Computer Science and Application 计算机科学与应用, 2026, 16(11), 142-155
Published Online November 2026 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2026.1611332

Hans汉斯

基于Flask框架的智能题库与自适应学习平台设计与实现

张　海*，吴　昊，谢姣皎，汤　炜

华侨大学信息科学与工程学院，福建　厦门

收稿日期：2026年10月6日；录用日期：2026年11月5日；发布日期：2026年11月12日

摘　要

针对传统教学辅导平台重资源存储、轻个性化服务、核心功能依赖人工、缺乏算法层智能调度等痛点，本文设计并实现了一套基于Flask框架的智能题库与自适应学习平台。系统以Flask为后端、SQLite与SQLAlchemy ORM提供数据管理，前端采用Bootstrap与MathJax实现响应式布局与公式渲染，基于Nginx + Gunicorn构建高并行开发环境。算法层面，系统融合项目反应理论(IRT)与贝叶斯知识追踪(BKT)模型，依据最大似然估计与最大信息量准则实现动态切题，并提出融入时间衰减因子的T-BKT模型补偿记忆遗忘方法。同时，集成阿里云百炼大语言模型，实现自动出题、逻辑一致性诊断、IRT参数冷启动预估、智能答疑与错因分析，形成“AI出题赋能 + 算法精准调度”的教学闭环。测试结果表明，平台功能完整，兼容性、并发性与安全性均满足设计要求，可为大规模个性化教学活动提供可行技术路径。

关键词

Flask框架，自适应学习，项目反应理论，贝叶斯知识追踪，大语言模型

Design and Implementation of an Intelligent Question Bank and Adaptive Learning Platform Based on the Flask Framework

Hai Zhang*, Hao Wu, Jiaojiao Xie, Wei Tang

College of Information Science and Engineering, Huaqiao University, Xiamen Fujian

Received: October 6, 2026; accepted: November 5, 2026; published: November 12, 2026

*通讯作者。

文章引用: 张海, 吴昊, 谢姣皎, 汤炜. 基于 Flask 框架的智能题库与自适应学习平台设计与实现[J]. 计算机科学与应用, 2026, 16(11): 142-155. DOI: 10.12677/csa.2026.1611332

Abstract

In response to the limitations of traditional teaching assistance platforms, characterized by a heavy reliance on resource storage, insufficient personalized services, core functions dependent on manual intervention, and a lack of intelligent scheduling through algorithms, this paper designs and implements an intelligent question bank and adaptive learning platform based on the Flask framework. The system employs Flask for the backend, utilizes SQLite and SQLAlchemy ORM for data management, and incorporates Bootstrap and MathJax for responsive layout and formula rendering on the front end. It is constructed on Nginx and Gunicorn to establish a highly parallel development environment. At the algorithmic level, the system integrates Item Response Theory (IRT) and Bayesian Knowledge Tracing (BKT) models. It implements dynamic topic selection based on maximum likelihood estimation and maximum information content criteria, and introduces the T-BKT model, which incorporates a time decay factor to address memory forgetting. Furthermore, the system integrates Alibaba Cloud's Baolian Large Language Model to facilitate automatic question generation, logical consistency diagnosis, IRT parameter cold start estimation, and intelligent question answering and error analysis. This integration creates a teaching loop characterized by "AI question generation empowerment and algorithmic precise scheduling." The test results indicate that the platform's functionality, compatibility, concurrency, and security all meet the design requirements, thereby providing a viable technical pathway for large-scale personalized teaching.

Keywords

Flask Framework, Adaptive Learning, Item Response Theory, Bayesian Knowledge Tracing, Large Language Model

Copyright © 2026 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 引言

传统教学辅导平台多采用一站式固化服务，涵盖题库管理、组卷、做题与批阅等标准化功能，难以满足不同学生的个性化需求。自适应学习平台通过算法分析与智能化手段深入解析学生学习行为，为其推荐更适合的学习路径与习题内容，是实现大规模个性化教学的重要途径之一。从教师角度，日常教学任务繁重，难以掌握每个学生的学情并进行针对性辅导；从学生角度，一刀切的作业布置方式无法兼顾个体差异，成绩较差者易丧失信心，成绩较好者难以持续进步。自适应学习平台可结合每个学生的作答情况，精准定位其薄弱知识点并智能推荐对应习题，从而提升整体学习效率。另外，平台引入大语言模型(LLM)解决多数平台“重资源存储、轻个性化服务”的问题，并采用项目反应理论(IRT)与贝叶斯知识追踪(BKT)等算法，实现学情的实时评估与题目精准调度[1] [2]。

目前，国外主流数据分析平台多采用 Flask 与 Python 相结合的方式进行开发。前端以 Flask 进行数据展示与调度，后端以 Python 和数据库完成数据采集与分析。Liang 等提出基于 Flask 的 Python Web 服务平台，采用“应用层－服务层－数据层”三层架构。后端用 Pandas 处理动力电池多源异构数据，前端结合 Flask 与 Pyecharts 实现动态可视化，验证了轻量级 Web 框架处理高频数据流的工程价值[3]。在自适应学习方面，项目反应理论(IRT)自 20 世纪中叶形成并不断成熟。Lord 提出单参数逻辑斯蒂模型(1PL)描述试题难度对答题概率的影响[4]，Lord 与 Novick 进一步将题目难度、区分度与猜测率整合到统一模型

中[5]。现代自适应学习引擎不仅依托 IRT 动态切题，还通过融入遗忘机制的知识追踪算法捕捉学习者的遗忘曲线[6]，使测量结果更具稳定性与可靠性。国内数据分析平台普遍采用 Flask 实现前后端分离。管玲玲等探讨了基于 Flask 与前端框架分离开发的实现路径，通过 RESTful API 实现前后端解耦与高并发负载均衡，为“实时题目展示、后台模型运算、实时 AI 调用”等模块间的稳定交互提供支撑[7]。

2. 相关技术与平台总体设计

2.1. 关键技术

2.1.1. Flask 框架与 WSGI 服务器

Flask 是基于 Python 的轻量级 Web 框架，以微核心为设计思想，负责实现智能组卷、学生能力值计算及大语言模型接口对接等核心功能。由于 Flask 自带开发服务器基于单线程设计，无法满足生产环境大量并发请求，平台遵循 WSGI (Web Server Gateway Interface)标准，将底层 Web 服务器与上层 Python 应用解耦。WSGI 核心逻辑可抽象为基于可调用对象的映射关系[8] [9]，其通信契约可表示为：

$$\text{application (environ, start_response)} \rightarrow \text{response_body} \tag{1}$$

其中 application 为 Flask 应用实例，environ 为包含 HTTP 请求信息的字典，start_response 为服务器提供的回调函数，response_body 为返回给客户端的可迭代响应体。平台采用 Gunicorn 作为线上 WSGI 服务器，以预派生多进程方式运行 Flask，由 Gunicorn 按压力情况将用户请求分配给各 Worker 进程执行，以提升整体并发性能。

2.1.2. 数据库与前端渲染

平台使用 SQLite 作为轻量级嵌入式数据库，数据由存储引擎、SQL 编译器与 API 接口组成，无需独立服务器即可将整个数据库实例置于单文件中[10]。为避免拼接原始 SQL 带来的注入风险与耦合问题，在业务逻辑层与 SQLite 之间引入 SQLAlchemy ORM 作为数据抽象中间件，其对象关系映射器可将 Python 实体类与数据库表结构建立映射，使开发者以面向对象方式操作数据，并把对象状态变化批量转化为底层 SQL 执行，从而为 IRT、BKT 等核心算法释放更多计算资源[11]。在前端，平台采用 Bootstrap 十二列栅格系统实现响应式布局，并引入 MathJax 对题库中的 LaTeX 公式进行客户端动态渲染[12]。复杂公式以轻量级纯文本形式存储于 SQLite，经 Jinja2 模板引擎插入 HTML 响应流，浏览器加载后由 MathJax 将文本重新绘制为矢量化公式，保证不同设备与高分辨率屏幕下公式边缘清晰，避免像素失真，同时便于数据埋点与学情分析。

2.1.3. Nginx + Gunicorn 部署架构

为满足系统对高并发 API 调用与大量数据运算的需求，平台采用 Nginx + Gunicorn 生产级部署方案。Nginx 负责反向代理与静态资源服务，并依据请求类型实现动静分离：CSS、JavaScript 及图片等静态资源直接由 Nginx 缓存返回；登录、组卷等需实时计算的请求则转发至 Gunicorn 容器，由其将请求转换为标准 WSGI 格式后交给 Flask 应用处理。该架构在逻辑上保证系统安全性与访问效率，在工程上构建了具有高并发处理能力与灵活扩展性的有机整体。如图 1 所示，系统整体分为客户端层、外部接入层、协议及容器层、业务应用层以及存储与外部服务层五大部分。

2.2. 自适应学习核心算法

2.2.1. 项目反应理论(IRT)模型

传统在线平台多采用经典测量理论，学生成绩受试卷难度影响较大，能力估值误差较大。为克服其局限，本文平台采用项目反应理论(IRT)。IRT 认为学生答对题目的概率取决于其潜在能力与题目属性，具有

参数分离特性。平台底层计算引擎采用最严格的三参数逻辑斯蒂模型(3PL)，学生对试题的答对概率为：

$$P(\theta)=c+(1-c)\Big/\left(1+\exp\left(-Da\left(\theta-b\right)\right)\right) \tag{2}$$

式中：θ 为能力参数，表示学生的潜在知识掌握水平；b 为难度参数，决定题目适用的能力区间；a 为区分度参数，反映题目对不同能力学生的鉴别能力；c 为猜测参数，表示能力极低学生凭随机猜测答对的下限概率；D 为常数因子(通常取 1.7)，用于使逻辑斯蒂函数逼近正态累积分布函数。上述参数对试题特征的影响可通过项目特征曲线(Item Characteristic Curve, ICC)直观表现，如图 2 所示。



Figure 1. Overall layered architecture diagram of the platform
图 1. 平台总体分层架构图

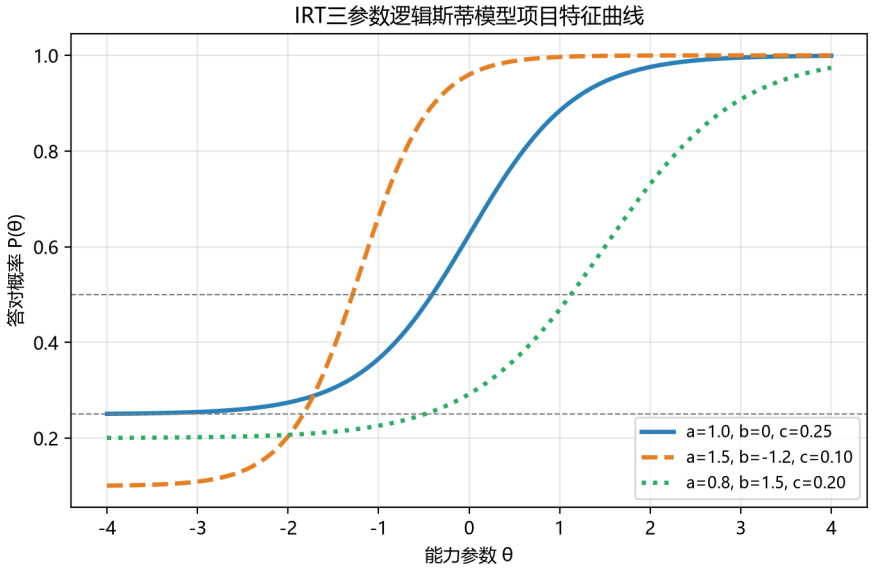


Figure 2. Schematic diagram of item characteristic curves for the three-parameter IRT model
图 2. IRT 三参数模型项目特征曲线示意图

在确定答对概率后，平台需为当前能力为 θ 的学生选择“利益最大”的题目，该问题可通过项目信息函数(Item Information Function, IIF)解决。3PL 模型下第 i 题的题目信息函数为：

$$I_i(\theta)=a_i^2\cdot Q_i(\theta)/P_i(\theta)\cdot\left[\left(P_i(\theta)-c_i\right)/\left(1-c_i\right)\right]^2 \tag{3}$$

其中 $Q_i(\theta)=1-P_i(\theta)$ 为答错概率。信息量越大，说明题目越能有效减少对学生能力估计的误差。系统根据最大信息量准则在题库中筛选信息量最大的题目推送至前端，实现“有的放矢、因材施教”。

2.2.2. 贝叶斯知识追踪(BKT)及其时间衰减改进

贝叶斯知识追踪(BKT)是基于隐马尔可夫模型(HMM)的序列模式识别方法，将学生交互数据视作时间序列，隐变量表示学生对知识点的掌握程度，观测变量为答题结果[13]。BKT 由四个核心参数构成：初始掌握概率 $P(L_0)$、学习转移概率 $P(T)$、猜对概率 $P(G)$与失误概率 $P(S)$。每次提交答案后，根据贝叶斯定理更新掌握程度：

答对时：

$$P(L\mid\text{答对})=P(L)\cdot(1-S)/\left[P(L)\cdot(1-S)+(1-P(L))\cdot G\right] \tag{4}$$

答错时：

$$P(L\mid\text{答错})=P(L)\cdot S/\left[P(L)\cdot S+(1-P(L))\cdot(1-G)\right] \tag{5}$$

随后依据学习转移概率估计学生下次练习前的学习情况，而标准 BKT 假定“知识一旦习得便不会遗忘”，难以刻画艾宾浩斯遗忘曲线。为此，平台在状态更新公式中引入时间衰减因子，构造 T-BKT 模型：

$$P(t)=P_0\cdot\exp(-\lambda\cdot\Delta t) \tag{6}$$

式中 Δt 为两次练习的时间间隔(天)，λ 为遗忘率参数。当间隔很短时衰减因子接近 1，系统主要依据最近作答平滑更新掌握度；当间隔较长时，掌握度按指数衰减，真实反映遗忘情况。当掌握度降至认知临界点后，系统判定该知识点进入“认知临界区”，将其标记为薄弱点并推送巩固习题，从而避免“复习真空期”。

2.3. 大语言模型集成

平台 AI 部分直接集成阿里云百炼大语言模型，其将底层计算资源、模型训练与推理部署抽象为即插即用的 MaaS 服务，开发者通过标准 API 即可调用工业级生成式 AI。根据不同业务需求对模型参数规模、推理时间与成本进行区分：对题目深层逻辑判断与长文本提取等复杂语义场景使用支持超长上下文的 Qwen-Max；对知识点关联分析使用 Qwen-Plus；对智能答疑、及时纠错等强时效性交互场景使用低延时推理的 Qwen-Turbo。AI 模块贯穿题目生成、学情分析与个性化推荐等核心功能。

2.4. 平台总体架构与核心功能

平台核心功能以“资源智能化构建、算法动态化调度、反馈多维化呈现”为闭环逻辑，由题库资产建设层、自适应习题推理层、教学反馈层与教学输出层四大模块构成。数据流前端为基于 AI 流式创题与文档智能解析的题库建设层；推理层采用 IRT 能力值估计与 T-BKT 掌握度更新的双因子交叉推荐策略；用户完成练习后，系统利用大语言模型输出过程性讲解，并生成多维能力画像、动态知识图谱与总结提升报告，形成“测评–诊断–提升”的闭环，如表 1 所示。

图 3 是一个完整的从外部非结构化数据到深度认知结果反馈的数据链路图。数据流最前端是题库资产建设层，系统采用基于 AI 的流式创题和文档智能化解析以克服传统题库人工维护的低效问题。在自适应习题推理层方面，系统利用双因子交叉推荐策略，实时结合 IRT 能力值估计与 T-BKT 掌握度更新，确保每一道推送题目都能精准匹配学习者的当前认知水平。用户完成练习之后，系统会在极短时间内利用

云上大语言模型输出详细、严谨的过程性讲解；当用户完成阶段性学习任务后，系统逻辑即刻由推理引擎转向教学反馈层。平台会根据作答过程语义化处理生成多维能力画像、动态知识图谱及总结提升报告等综合性评价结果。同时会立刻激活底层算法，使得 T-BKT 严格按照时间衰减原则不断更新知识点掌握概率，同时，更新用户的总体 IRT 能力水平，以形成“测评－诊断－提升”的完美闭环。

Table 1. Core functional modules of the platform
表 1. 平台核心功能模块

功能模块	主要职责	关键技术
题库资产建设层	AI 自动出题、文档解析、题目质量监测	LLM 流式生成、启发式解析、交叉验证
自适应习题推理层	能力估计、动态切题、掌握度建模	IRT、T-BKT、最大信息量准则
教学反馈层	学情画像、知识图谱、练习报告	LLM 语义分析、T-BKT 状态映射
教学输出层	智能答疑、错因分析、个性化建议	Qwen 系列大模型

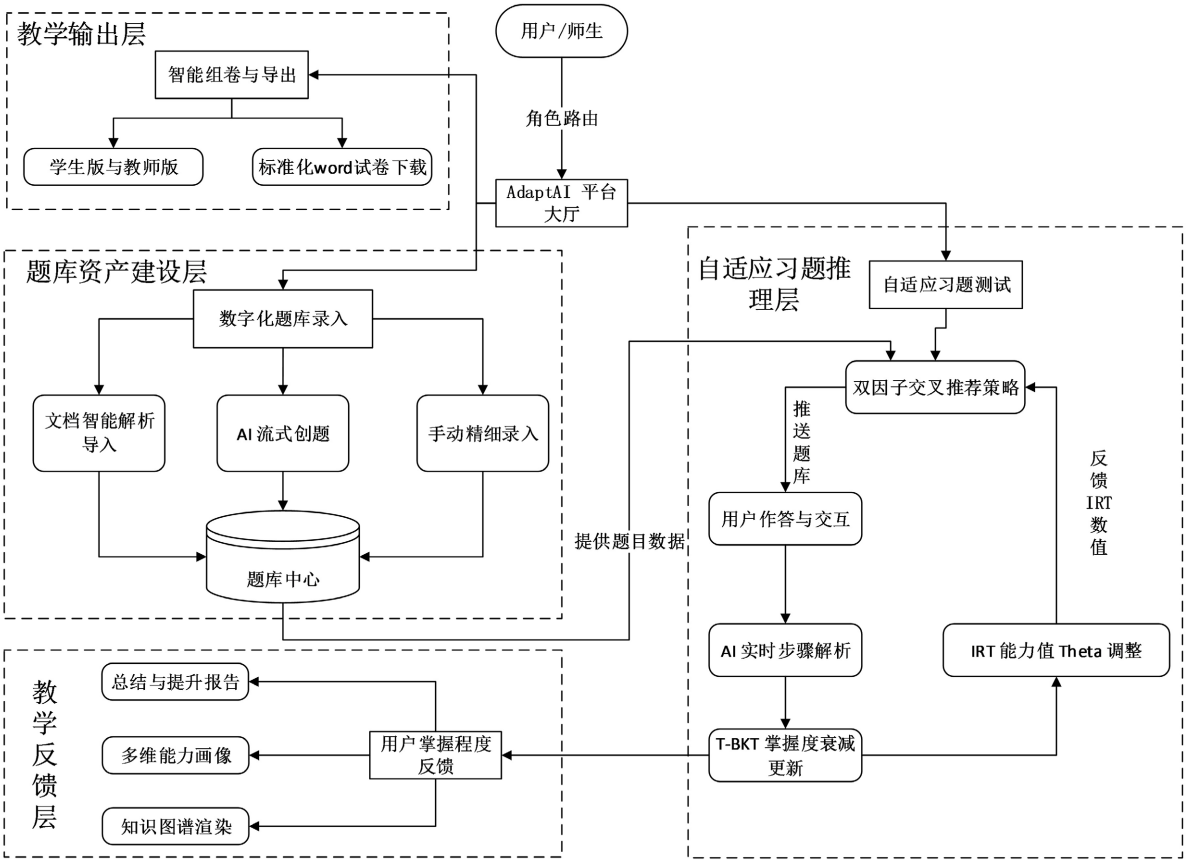


Figure 3. Architecture diagram of core functional flows for the adaptive learning platform
图 3. 自适应学习平台核心功能流转架构图

3. 多源驱动的智能题库管理系统实现

3.1. 多源题目录入

3.1.1. AI 自动生成

平台“AI 创题引擎”通过结构化 Prompt 工程与约束性思维链(CoT)实现题目自动生成。系统设置多

个约束矩阵，在提示词内限定题型、知识点与难度等“思考边界”，并加入 JSON 格式要求，使模型同时产出题目、解答过程及知识点关联程度。同时，后端对新生成题目进行多维特征提取与标签挂载，通过大语言模型的语义理解能力将其映射到知识图谱节点。针对大模型生成延迟不可预知、同步请求易超时(HTTP 504)的问题，平台采用切片流式生成技术，以分段生成方式确保每次与后端建立的连接均处于安全时间内。

3.1.2. 基于启发式规则的文档解析

为将非结构化文档转化为结构化题目信息，平台采用基于启发式规则及语义关联的解析方法。对 Word 文档，通过解析底层 OpenXML 并递归遍历 XML 树以获取段落及运行块属性；对 PDF 这一“显示导向型”文件，采用基于坐标的布局分析，先用文本流分析器获取字符物理位置，再据字符间距进行“行合并”与“块聚类”，并通过行末与行首语义连贯性判断实现逻辑流重构，以解决断行问题。解析结果交由大语言模型批量转化为 JSON 对象，由 SQLAlchemy ORM 写入关系型数据库。

3.2. 题库质量监测与 IRT 参数冷启动

针对多源上传内容易存在错误、噪声冗余与逻辑碎片化的问题，平台利用大语言模型实现题目逻辑一致性诊断。大模型在获取题干、选项与解析后，扮演“虚拟考试员”角色自行推理答案并与标准答案比对，通过则标记“校验通过”，否则停止入库并标注“异常”及具体错误信息返回模型修改，从而避免存在科学性问题的题目入库。

为解决新题目因缺乏答题数据而难以启动自适应过程的问题，平台提出 IRT 参数冷启动预估方法：根据题目文本难度、公式嵌套层数与选项干扰程度，用回归方法预设初始难度参数 b，并根据干扰度动态计算猜测参数 c 与区分度参数 a。随着系统持续收集答题反馈，后台通过滑动窗口与阈值触发机制，运用极大似然估计或贝叶斯最大后验估计对三参数进行二次校准，形成“先验预估结合后验修正”的闭环，使题库具备自我进化能力。

3.3. 基于“虚实结合”的智能组卷

传统组卷多采用随机抽取或轮流调用，难以兼顾难度、题型比例与知识点覆盖率。本文平台实现了基于多约束条件的智能组卷：调度器先从 SQLite 本地题库静态检索，若发现考点缺失或冷门，则异步调用百炼大模型按 Prompt 矩阵受控生成题目并流式放入组卷缓存，形成“本地静态检索 + 云端动态补充”的虚实结合模式。为防大模型长连接阻塞服务器线程，系统引入 Celery 分布式任务队列，检测到考点空缺即向客户端返回 202 状态码与 task_id 进行异步处理，提高系统鲁棒性。

4. 基于多算法融合的自适应学习平台实现

4.1. 自适应学习引擎总体架构

自适应学习引擎由实时测试、多维诊断、精确推送与智能辅导四部分紧密相连，IRT 算法与 BKT 模型构成双引擎动力核心。在测验环节，系统选取具有标准 IRT 参数的题目并收集各难度作答信息；在诊断环节，用 IRT 对学生即时能力进行极大似然估计，并用 BKT 监控各知识点正确率，形成能力画像；在推荐环节，依据最大信息量原则从题库中选出最适合学生“最近发展区”的题目；在辅导环节，AI 针对问题及其上下文提供个性化指导与答疑。其闭环流程如图 4 所示。

智能题库不仅是教学资源库，也是自适应学习系统中的数据源，如图 5 所示，这是系统的核心题库资产管理页面。此页面包括 AI 自动生成、智能文档导入及人工精细化录入三种方式，在页面下方以文件夹矩阵形式存储各类异构资源，涵盖了《天线原理》《微波工程》等课程。这种结构化的展示方式不仅直

观反映了系统对数字化知识资产的管理能力，更体现了系统从“原始文档”到“结构化题库”的标准化加工流程，确保了每一道入库题目都具备高精度的算法标定属性。

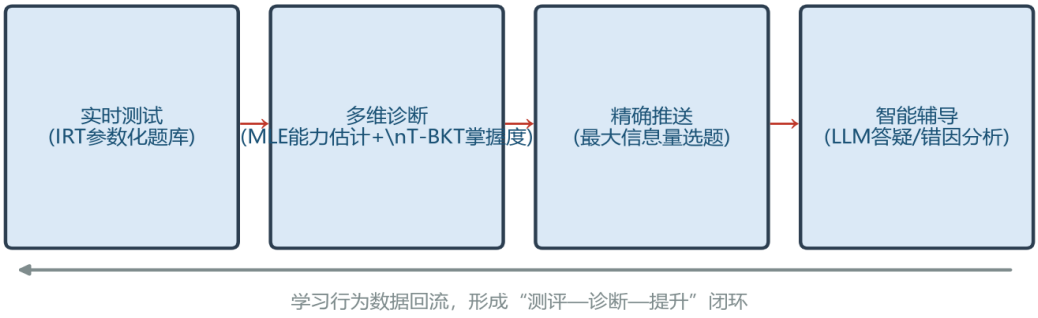


Figure 4. Adaptive learning closed-loop flowchart
图 4. 自适应学习闭环流程图



Figure 5. Interactive interface of the intelligent question bank asset management center
图 5. 智能题库资产管理中心交互界面

4.2. 基于 IRT 的动态切题策略

4.2.1. 能力水平的最大似然估计

平台未采用静态累计得分方式，而是使用项目反应理论中的最大似然估计法评估学生能力。即在已知答题序列的基础上，找到最有可能导致该结果的 θ 作为学生当前真实能力。学生刚进入模块或估计值较低时，系统将自适应难度定位到较低层次，并因 T-BKT 掌握度较低而进入“重点攻克薄弱知识点”的

复习状态；随着学生连续答对，似然函数峰值向高能力区间移动，系统随即提高自适应难度等级并向学生呈现综合性挑战题目，实现千人千面的学习路径规划。

4.2.2. 基于最大信息量准则的题目匹配

获得当前能力估计后，平台依据最大信息量准则选取使能力估计误差最小的题目，而非简单“难度一致”。如图 6 所示，三条 IIF 曲线分别对应高区分度、常规与低难度题目，系统在当前 θ 处选取信息量(垂直高度)最大的题目，从而以最高测量效率加速能力估计收敛。

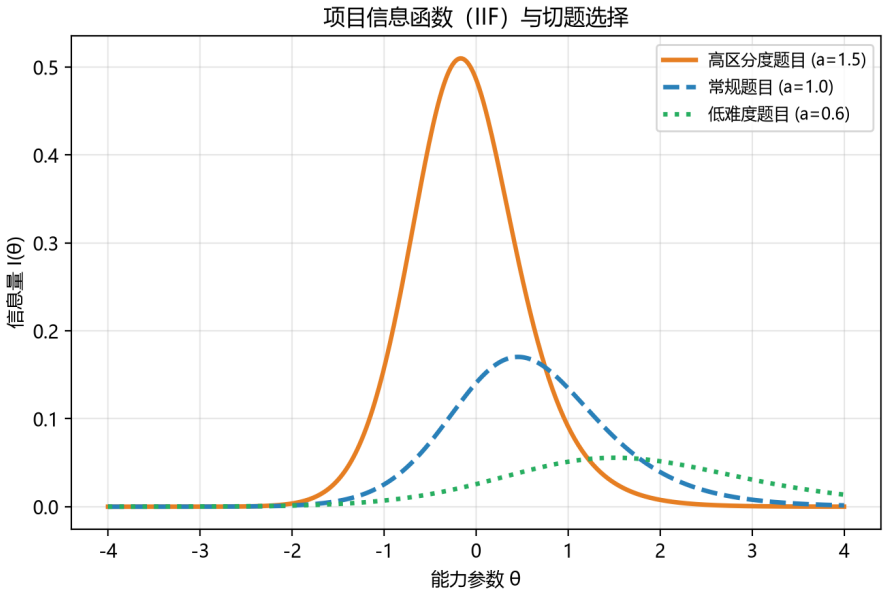


Figure 6. Project Information Function (IIF) curve and topic selection
图 6. 项目信息函数(IIF)曲线与切题选择

为避免单纯追求信息量导致题目难度跨度过大，平台在选题函数中融入曝光控制因子与难度平滑权重等多维约束：当学生能力迅速提升时，适当降低信息量权重并选择难度较低但区分度高的题目；同时对同一题目增加曝光衰减惩罚，防止学生反复接触相同题目，其核心逻辑如式(7)所示。

$$\text{Score} = I(\theta)\cdot\exp(-0.08\cdot\text{exposure})\cdot\exp\left(-(\theta-b)^2/\left(2\sigma^2\right)\right)\cdot\text{mastery_boost} \tag{7}$$

4.3. 基于 T-BKT 的掌握度实时建模

4.3.1. 遗忘补偿机制与惰性更新

为避免系统每时刻对所有用户进行大规模并行更新，平台在前端实现“按需计算”的惰性求值工作流。学生登录或点击切题时，系统读取该学生各知识点最近一次练习的时间戳，与当前时间求差得到 Δt，再以时间衰减因子对先前掌握度进行实时“折旧”；若折旧后的掌握度低于认知临界值，则触发遗忘补偿，将该知识点加入高优推送队列。学生作答后，系统以衰减后的掌握度为先验概率，结合答题正确与否及当前猜测率、失误率，用贝叶斯公式求取后验概率，并叠加学习转移概率得到最新掌握度，最后清空时间戳写入数据库。图 7 表示了该遗忘与补偿过程。

算法参数(衰减率 λ、认知临界阈值)可根据不同学科、年级或用户群体自适应调整。T-BKT 的掌握度实时更新伪代码如表 2 所示，其中“衰减 - 作答 - 更新”的数据流在准确描述遗忘的同时，也显著减少服务器不必要的工作量。

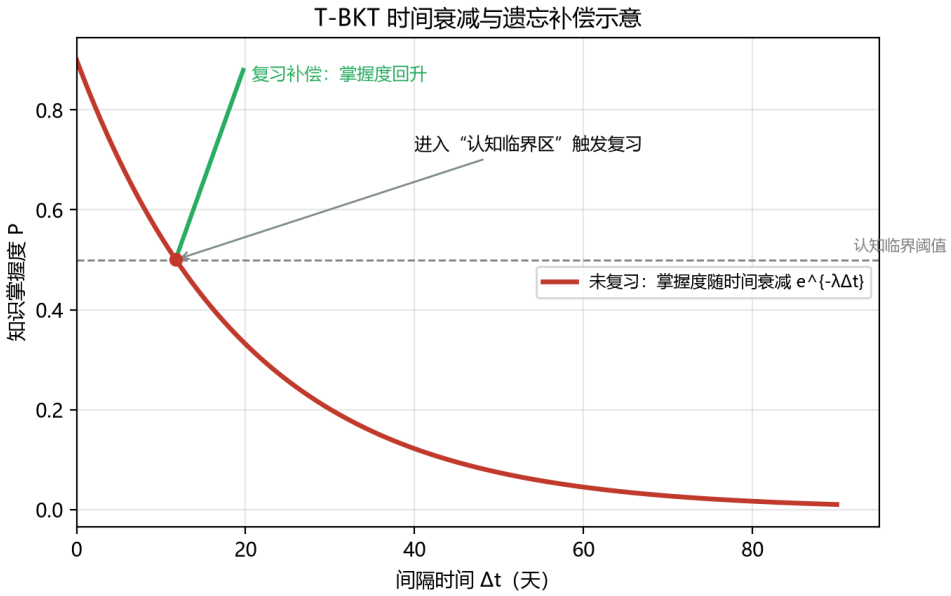


Figure 7. Schematic diagram of time-dependent mastery decay and forgetting compensation in T-BKT
图 7. T-BKT 掌握度时间衰减与遗忘补偿示意图

Table 2. Dynamic update algorithm for T-BKT proficiency assessment
表 2. T-BKT 掌握度动态更新算法

阶段	操作	数学表达
遗忘诊断	读取上次练习时间并求时间差	$\Delta t = \text{now} - \text{last_time}$
时间衰减	对历史掌握度进行折旧	$P(t) = P_0 \cdot \exp(-\lambda \cdot \Delta t)$
临界判定	与认知临界阈值比较	$P < \text{threshold} \rightarrow$ 触发复习
贝叶斯更新	结合作答结果更新后验	$P(L\|\text{answer})$ (式 2-4/2-5)
状态转移	叠加学习转移概率	$P \leftarrow P(L) + (1 - P(L)) \cdot P(T)$

4.3.2. 动态学情画像生成

T-BKT 的推理结果用于生成“动态学情图谱”：平台通过自然语言处理理解题库文档与教学大纲中实体之间的联系，自动生成具有前后依赖关系的网状知识结构，为 BKT 状态挂载提供支点。前端渲染引擎实时监听后端输出的各知识点掌握率，并依据阈值进行颜色分段着色——掌握度高于 80%渲染为绿色“已掌握”，介于 60%~80%渲染为黄色“需复习”，低于 60%或经时间衰减低于阈值则渲染为红色薄弱节点。该“红绿灯”式视觉反馈让学生可快速定位薄弱之处，并可点击薄弱节点激活相关题目，完成从“学情精准诊断”到“闭环巩固干预”的衔接。

在宏观层面，系统还将同一章节的微观知识点掌握度加权汇总投射到能力雷达图，实现从“宏观雷达定界”到“微观节点溯源”的两层次展示。如图 8 所示，系统为《天线原理》等课程生成的智能学习图谱，不仅展示了学科的整体知识结构，而且能够直观反映学生在各知识点上的掌握水平。动态图谱的设计逻辑以及交互流程是建立在对数据进行分析的基础上并结合算法实现。图谱摒弃了人工硬编码方式，采用平台题库中心的 AI 自动生成能力。系统通过自然语言处理方法理解题库文档以及各种实体之间的联系，自动生成一个有严格的前向、后向依赖关系的网状结构，其中每个小点都是一个具体的知识点。

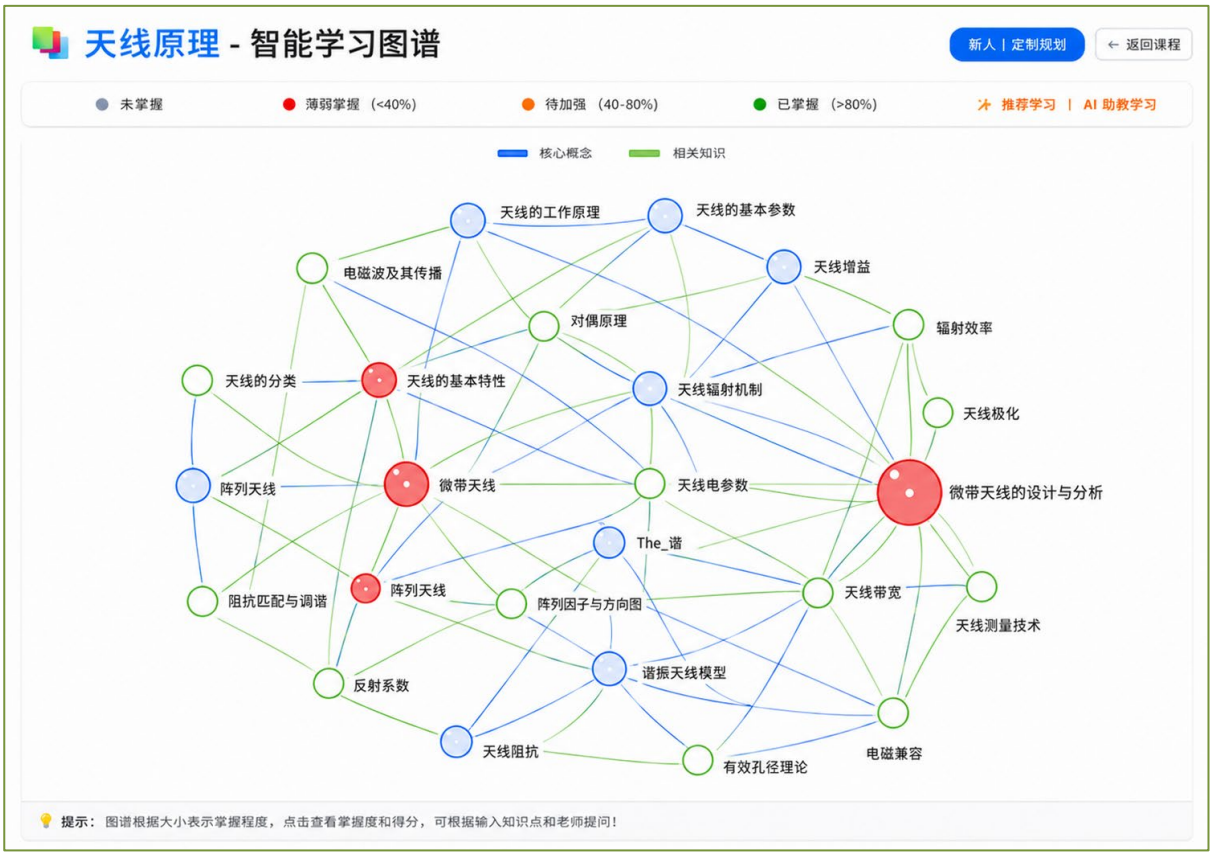


Figure 8. Intelligent learning status knowledge graph interactive interface based on T-BKT probability mapping
图 8. 基于 T-BKT 概率映射的智能学情知识图谱交互界面

4.4. T-BKT 模型验证与参数设定

4.4.1. 数据集与指标

采用公开数据集 ASSISTments 2009~2010 Skill Builder [14]和平台早期作答数据。按“学生–知识点”组织序列，计算相邻作答时间差 Δt，按学生分层 5 折交叉验证。评估指标为 AUC、Accuracy 与 RMSE (RMSE = $(1/N)\cdot\sum(y_i-\hat{y}_i)^2$ 的平方根)。

4.4.2. 预测性能对比

选取标准 BKT (无遗忘机制)、本文 T-BKT、深度知识追踪 DKT 及滑动窗口正确率基线，在同一数据划分与指标口径下对比，结果如表 3 所示。

Table 3. Comparison of predictive performance between T-BKT and the baseline model (ASSIStments 2009)
表 3. T-BKT 与基线模型预测性能对比(ASSISTments 2009)

模型	类型	AUC	说明
标准 BKT	概率/可解释	0.73	无遗忘机制，公开实测值
T-BKT (本文)	概率/可解释	0.77	含时间衰减遗忘
DKT	深度模型	0.86	深度基线，需大规模训练数据

标准 BKT 的 AUC = 0.73 为 Khajah 等在 ASSISTments 2009 的公开实测结果[15]；DKT 的 AUC = 0.86

为 Piech 等公开实测结果[16]。含遗忘机制可提升知识追踪预测能力(Nagatani 等[6])。T-BKT 与标准 BKT 同属可解释概率模型，通过引入时间衰减因子刻画遗忘过程，可解释性与冷启动适应性优于需大规模数据的 DKT。

4.4.3. 遗忘率 λ 的设定方法

T-BKT 掌握度按 $P(t)=P_0\cdot\exp(-\lambda\Delta t)$ 指数衰减，λ 为日遗忘率。设定分三步：① 经验初值——由半衰期推导，令 $P(T_{1/2})=0.5P_0$，得 $\lambda=\ln 2/T_{1/2}$，据教学经验设“掌握度减半天数”；② 数据优化——验证集网格搜索 $\lambda\in[0.005,0.5]$，以 AUC 最大为准；③ 自适应——按学科、年级、学生画像分别拟合。表 4 给出 λ 取值建议区间。

Table 4. Recommended ranges for the forgetting rate λ
表 4. 遗忘率 λ 取值建议区间

场景	半衰期 $T_{1/2}$	λ 建议区间
短期记忆型(公式、术语)	1~3 天	0.23~0.69
常规课程知识点	7~14 天	0.05~0.10
长期巩固型(章节主干)	30~60 天	0.012~0.023

4.4.4. BKT 初始参数设定

初始掌握概率 $P(L_0)$：由知识图谱先修关系映射，或由 IRT 能力冷启动预估，经逻辑函数映射到[0, 1]，新知识点取经验值 0.1~0.2。学习转移概率 $P(T)$取经验区间 0.1~0.3 或 EM 拟合；猜对概率 $P(G)$、失误概率 $P(S)$按题型设定或由 EM (Baum-Welch)从历史作答估计。数据不足时先用 IRT 预估映射初始参数，随数据积累再以 EM 二次校准，形成“先验预估–后验修正”闭环。

4.5. 关键技术细节实现

4.5.1. 知识图谱自动构建

采用“规则 + 机器学习 + 大语言模型”混合流程，六环节如下：预处理(清洗、去停用词、公式占位)→分词与词性标注(jieba + 领域词典)→命名实体识别(词典 + 规则，LTP，LLM 二次识别)→关系抽取(规则模板/依存句法 LTP-THULAC/LLM 提示抽取三元组)→实体消歧(同义词归一)→置信度入库与可视化(ECharts/D3)。各环节方法与工具见表 5。

Table 5. NLP methods and tools used in each phase of knowledge graph construction
表 5. 知识图谱构建各环节 NLP 方法与常用工具

环节	方法	常用工具/技术
分词与词性标注	词典辅助分词、词性标注	jieba、LTP、THULAC
命名实体识别	词典 + 规则，LLM 二次识别	领域词典、LTP、Qwen
关系抽取	规则匹配/依存句法/LLM 提示	LTP、THULAC、Qwen 系列
实体消歧与入库	同义词归并、三元组存储	编辑距离、SQLite/ECharts

4.5.2. IRT 参数二次校准

采用“滑动窗口 + 阈值触发”的后验校准：数量触发(某题累积作答 ≥ 30 次)、时间触发(距上次校准 ≥ 7 天)、漂移触发(参数偏差$|\Delta b|>0.2$)。对每道题维护最近 200 条有效作答的滑动窗口，在窗口内以极大似然(MLE)或贝叶斯最大后验(MAP)估计 3PL 参数(b,a,c)，对数似然 $L=\sum\left[y\cdot\ln P+(1-y)\cdot\ln(1-P)\right]$，

牛顿–拉弗森或梯度上升迭代，以冷启动预估值为初始点。校准在后台异步执行，不影响切题实时性。

4.5.3. Celery 异步出题工作流

采用 Redis(Broker)—Celery Worker—Result Backend 架构，与 Gunicorn 业务进程分离。具体流程如下：客户端发起出题→后端检测考点空缺→立即返回 HTTP 202 及 task_id 释放线程→Worker 调百炼 LLM 按 Prompt 矩阵流式生成→逻辑校验并入库→前端轮询 task_id 获取题目。任务经历 PENDING→STARTED→ SUCCESS/FAILURE 状态机；失败有限重试 3 次并记死信，以 task_id + 考点幂等键防重复生成，并发限制防 LLM 接口过载。

4.6. 大语言模型驱动的智能答疑与错因分析

在 IRT 完成难度匹配、T-BKT 完成掌握度跟踪之后，平台进一步引入大语言模型实现针对性答疑与互动式辅导。学生完成题目练习后，后端调度器自动生成多维 Prompt，将题干、标准答案、学生作答及 T-BKT 最新掌握度一并传入大模型，实现“靶向信息注入”，使 AI 可准确进行错因分析，并仅根据学生提问逐步启发引导，避免直接给出答案。当用户完成一次完整的自适应练习后，系统切换至宏观总结模式，由 LLM 生成“智能练习诊断报告”与专属 AI 导师点评，将抽象分数转化为兼具学习指导与情感支持的具体学习计划，完成从“数据分析”到“教学干预”的闭环。

5. 平台有效性的量化评估

5.1. 实验设计

招募我校信息类本科生，随机分为实验组($n = 30$，使用本自适应平台)与对照组($n = 30$，使用静态题库平台，关闭 IRT 动态切题、T-BKT 推送与 AI 讲解，仅顺序展示题目)。两组完成相同知识单元的等价学习任务，流程为“前测–学习–后测–满意度问卷”四个阶段。评价指标包括：前后测百分制成绩、系统日志记录的练习总时长(min)、首次作答正确率、李克特 5 级满意度问卷(感知有用性、感知易用性、学习投入、整体满意度、再次使用意愿 5 维度)。

5.2. 实验结果

如表 6 所示，结果表明：实验组后测成绩显著高于对照组($p < 0.001$, $d = 0.91$)；实验组以更短学习时长达成相同任务，首次作答正确率与满意度评分显著更高，说明自适应机制可有效提升学习成效与体验。两组前测无显著差异($p = 0.77$)，表明基线可比。

Table 6. Empirical research results
表 6. 实证研究结果

指标	实验组	对照组	统计量	p	效应量 d
前测成绩	62.8 ± 9.6	63.5 ± 9.2	t = −0.30	0.77	0.11
后测成绩	78.5 ± 10.3	70.7 ± 10.6	t = 3.78	<0.001	0.91
练习总时长(min)	46.2 ± 12.0	59.1 ± 14.7	t = 3.75	<0.001	0.95
首次作答正确率	0.79 ± 0.10	0.65 ± 0.12	t = 5.02	<0.001	1.26
满意度总均分	4.32 ± 0.38	3.58 ± 0.45	t = 7.10	<0.001	1.80

6. 结论

本文面向个性化教育需求，结合教育测量理论、Web 开发技术与大语言模型，设计并实现了基于 Flask

框架的智能题库与自适应学习平台。平台以 Flask 为后端、Nginx + Gunicorn 搭建高可靠运行环境，完成了从知识点整理、认知水平监测到个性化学习规划全流程。在题库建设方面，系统采用启发式文档解析与 AI 自动生成相结合的方式，并利用大语言模型进行逻辑一致性检查与 IRT 参数冷启动预估，缓解了自适应系统初始阶段数据不足的问题。在核心引擎方面，系统融合 IRT 与 BKT，依据最大似然估计与最大信息量准则实现动态选题，提出融入时间衰减因子的 T-BKT 模型补偿记忆遗忘，提高了学习状态评估的真实性与连续性。测试结果表明，平台在功能性、兼容性、并发性与安全性等方面均达到设计要求。未来可尝试引入多模态大模型，实现文本、图像、公式、语音等多源信息的统一理解，支持复杂作业批改与实验报告分析；构建包含复杂图表的多模态柔性题型矩阵；并结合联邦学习、知识图谱与强化学习等技术，对学生长期学习行为进行持续建模，实现更精准的学习路径推荐与动态干预。

基金项目

华侨大学 2025 年本科教育教学改革研究项目(HQJGYB2523)：人工智能背景下电磁信息类课程群综合改革与实践。

参考文献

[1] 杨慷, 贺文洁, 王海龙. 单参数单维度 Rasch 模型的优势与意义[J]. 心理科学, 2021, 44(6): 1491-1498.

[2] 漆书青, 戴海琦, 丁树良. 现代教育与心理测量学原理[M]. 北京: 高等教育出版社, 2002.

[3] Liang, Z., Liang, Z., Zheng, Y., Liang, B. and Zheng, L. (2021) Data Analysis and Visualization Platform Design for Batteries Using Flask-Based Python Web Service. *World Electric Vehicle Journal*, **12**, Article No. 187. https://doi.org/10.3390/wevj12040187

[4] Lord, F.M. (1952) A Theory of Test Scores. Psychometric Society.

[5] Lord, F.M. and Novick, M.R. (1968) Statistical Theories of Mental Test Scores. Addison-Wesley.

[6] Nagatani, K., Zhang, Q., Sato, M., Chen, Y., Chen, F. and Ohkuma, T. (2019) Augmenting Knowledge Tracing by Considering Forgetting Behavior. *The World Wide Web Conference*, San Francisco, 13-17 May 2019, 3101-3107. https://doi.org/10.1145/3308558.3313565

[7] 管玲玲, 李浩然. 基于 Flask 前后端分离 Web 开发的实现[J]. 电脑编程技巧与维护, 2025(6): 168-170.

[8] Eby, P.J. (2010) PEP 3333—Python Web Server Gateway Interface v1.0.1. https://peps.python.org/pep-3333/

[9] 李绘英. 基于 WSGI 规范的 Web 数据库应用系统研发[D]: [硕士学位论文]. 武汉: 武汉理工大学, 2012.

[10] 盛凯, 曾壹, 王守韦. SQLite 数据库在调度集中系统中的应用研究[J]. 铁道通信信号, 2024, 60(10): 39-43.

[11] 陈忠菊. 基于 SQLAlchemy 的研究和在数据库编程中的应用[J]. 电脑编程技巧与维护, 2015(1): 62+85.

[12] 黄沙日娜, 赵国亮. 基于 MathJax 在线考试系统的题库构建与实现[J]. 高师理科学刊, 2013, 33(5): 41-43.

[13] Baum, L.E. and Petrie, T. (1966) Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, **37**, 1554-1563. https://doi.org/10.1214/aoms/1177699147

[14] Feng, M., Heffernan, N.T. and Koedinger, K.R. (2009) Addressing the Assessment Challenge with an Online System That Tutors as It Assesses. *User Modeling and User-Adapted Interaction*, **19**, 243-266. https://doi.org/10.1007/s11257-009-9063-7

[15] Khajah, M., Lindsey, R.V. and Mozer, M.C. (2016) How Deep Is Knowledge Tracing? *Proceedings of the 9th International Conference on Educational Data Mining*, Raleigh, 29 June-2 July 2016, 94-101.

[16] Piech, C., Bassen, J., Huang, J., *et al*. (2015) Deep Knowledge Tracing. *Advances in Neural Information Processing Systems* 28 (*NIPS* 2015), Montreal, 7-12 December 2015, 505-513.