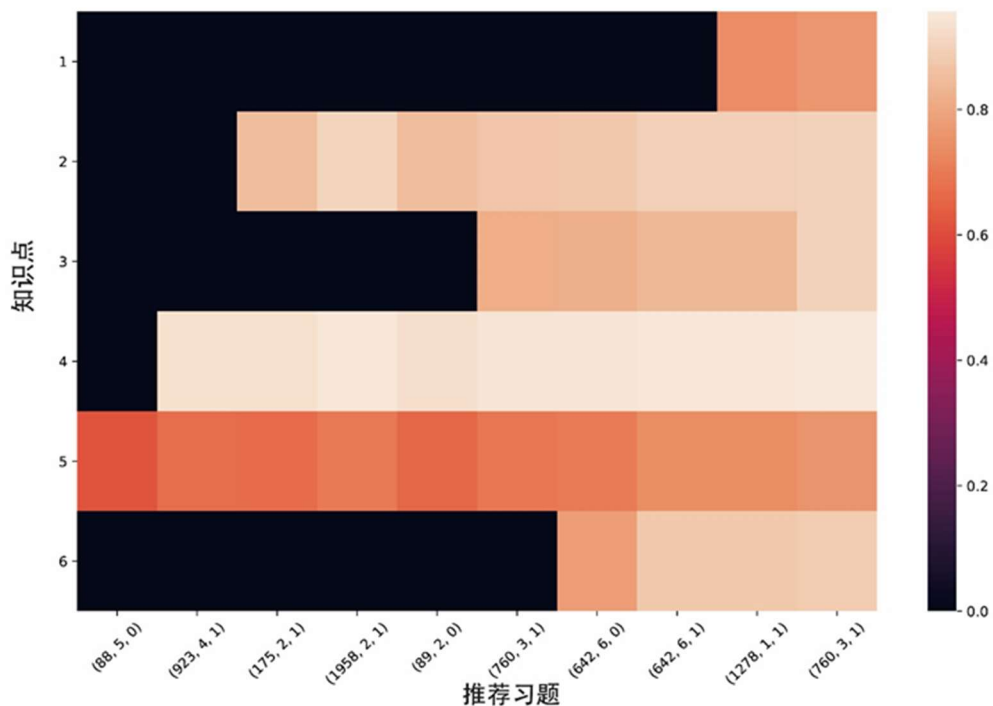


图 4-4 学生预测知识状态变化

Figure 4-4 Student Predicted Knowledge Status Change

我们现在观察推荐的题目序列，有以下的观察结果：

- 1) 如图 4-4 所示，前五个题目与概念 2, 4, 5 有关，且后面的 5 个推荐题目和概念 1, 3, 6 有关，这说明 RL 推荐策略尝试探索学生在其他概念的掌握情况，来综合提升学生“数论”一级知识概念的整体掌握程度。因此，这种推荐是有意义的。
- 2) 在学生成功正确作答 ID 为 760 习题，760 习题与概念 3 有关（分解质因数），算法向学生推荐了习题 642，642 与概念 6 有关（最大公约数与最小公倍数），而概念 6 与概念 3 有一定的关系，因此，这种相关知识概念的习题推荐是有意义的。
- 3) 当学生错误地作答题目 642，习题推荐策略再次向学生推荐了题目 642，这一次，学生成功地做对该题目，这意味着，习题推荐策略发现当学生做

错某到题目，再次给他推荐该题目，学生有可能做对，且学生能获得最大的收获。这种推荐方式，是有意义的。

- 4) 在学生成功地完成了 642 题目，642 题目与概念 6 有关（最大公约数与最小公倍数），如图三所示，学生在概念 3（分解质因数）的预测知识状态也得到略微的提升，这说明了概念 3 与概念 6 确实相关，同时，在第二个观察中也说明了这一点。
- 5) 在推荐完 642 题目后，策略再次转向其他的概念，它向学生推荐了习题 1278，1278 习题与概念 1 有关。但是通过图 4-4 可以看出，即使学生正确地作答了该题目，然而，概念 1 的预测知识状态仍然不高，这是因为 1278 题目比较简单。

## 4.6 本章小结

本章主要介绍了我们的基于深度强化学习进行策略优化的习题推荐系统。该系统采用改进的 DKVMN-CA 知识追踪模型作为学生模拟器，策略根据学生原生的做题历史，进行习题推荐。相比于启发式算法，策略的每次推荐习题，都考虑其长期收益，而非下一步的短期收益，因此，策略总是能够找到一条合适的做题路径来持续地提升学生的成绩。具体工作如下：

- (1) 通过 DKVMN-CA 搭建了学生模拟器，以完成强化学习中智能体与学生环境进行交互采样的任务，进而优化策略。
- (2) 采用真实的学生做题数据，多次进行实验，分析对比了启发式习题推荐策略 Expectimax 算法与深度强化学习推荐策略在提升学生知识状态方面的区别，启发式算法 Expectimax 最大化当前学生的短期的成绩提升，虽然经过较少的题目推荐，学生的能力也能快速提升，但是很容易使学生进入成绩提升的瓶颈，即无法再进行合适的习题推荐使学生的成绩得到明显提升。而强化学习的习题推荐策略，使每一步的习题推荐都考虑未来一定时间的收益，其习题推荐策略不仅使学生在短时间内成绩快速提升，且在有限的习题推荐次数中，总能根据学习的做题历史，找到合适的习题推荐路径，让学生的成绩持续提升。在 50 次的习题推荐中，采用强化学习习题推荐策略给学生进行推荐，其预测知识状态相比启发式的 Expectimax 算法高 5%，且随着推荐次数的增加，差距继续增大。
- (3) 可视化学生的习题推荐过程与知识状态变化过程，具体分析强化学习推荐策略的推荐习题，发现推荐的习题的知识概念与历史题目的知识概念有关系，比如“最大公约数与最小公倍数”与“分解质因数”；推荐策略会让学



生进行尝试过去没有做过的知识概念的题目，发现其在该知识概念的掌握情况，以综合提升学生在各个知识概念的能力；推荐策略会根据学生的做错题目进行针对性地再次推荐，来让学生训练，掌握其薄弱的地方。这说明了 TRPO 算法应用于智能导学系统中习题推荐的有效性和可行性。

## 5 总结及展望

### 5.1 总结

本文基于真实的在线教育平台 IPS 的学生习题作答行为数据，测量并分析了多种与学生作答结果相关的特征，并基于这些特征和动态键值记忆网络，设计了概念敏感的动态键值记忆网络 DKVMN-CA，显著地提升了知识追踪的性能。然后创造性地将深度强化学习应用于习题推荐中去，设计了新的习题推荐系统。实验结果证明该系统相比于传统的启发式习题推荐，学生能有效且持续地提升学习成绩。具体贡献如下：

- (1) 为了提升知识追踪模型的性能，创新性地题目概念的等级结构引入神经网络的元网络的设计，设计了新的深度学习知识追踪模型，改进了模型追踪性能。通过测量真实的学生做题行为数据，发现习题难度，关卡，多级知识概念特征等与学生作答结果之间的相关性，并采用合理的特征表示方法，对题目难度，关卡，做题时间等特征进行科学地表示。基于动态键值记忆网络针对题目的多级知识概念特征进行模型修改，以有效地利用多级知识概念特征来预测学生习题作答情况，经过实验，加入多级知识概念特征的 DKVMN-CA 模型相比于原模型在 AUC 上提升 1.2%，明显高于不进行模型修改只进行特征拼接方法的 0.2% 的提升。又在 DKVMN-CA 模型的基础上加入新的特征，如关卡特征，题目难度，做题时间等特征，进一步提升了知识追踪模型的性能，DKVMN-CA 模型 AUC 最高 0.739，相比于 DKT 模型 AUC 高出 2.7%，相比于目前知识追踪效果最好的 DKVMN 模型 AUC 高 1.9%。
- (2) 创新性将深度增强学习引入习题推荐算法中，提出基于深度强化学习的习题推荐策略。根据题目推荐的具体场景，即学生针对某个一级知识概念进行专题训练，设计了习题推荐策略。将习题推荐建模为部分可观测马尔可夫决策过程，使用深度强化学习来获得习题推荐策略，习题推荐策略能够根据学生的做题历史来推荐习题。通过改进的知识追踪模型 DKVMN-CA 建立学生模拟器，即强化学习要素中的环境模型，通过学生环境模型，准确地预测某一学生对某一道题目做对的概率，并将其应用于深度强化学习的智能体训练中去。实验证明：通过强化学习训练获得的策略能更持续且有效地提升学生的成绩，知识水平相比于采用启发式 Expectimax 算法做

习题推荐策略高 5%，且差距随着推荐次数的增加，进一步扩大。可视化分析表明：深度增强学习在智能习题推荐中是有效和可行的。据我们所知，这是目前首次将深度增强学习应用于数学习题推荐的具体应用场景，为未来的习题推荐方法提供了新的参考。

## 5.2 未来工作展望

随着人工智能技术的不断更新与在线教育产业的不断发展，将人工智能应用于教育，实现智能教育，对于提升教育质量，加快推动人才培养模式、教学方法改革，构建包含智能学习、交互式学习的新型教育体系有着重要意义。深度强化学习为自适应教育提供了新思路，通过深度知识追踪建立性能良好的学生模型又是强化学习的重要要素之一。未来将着重进一步提升深度知识追踪模型的性能，将自然语言处理技术应用于题目文本，建立新的题目向量空间，将习题文本特征应用于知识追踪。在强化学习推荐策略方面，如何设计更加有效合理的奖赏函数，以进一步提高习题推荐的质量，仍有很大的发展空间。

## 参考文献

- [1] Pelánek R, Jarusek P. Student Modeling Based on Problem Solving Times[J]. *International Journal of Artificial Intelligence in Education*. 2015, 25(4):493-519.
- [2] Corbett A T, Anderson J R. Knowledge tracing: Modeling the Acquisition of Procedural Knowledge[J]. *User Modeling and User-Adapted Interaction*, 1994, 4(4):253-278.
- [3] Wang Z, Zhu J, Li X, et al. Structured Knowledge Tracing Models for Student Assessment on Coursera[C]// *ACM Conference on Learning*. ACM, 2016.
- [4] Baker R S, Corbett A T, Aleven V. More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing[J]. *Lecture Notes in Computer Science*, 2008, 5091:406-415.
- [5] Yudelson M V, Koedinger K R, Gordon G J. Individualized Bayesian Knowledge Tracing Models[M]// *Artificial Intelligence in Education*, 2013.
- [6] Pardos Z A, Heffernan N T. KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model[M]// *User Modeling, Adaption and Personalization*. Springer Berlin Heidelberg, 2011.
- [7] Pavlik P I, Cen H, Koedinger K R. Performance Factors Analysis -- A New Alternative to Knowledge Tracing[C]// *Conference on Artificial Intelligence in Education: Building Learning Systems That Care: from Knowledge Representation to Affective Modelling*. IOS Press, 2009.
- [8] Cen H, Koedinger K, Junker B. Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement[C]// *International Conference on Intelligent Tutoring Systems*, 2006.
- [9] Baker R S J D, Pardos Z A, Gowda S M, et al. Ensembling Predictions of Student Knowledge within Intelligent Tutoring Systems[C]// *International Conference on User Modeling*. Springer-Verlag, 2011.
- [10] Mohammad K, Rowan M. W., Robert V. L, et al. Incorporating Latent Factors into Knowledge Tracing to Predict Individual Differences in Learning[C]. *Conference on Educational Data Mining*, 2014.
- [11] Khajah M M, Huang Y, González-Brenes JP, et al. Integrating Knowledge Tracing and Item Response Theory: A Tale of Two Frameworks[J]. *Personalization Approaches in Learning Environments*, 2014.
- [12] Anderw S. Lan, Andrew E. Waters, Christoph Studer, BLAh: Boolean Logic Analysis for Graded Student Response Data[J], *IEEE Journal of Selected Topics in Signal Processing*, 2017, 11(5):754-64.
- [13] Reddy S, Labutov I, Joachims T. Latent Skill Embedding for Personalized Lesson Sequence Recommendation[J]. *CoRR*, abs/1602.07029, 2016.
- [14] Khajah M, Lindsey R V, Mozer M C. How Deep is Knowledge Tracing[J]. *arXiv preprint arXiv:1604.02416*, 2016.

- [15] Piech C, Bassen J, Huang J, et al. Deep knowledge tracing[C]// Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, 2015.
- [16] Yeung C K, Yeung D Y. Addressing Two Problems in Deep Knowledge Tracing via Prediction-Consistent Regularization[C]. Annual ACM Conference, 2018.
- [17] Yu Su, Qingwen Liu, Qi Liu, Zhenya Huang. Exercise-Enhanced Sequential Modeling for Student Performance Prediction[C]. The Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
- [18] Yang H, Cheung L P. Implicit Heterogeneous Features Embedding in Deep Knowledge Tracing[J]. Cognitive Computation, 2017.
- [19] Zhang L. Incorporating Rich Features into Deep Knowledge Tracing[J]. Masters Theses, 2017.
- [20] Zhang J, Shi X, King I, et al. Dynamic Key-Value Memory Networks for Knowledge Tracing[J]. WWW, 2016.
- [21] Chaudhry R, Singh H, Dogga P, et al. Modeling Hint-Taking Behavior and Knowledge State of Students with Multi-Task Learning[C]. Conference on Educational Data Mining. 2018.
- [22] Yigal Rosen, Ilia Rushkin, Rob Rubin, Liberty Munson, Andrew Ang, Gregory Weber, Glenn Lopez, Dustin Tingley, The Effects of Adaptive Learning in a Massive Open Online Course on Learners' Skill Development[C]. Annual ACM Conference on Learning at Scale, 2018.
- [23] Ralf Teusner, Thomas Hille, Thomas Staubitz, Effects of Automated Interventions in Programming Assignments-Evidence from a Field Experiment[C]. Annual ACM Conference on Learning at Scale, 2018.
- [24] Chounta IA, McLaren BM, Albacete PL, et al. Modeling the Zone of Proximal Development with a Computational Approach[C]. Conference on Educational Data Mining, 2017.
- [25] Wang Q, Zeng C, Zhou W, et al. Online Interactive Collaborative Filtering Using Multi-Armed Bandit with Dependent Arms[J]. IEEE Transactions on Knowledge and Data Engineering, 2017.
- [26] Li L, Chu W, Langford J, et al. A Contextual-Bandit Approach to Personalized News Article Recommendation[J]. WWW, 2010.
- [27] Clement B, Roy D, Oudeyer P Y, et al. Multi-Armed Bandits for Intelligent Tutoring Systems[J]. Journal of Educational Data Mining, 2013, 7(4): A-705.
- [28] Mu T, Goel K, Brunskill E, Program2Tutor: Combining Automatic Curriculum Generation with Multi-Armed Bandits for Intelligent Tutoring Systems[C]. Conference on Neural Information Processing Systems, 2017: 424-429.
- [29] Mu T, Wang S, Andersen E, et al. Combining Adaptivity with Progression Ordering for Intelligent Tutoring Systems[C]. Annual ACM Conference on Learning at Scale, 2018.
- [30] Andrew S. Lan, Richard G. Baraniuk, A Contextual Bandits Framework for Personalized Learning Action Selection[C]. Conference on Educational Data Mining,

- 2016.
- [31] Kolchinski Y A, Ruan S, Schwartz D, et al. Adaptive Natural-Language Targeting for Student Feedback[C]// ACM Conference on Learning at Scale. ACM, 2018.
  - [32] Piech C J. Uncovering Patterns in Student Work: Machine Learning to Understand Human Learning[D]. Stanford University, 2016.
  - [33] Rafferty A N, Brunskill E, Griffiths T L, et al. Faster Teaching via POMDP Planning[J]. Cognitive Science, 2015.
  - [34] Whitehill J, Movellan J. Approximately Optimal Teaching of Approximately Optimal Learners[J]. IEEE Transactions on Learning Technologies, 2017:1-1.
  - [35] Reddy S, Dragan A, Levine S. Accelerating Human Learning with Deep Reinforcement Learning[J]. Teaching Machines, Robots, and Humans of NIPS, 2017.
  - [36] Mandel T, Liu Y E, Levine S, et al. Offline Policy Evaluation Across Representations with Applications to Educational Games[C]. Conference on Autonomous agents and multi-agent systems, 2014:1077-1084.
  - [37] Arulkumaran K, Deisenroth M P, Brundage M, et al. A Brief Survey of Deep Reinforcement Learning[J]. arXiv preprint arXiv:1708.05866, 2017.
  - [38] 周志华. 机器学习[M]. 清华大学出版社, 2016.
  - [39] Cho K, Berrienoer B V, Gulcehre C, et al, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[C]. Conference on Empirical Methods on Natural Language Processing, 2014.
  - [40] Krizhevsky A , Sutskever I , Hinton G. ImageNet Classification with Deep Convolutional Networks[J]. Advances in Neural Information Processing Systems, 2012, 25(2):84-90.
  - [41] Mnih V, Badia, Adrià Puigdomènech, Mirza M, et al. Asynchronous Methods for Deep Reinforcement Learning[J]. arXiv preprint arXiv:1602.01783, 2016.
  - [42] Schulman J, Levine S, Moritz P, et al. Trust Region Policy Optimization[J]. Computer Science, 2015:1889-1897.
  - [43] <http://www.tensorfly.cn/>.
  - [44] Li Y, Du N, Bengio S. Time-Dependent Representation for Neural Event Sequence Prediction[J]. ICLR, 2018.
  - [45] Pascanu R, Mikolov T, Bengio Y. On the Difficulty of Training Recurrent Neural Networks[J]. arXiv preprint arXiv:1211.5063, 2012.
  - [46] Chung J, Gulcehre C, Cho K H, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

## 作者简历及攻读硕士学位期间取得的研究成果

### 一、作者简历

艾方哲，男，1994年11月生。2013年9月至2017年7月就读于曲阜师范大学物理工程学院通信工程专业，取得工学学士学位。2017年9月至2019年6月就读于北京交通大学电子与通信工程专业，研究方向是信息网络，取得工程硕士学位。攻读硕士学位期间，主要从事智能导学系统中习题推荐策略方面的研究工作。

### 二、发表论文

- [1] Fangzhe Ai, Yishuai Chen, Yuchun Guo, et al. Concept-Aware Deep Knowledge Tracing and Exercise Recommendation in an Online Learning System[C]. The 12th International Conference on Educational Data Mining, 2019.

### 三、参与科研项目

- [1] 基于知识追踪的智能导学系统  
[2] 基于大规模在线视频流系统的会话级 QoE 预测  
[3] 基于机器学习算法的交通指标预测

## 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：



签字日期：

2019年5月30日



## 学位论文数据集

表 1.1: 数据集页

关键词*	密级*	中图分类号	UDC	论文资助
知识追踪; 习题推荐; 预分类测; 深度强化学习	公开			
学位授予单位名称*	学位授予单位代码*	学位类别*	学位级别*	
北京交通大学	10004	工学	硕士	
论文题名*	并列题名			论文语种*
基于知识追踪的智能导学算法设计				中文
作者姓名*	艾方哲	学号*	17125001	
培养单位名称*	培养单位代码*	培养单位地址	邮编	
北京交通大学	10004	北京市海淀区西直门外上园村 3 号	100044	
学科专业*	研究方向*	学制*	学位授予年*	
电子与通信工程	信息网络	2	2019	
论文提交日期*	2019.6.3			
导师姓名*	陈一帅	职称*	副教授	
评阅人	答辩委员会主席*	答辩委员会成员		
	郭宇春	赵永祥 李纯喜 郑宏云 张立军		
电子版论文提交格式 文本() 图像() 视频() 音频() 多媒体() 其他() 推荐格式: application/msword; application/pdf				
电子版论文出版(发布者)	电子版论文出版(发布)地		权限声明	
论文总页数*	52 页			
共 33 项, 其中带*为必填数据, 为 21 项。				