DOI:10.7544/issn1000-1239.20200848
59(1):81-104,2022

基于深度学习的知识追踪研究进展

刘铁园[1,2] 陈 威[1] 常 亮[1] 古天龙[1,3]
[1](广西可信软件重点实验室(桂林电子科技大学) 广西桂林 541004)
[2](桂林电子科技大学电子工程与自动化学院 广西桂林 541004)
[3](暨南大学信息科学与技术学院/网络安全学院 广州 510632)
(yangmail2002@guet.edu.cn)

Research Advances in the Knowledge Tracing Based on Deep Learning

Liu Tieyuan[1,2], Chen Wei[1], Chang Liang[1], and Gu Tianlong[1,3]
[1](*Guangxi Key Laboratory of Trusted Software* (*Guilin University of Electronic Technology*), *Guilin*, *Guangxi* 541004)
[2](*School of Electronic Engineering and Automation*, *Guilin University of Electronic Technology*, *Guilin*, *Guangxi* 541004)
[3](*School of Information Science and Technology/School of Cyber Security*, *Jinan University*, *Guangzhou* 510632)

Abstract Knowledge tracing is an important research direction in the field of educational data mining. The goal is to determine the degree of students' mastery of knowledge by establishing a model of students' knowledge changes over time and to mine potential learning rules from their learning trajectories. Fulfilling this goal means personalized guidance to students from the achievement of assisted education through artificial intelligence. Due to its powerful feature extraction capabilities, deep learning has been proven to significantly improve the performance of knowledge tracing models and has attracted more and more attention. Starting from the most basic deep knowledge tracing model, this paper comprehensively reviews the research progress in this field and provides both the technical improvement and an evolutionary map. The 3 main technical improvement directions have been elaborated and compared: 1) improvement of interpretable problems, 2) problems of long-term dependence, and 3) improvement for lack of learning features. At the same time, the existing models in the field have been classified, the public data sets have been sorted out, and the main areas of application are investigated for researchers. Finally, the future research direction of knowledge tracing based on deep learning is explored.

Key words education data mining; deep learning; knowledge tracing; recurrent neural network; artificial intelligence assisted education

摘 要 知识追踪是教育数据挖掘领域的一个重要研究方向,其目标是通过建立学生知识状态随时间变化的模型,来判断学生对知识的掌握程度并从学生的学习轨迹中挖掘出潜在的学习规律,从而提供个性化的指导,达到人工智能辅助教育的目的.深度学习因其强大的特征提取能力,已被证明能显著提升知识追踪模型的性能而越来越受到各方重视.以最基本的深度知识追踪模型为起点,全面回顾了该研究

收稿日期:2020-10-23;**修回日期**:2021-05-27
基金项目:国家自然科学基金项目(U1811264,61966009);广西可信软件重点实验室研究课题(KX202058);广西研究生教育创新计划项目(YCBZ2021072)
This work was supported by the National Natural Science Foundation of China (U1811264, 61966009), the Project of Guangxi Key Laboratory of Trusted Software (KX202058), and the Innovation Project of Guangxi Graduate Education (YCBZ2021072).
通信作者:古天龙(cctlgu@guet.edu.cn)

领域的研究进展,给出了该研究领域技术改进、演化脉络图,并从针对可解释问题的改进、针对长期依赖问题的改进、针对缺少学习特征问题的改进 3 个主要技术改进方向做了深入阐述和比较分析,同时对该领域中的已有模型做了归类,整理了可供研究者使用的公开数据集,考察了其主要应用,最后,对基于深度学习的知识追踪的未来研究方向进行了展望.

关键词 教育数据挖掘;深度学习;知识追踪;循环神经网络;人工智能辅助教育

中图法分类号 TP391

近年来,随着智能辅导系统(intelligent tutoring system, ITS)和大型开放式网络课程(massive open online courses, MOOCs)等在线教育平台的发展和普及,数百万的用户选择通过在线平台学习.相比传统的线下教育,在线学习系统最显著的优势在于其能保留学习者详尽的学习轨迹,提供了调查不同轨迹下学习者行为效能的条件[1].然而,在线学习平台上学生与教师人数的悬殊使人工的辅导变得不现实.如何利用在线学习系统的优势,从学生的学习轨迹中挖掘出潜在的学习规律,以提供个性化的指导,达到人工智能辅助教育的目的,成为了研究者密切关注的问题.

知识追踪(knowledge tracing, KT)是实现人工智能辅助教育的有力工具,目前已经成为了 ITS 的一个主要组成部分[2],被广泛应用于各个在线教育平台,如 edX[3]、Coursera[4] 和爱学习[5].KT 旨在建立学生知识状态随时间变化的模型,以判断学生对知识的掌握程度.通常情况下,KT 的任务可以被形式化为一个有监督的序列学习任务,给出学生的历史学习交互记录 $I_t=\{i_1,i_2,\cdots,i_t\}$,通过预设的模型从中提取出学生隐式的知识状态,并追踪其随时间的变化.学习交互通常表示为一个题目-答案元组 $i_t=(q_t,a_t)$,意为学生在时刻 t 回答了问题 q_t,a_t 则指示了回答的情况.由于很难直接衡量学习者的实际学习状态,现有的 KT 模型通常采用一种替代解决方案,使模型预测下一个题目答对的概率 $P(a_{t+1}=correct\,|\,q_{t+1},I_t)$.

BKT(Bayesian knowledge tracing)模型[6]是目前最流行的 KT 模型之一,BKT 将学习者的潜在知识状态建模为一组二元变量,代表是否掌握某个知识成分(knowledge component, KC).KC 可以被泛化地理解为知识概念、原理、事实或者技能.对于一个具体的题目 q_t,KC 可以视作答对 q_t 所必须掌握的知识.在单 KC 模型中(即一个题目对应一个 KC),题目与 KC 可以视为是等价的.

在每一次学习交互后,BKT 使用隐 Markov 模型(hidden Markov model, HMM)更新这些二元变量的概率.在提出后的 20 年来,BKT 一直被视为 KT 领域的首选方法,其他机器学习模型,如 BKT 的变体[7]、逻辑回归的变体[8] 以及项目反应理论[9](item response theory, IRT)与 BKT 的性能差异都很小[10].

虽然 BKT 在 KT 领域取得了很大成功,但是其本身也存在着很大的问题.首先,变量与 KC 之间的对应是模糊的,无法做到一一对应,且二元变量的设置不符合现实中的学习过程[11].其次,对每个 KC 分开建模的方式使 BKT 无法捕捉不同 KC 间的关系,也丧失了对未定义 KC 和复杂 KC 的建模能力.

深度学习以其强大的特征提取能力引起了研究者的广泛关注,许多研究者将其应用到 KT 领域,称为基于深度学习的知识追踪(deep learning based knowledge tracing, DLKT).相对于传统的机器学习模型,DLKT 不需要人工标注的 KC 信息,且能够捕捉到更复杂的学生知识表征,还可以发现并利用 KC 之间的关联信息.目前,对 DLKT 的研究已经成为了 KT 领域的一大研究热点.

DLKT 的开创性工作深度知识追踪(deep knowledge tracing, DKT)模型[11]于 2015 年提出,在 2015 年后,涉及到 KT 领域的代表性综述性论文有文献[1,12-16].其中,文献[1]从知识点、学习者和数据 3 方面总结了 KT 模型在教育领域的应用.文献[12]侧重于讨论特定场景下模型的选择问题.文献[13]将学习者模型分为知识状态、认知行为、情感、综合 4 类并做了详细阐述,KT 模型属于其中的知识状态模型.文献[14-16]对目前的 KT 模型从教育角色、教育过程等角度进行了梳理和分析比较,但侧重在机器学习,且其中所涉及的模型较少,仅介绍了几个具有代表性的模型.总体来说,这些文献涉及到 DLKT 模型的内容较少,没有聚焦于 DLKT 领域.

本文着眼于 DLKT 领域的相关研究,对该领域的技术演化和最新研究进展进行了系统性的调研与梳理.

1 基于深度学习的知识追踪 DLKT

1.1 DLKT 相关研究概览

我们查阅了近 5 年半(2015-01—2020-06)DLKT 的相关论文.其中,中文文献来源于中国知网(CNKI),英文文献来自 IEEE Electronic Library, Elsevier Science, EI Compendex, Web of Science, Springer Link, ACM Digital Library 等数据库,并使用学术搜索引擎(谷歌学术、百度学术、必应学术)查漏补缺.在检索时,中文使用"知识追踪"和"深度知识追踪"作为关键词,英文使用"knowledge tracing"和"deep knowledge tracing"为关键词.对获得的相关文献进行分析整理,去掉不相关的和没有使用深度学习方法的文献,共筛选出 69 篇文献,图 1 给出了这些文献的发表情况.

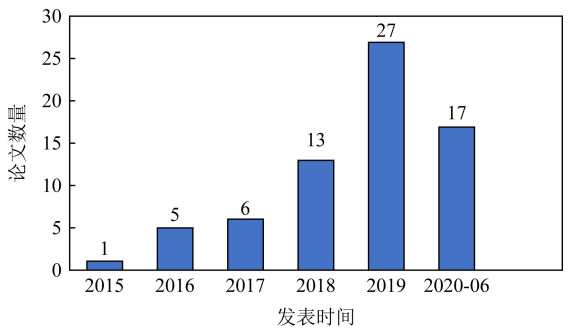

Fig. 1 Publication of DLKT related literatures in recent five and a half years

图 1 近 5 年半 DLKT 相关文献发表情况

可以看出,近 5 年半 DLKT 相关文献在国内外发文量呈明显上升趋势,在新冠疫情导致许多会议延期的前提下,2020 年上半年也有 17 篇论文发表.

发表仅 5 年,DLKT 的开创性工作 DKT 就有 433 次引用(数据来源:谷歌学术),相关研究在 NIPS, AAAI, SIGIR, WWW 等顶会上均有报道,可见该领域研究被学界的认可和接受.

1.2 符号定义

在本节,我们给出文中常见符号的定义(其他非常见符号以文中说明为准),表 1 展示了每种符号的代表含义.除特殊说明外,本文使用的符号皆以表 1 为准.我们使用 k_t 表示知识成分 KC,q_t 代表题目,a_t 代表对应的回答,下标 t 指示了时刻(如无必要,则省略).两者的组合(q_t,a_t)构成了一个学习交互 x_t.通常情况下,KC 与题目存在对应关系,两者可以相互转化,因此,学习交互同样可以表示为(k_t,a_t),根据具体的模型选择.在时刻 t,模型的预测值 y_t 表示下一时刻回答正确的概率.使用 L 表示损失函数,L_{BCE}特指二元交叉熵(binary cross entropy, BCE)函数.使用大写的粗体字母(如 $\boldsymbol{W}$)表示矩阵,使用小写的粗体字母(如 $\boldsymbol{b}$)表示向量.特别地,$\boldsymbol{W}$ 和 $\boldsymbol{b}$ 特指权重矩阵和偏置向量.

Table 1 Symbol Definition

表 1 符号定义

符号	含义
$t\in\{t_1,t_2,\cdots,t_T\}$	时刻,共 T 个时刻
k	KC(知识成分)
q	题目
$\boldsymbol{e}$	题目的嵌入
a	回答
$i=(q,a)$	学习交互
$\boldsymbol{x}$	学习交互的嵌入
p_t	时刻 $t+1$ 题目答对的预测概率
$\boldsymbol{y}_t=(p_1,p_2,\cdots,p_t)$	多个预测值组成的向量
$\boldsymbol{W}$	权重矩阵
$\boldsymbol{b}$	偏置
L	损失函数
L_{BCE}	二元交叉熵函数
$\oplus$	向量的连接操作

1.3 DLKT 基本模型

Piech 等人[11]提出的 DKT 模型是 DLKT 领域的开创性工作,也是 DLKT 领域的基本模型.DKT 的结构如图 2 所示,其以循环神经网络(recurrent neural network, RNN)为基础结构.RNN 是一种具有记忆性的序列模型,序列结构使其符合学习中的近因效应并保留了学习轨迹信息[17].这种特性使 RNN(包括长短期记忆网络[18](long short term memory, LSTM)和门控循环网络[19](gated recurrent unit, GRU)等变体成为了 DLKT 领域使用最广泛的模型.DKT 以学生的学习交互记录$\{i_1,i_2,\cdots,i_t\}$为输入,通过 one-hot 编码或压缩感知[20](compress sensing),i_t 被转化为向量输入模型.在 DKT 中,RNN 的隐藏状态 $\boldsymbol{h}_t$ 被解释为学生的知识状态,$\boldsymbol{h}_t$ 被进一步通过一个 $Sigmoid$ 激活的线性层得到预测结果 $\boldsymbol{y}_t$.$\boldsymbol{y}_t$ 的长度等于题目数量,其每个元素代表学生正确回答对应问题的预测概率.具体的计算过程为

$$\begin{aligned}\boldsymbol{h}_t&=\tanh(\boldsymbol{W}_{hx}\boldsymbol{x}_t+\boldsymbol{W}_{hh}\boldsymbol{h}_{t-1}+\boldsymbol{b}_h),\\ \boldsymbol{y}_t&=Sigmoid(\boldsymbol{W}_{yh}\boldsymbol{h}_t+\boldsymbol{b}_y).\end{aligned}\tag{1}$$

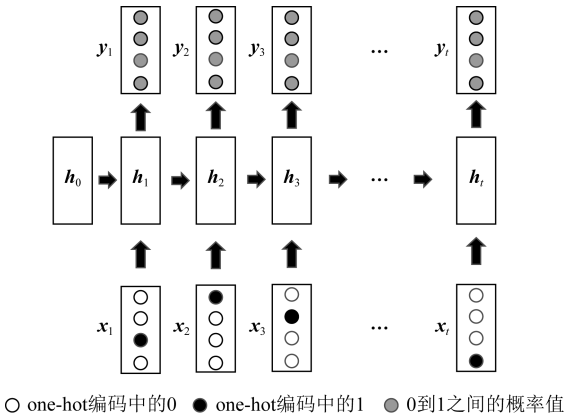

Fig. 2 Architecture for DKT model

图 2 DKT 模型结构图

DKT 的目标函数是观测序列的非负对数似然函数,用 L_{BCE} 表示二元交叉熵(BCE),则学生的损失值为

$$L=\sum_{t=t_1}^{t_T}L_{\mathrm{BCE}}(p_t,a_{t+1}). \tag{2}$$

相对于以 BKT 为代表的传统机器学习模型,DKT 不需要人工标注的数据就有更好的表现(AUC 提高了 20%[21]),且能够捕捉并利用更深层次的学生知识表征[22-23],这使其非常适合以学习为中心的教学评估系统[24].

2 DKT 的改进方法

尽管 DKT 的预测性能优于现有的经典方法,但由于它在教育应用中的实用性还有待提高,而被其他少量学者所批判[25-29].这主要是因为隐藏状态 $\boldsymbol{h}_t$ 在本质上很难被解释为知识状态,而且 DKT 模型没有对知识交互进行深入分析[30],导致其可解释性很差.

RNN 存在长期依赖问题,其变体(如 LSTM 和 GRU)仅仅提高了序列学习的容量(对于 LSTM 来说,容量在 200 左右[31]),但并没有解决问题.DKT 中以 RNN 为基础,因此其同样存在长期依赖问题.长期依赖问题使 DKT 无法利用长序列的输入[32],且会导致重构错误和波动准则[33-35].

在 DKT 中,模型的输入为 one-hot 编码的学生交互序列,而仅用交互序列作为输入浪费了在线平台所保留的丰富学习轨迹信息,且输入的学习特征太少也会影响模型的表现.

可解释性差、长期依赖问题和学习特征少是 DKT 模型最显著的 3 个问题,许多研究者致力于对其进行扩展和改进,以解决这些问题.我们将各种改进方法梳理为图 3 所示:

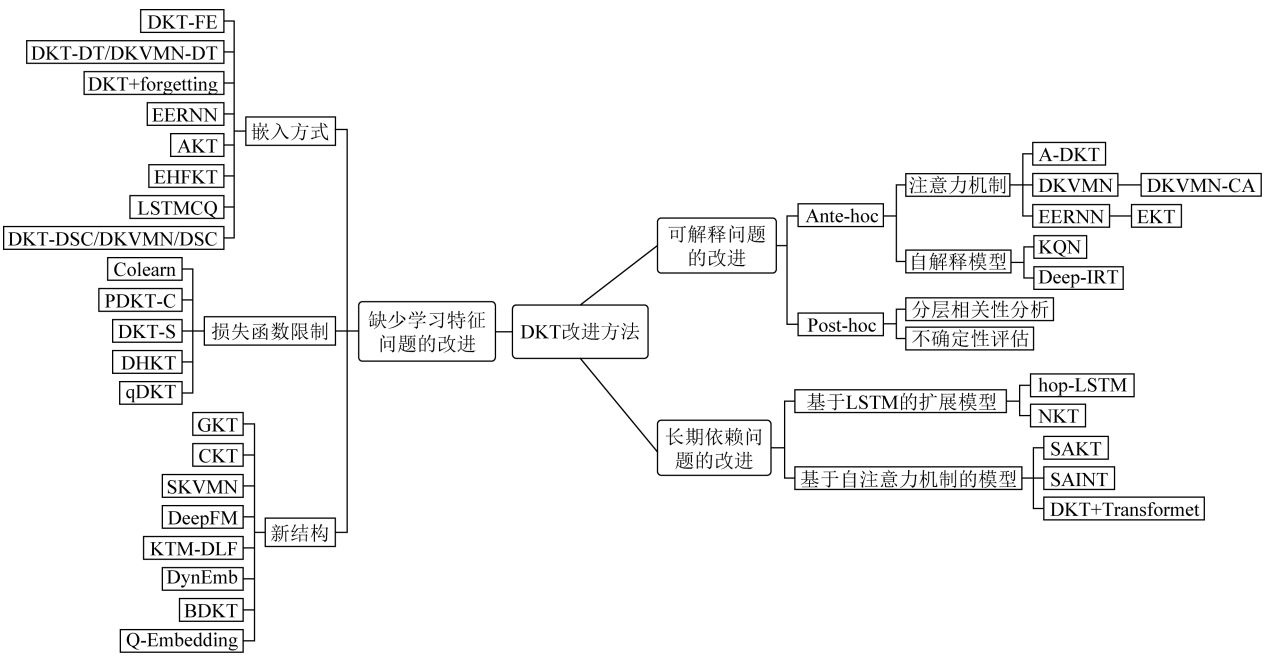

Fig. 3 DKT improved method venation diagram

图 3 DKT 改进方法脉络图

2.1 针对可解释性问题的改进

根据使用方法的不同,针对可解释性问题的改进可以进一步细分为 Ante-hoc 可解释性方法和 Post-hoc 可解释性[36]方法.

2.1.1 Ante-hoc可解释性方法

Ante-hoc可解释性指模型本身内置可解释性.对于复杂的模型,可以通过构建将可解释性直接结合到具体的模型结构中的学习模型来实现模型的内置可解释性,在深度学习中,一种有效的可解释性模块就是注意力机制.此外,Ante-hoc可解释性还可以通过采用结构简单、易于理解的自解释模型实现[36].

1) 引入注意力机制

① A-DKT(attention-DKT)

Liu等人[37]在模型A-DKT中使用了Jaccard系数计算KC之间的注意力权重.设 k_a,k_b 为不同的KC,两者间的权重值为

$$w_{a,b}=\frac{|k_a \cap k_b|}{|k_a \cup k_b|}=\frac{|k_a \cap k_b|}{|k_a|+|k_b|-|k_a \cap k_b|}. \quad (3)$$

在时刻 t,计算当前KC与之前所有KC的注意力权重,然后相加,得到总的注意力值 w_t:

$$w_t=\sum_{i=0}^{t-1} j_{i,t}. \quad (4)$$

最后,结合LSTM与注意力值得到预测结果:

$$\boldsymbol{y}_t=Sigmoid(\boldsymbol{W}_{yh}\boldsymbol{h}_t+\boldsymbol{b}_y+w_t). \quad (5)$$

② DKVMN(dynamic key-value memory network)

Zhang等人[38]受到MANN(memory-augmented neural network)的启发,提出了DKVMN模型.DKVMN使用矩阵 $\boldsymbol{M}^{key}$ 存储KC,$\boldsymbol{M}^{key}$ 的每一列代表一个KC;使用矩阵 $\boldsymbol{M}^{value}$ 存储知识状态,$\boldsymbol{M}^{value}$ 的每一列表示 $\boldsymbol{M}^{key}$ 中对应KC的掌握程度.DKVMN的结构如所图4所示.

注意力机制主要体现在 $\boldsymbol{w}_t$ 的计算上:在时刻 t,将题目的嵌入向量 $\boldsymbol{e}_t$ 与每一个KC(即 $\boldsymbol{M}^{key}$ 的每一列)作内积,经过一个激活函数后得到 $\boldsymbol{w}_t$.$\boldsymbol{w}_t(i)$代表题目 q_t 与KC k_i 之间的相关性,即注意力权重:

$$\boldsymbol{w}_t(i)=Softmax(\boldsymbol{e}_t\boldsymbol{M}_t^{key}(i)). \quad (6)$$

根据注意力权重计算学生对题目 q_t 的掌握程度 $\boldsymbol{r}_t$:

$$\boldsymbol{r}_t=\sum_i \boldsymbol{w}_t(i)\boldsymbol{M}_t^{value}(i). \quad (7)$$

题目的嵌入向量 $\boldsymbol{e}_t$ 隐式地包含了难度信息,将其与 $\boldsymbol{r}_t$ 连接,得到的 $\boldsymbol{f}_t$ 同时包含难度和掌握程度信息:

$$\boldsymbol{f}_t=\tanh(\boldsymbol{W}_1^{\mathrm{T}}[\boldsymbol{r}_t \oplus \boldsymbol{e}_t]+\boldsymbol{b}_1). \quad (8)$$

最后,$\boldsymbol{f}_t$ 通过一个 $Sigmoid$ 激活的线性层得到学生表现的预测值 $\boldsymbol{y}_t$:

$$\boldsymbol{y}_t=Sigmoid(\boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{f}_t+\boldsymbol{b}_2). \quad (9)$$

③ DKVMN-CA(concept-aware DKVMN)

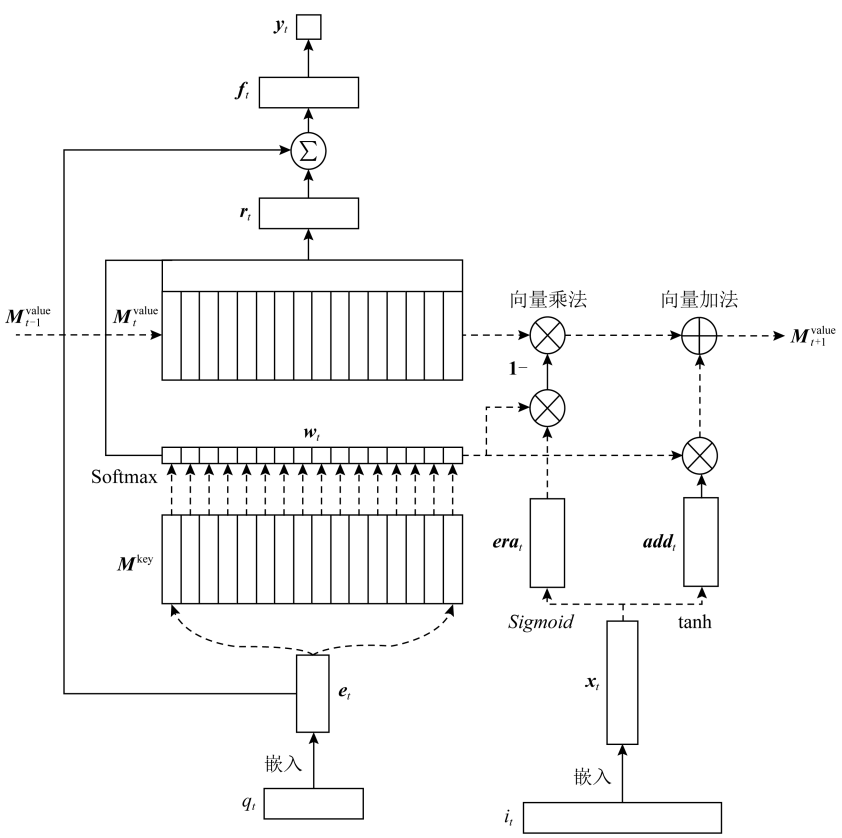

Fig. 4 Architecture for DKVMN model

图4 DKVMN模型结构图

Ai 等人[39]进一步扩展了 DKVMN 模型,使其支持了人工标注的 KC 信息.即 $\boldsymbol{M}^{key}$ 中的 KC 由人工定义,而不是机器生成.

④ EERNN(exercise-enhanced RNN)

Su 等人[40]在其模型 EERNN 中使用了余弦相似度计算隐藏状态之间的注意力权重 $\boldsymbol{w}_t$:

$$\boldsymbol{w}_t=\sum_{i=0}^{t}\cos(\boldsymbol{e}_t,\boldsymbol{e}_i)\cdot\boldsymbol{h}_i. \tag{10}$$

需要注意的是,余弦相似度计算的并不是隐藏状态之间的相似性,而是题目之间的相似性.EERNN 的预测过程为:

$$\begin{aligned}\boldsymbol{r}_t&=ReLU(\boldsymbol{W}_1\cdot[\boldsymbol{w}_{t+1}\oplus\boldsymbol{e}_{t+1}]+\boldsymbol{b}_1),\\ \boldsymbol{y}_t&=Sigmoid(\boldsymbol{W}_2\cdot\boldsymbol{r}_t+\boldsymbol{b}_2).\end{aligned} \tag{11}$$

⑤ EKT(exercise-aware KT)

Huang 等人[41]提出的 EKT 模型综合了 EERNN 模型与 DKVMN 模型,因此拥有双重注意力模块.其主要计算过程与 EERNN 和 DKVMN 模型相同,此处不再赘述.

2) 自解释模型

① KQN(knowledge query network)

Lee 等人[42]用向量点积来模拟知识状态与 KC 的相互作用,提出了 KQN 模型.假设知识状态与 KC 都为 2 维向量,如图 5 所示,在知识状态由 $\boldsymbol{v}_2$ 转变到 $\boldsymbol{v}_3$ 的过程中,学生对 KC $\boldsymbol{k}_1$ 的掌握程度由 $\boldsymbol{v}_2\cdot\boldsymbol{k}_1=1$ 增长到了 $\boldsymbol{v}_3\cdot\boldsymbol{k}_1=2$.这种向量化的表示与运算使 KQN 具有直观性和可解释性.

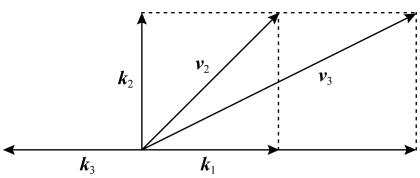

Fig. 5 An example of KQN

图 5 KQN 的一个例子

在 KQN 中,学生知识状态向量 $\boldsymbol{v}_t$ 由 RNN 的隐藏状态 $\boldsymbol{h}_t$ 经过一个全连接层得到:

$$\boldsymbol{v}_t=\boldsymbol{W}_{vh}\cdot\boldsymbol{h}_t+\boldsymbol{b}_v. \tag{12}$$

在时刻 t,one-hot 编码的 KC $\boldsymbol{k}_{t+1}$ 通过一个多层感知机和 L2 正则化.生成新的嵌入表示:

$$\begin{aligned}\boldsymbol{o}_{t+1}&=ReLU(\boldsymbol{W}_1\cdot ReLU(\boldsymbol{W}_0\cdot\boldsymbol{k}_{t+1}+\boldsymbol{b}_0)+\boldsymbol{b}_1),\\ \boldsymbol{k}_{t+1}&=L_2\text{-normalize}(\boldsymbol{o}_{t+1}).\end{aligned} \tag{13}$$

KQN 用 $\boldsymbol{v}_t$ 与 $\boldsymbol{v}_{t+1}$ 的内积定义知识交互,并作出预测:

$$\boldsymbol{y}_t=Sigmoid(\boldsymbol{v}_t\cdot\boldsymbol{k}_{t+1}). \tag{14}$$

② Deep-IRT(deep item response theory)

Yeung 等人[43]综合了 IRT 理论[9]与 DKVMN 模型,提出了 Deep-IRT 模型.Deep-IRT 通过 2 个全连接神经网络,利用 DKVMN 中的 $\boldsymbol{f}_t$ 与 $\boldsymbol{e}_t$(式(8))分别对学生 n 的能力 $\boldsymbol{\theta}_{t,n}$ 和题目难度 $\boldsymbol{\beta}_t$ 建模:

$$\begin{aligned}\boldsymbol{\theta}_{t,n}&=\tanh(\boldsymbol{W}_\theta\boldsymbol{f}_t+\boldsymbol{b}_\theta),\\ \boldsymbol{\beta}_t&=\tanh(\boldsymbol{W}_\beta\boldsymbol{e}_t+\boldsymbol{b}_\beta).\end{aligned} \tag{15}$$

最后,通过 IRT 函数得出预测结果:

$$\boldsymbol{y}_t=Sigmoid(\boldsymbol{\theta}_{t,n}-\boldsymbol{\beta}_t). \tag{16}$$

2.1.2 Post-hoc 可解释性方法

Post-hoc 可解释性也称事后可解释性,发生在模型训练之后,旨在利用解释方法或构建解释模型,解释学习模型的工作机制、决策行为和决策依据[36].在 DLKT 领域,主要有 LRP 和不确定性评估方法.

1) LRP(layer-wise relevance propagation)

Lu 等人[44]应用分层相关性传播方法,通过将相关性从模型的输出层反向传播到输入层,来解释基于 RNN 的 DLKT 模型.LRP 的核心是利用反向传播将高层的相关性分值递归地传播到低层直至传播到输入层.具体来说,将 RNN 不同单元之间的连接分为加权连接和乘法连接并分别定义两者相关性的传播.如图 6 所示,通过计算 2 种类型连接的反向传播相关性,就可以对基于 RNN 的 DLKT 模型进行解释.

$$R_{\boldsymbol{h}_t}=\frac{\boldsymbol{W}_{yh}\boldsymbol{h}_t}{\boldsymbol{W}_{yh}\boldsymbol{h}_t+\boldsymbol{b}_y+\varepsilon\cdot\text{sgn}(\boldsymbol{W}_{yh}\boldsymbol{h}_t+\boldsymbol{b}_y)}\cdot R^d_{\boldsymbol{y}_t}, \tag{17}$$

$$R_{\boldsymbol{C}_t}=R_{\boldsymbol{h}_t}, \tag{18}$$

$$R_{\boldsymbol{f}_t\boldsymbol{C}_{t-1}}=\frac{\boldsymbol{f}_t\boldsymbol{C}_{t-1}}{\boldsymbol{C}_t+\varepsilon\cdot\text{sgn}(\boldsymbol{C}_t)}\cdot R_{\boldsymbol{C}_t}, \tag{19}$$

$$R_{\boldsymbol{C}_{t-1}}=R_{\boldsymbol{f}_t\boldsymbol{C}_{t-1}}, \tag{20}$$

$$R_{\boldsymbol{i}_t\widetilde{\boldsymbol{C}}_t}=\frac{\boldsymbol{i}_t\widetilde{\boldsymbol{C}}_t}{\boldsymbol{C}_t+\varepsilon\cdot\text{sgn}(\boldsymbol{C}_t)}\cdot R_{\boldsymbol{C}_t}, \tag{21}$$

$$R_{\widetilde{\boldsymbol{C}}_t}=R_{\boldsymbol{i}_t\widetilde{\boldsymbol{C}}_t}. \tag{22}$$

其中,$R^d_{\boldsymbol{y}_t}$ 为预测值 $\boldsymbol{y}_t$ 第 d 维的值.$\text{sgn}()\in\{-1,1\}$ 指示正负性,ε 为优化计算超参数,R 表示相关性.$\boldsymbol{C},\widetilde{\boldsymbol{C}},\boldsymbol{i}$ 为 LSTM 计算过程的中间参数.时刻 t 的输入 $\boldsymbol{x}_t$ 的相关性 $R_{\boldsymbol{x}_t}$ 可以推导为

$$\begin{aligned}R_{\boldsymbol{x}_t}=\Big\{\boldsymbol{W}_{cx}\boldsymbol{x}_t\Big\}\Big/\Big\{&\boldsymbol{W}_{ch}\boldsymbol{h}_{t-1}+\boldsymbol{W}_{cx}\boldsymbol{x}_t+\boldsymbol{b}_c+\\ &\varepsilon\cdot\text{sgn}(\boldsymbol{W}_{ch}\boldsymbol{h}_{t-1}+\boldsymbol{W}_{cx}\boldsymbol{x}_t+\boldsymbol{b}_c)\Big\}\cdot R_{\widetilde{\boldsymbol{C}}_t}.\end{aligned} \tag{23}$$

2) 不确定性评估

Ding 等人[45]和 Hu 等人[46]都使用不确定性评估方法解决可解释性差的问题,这里主要阐述前者

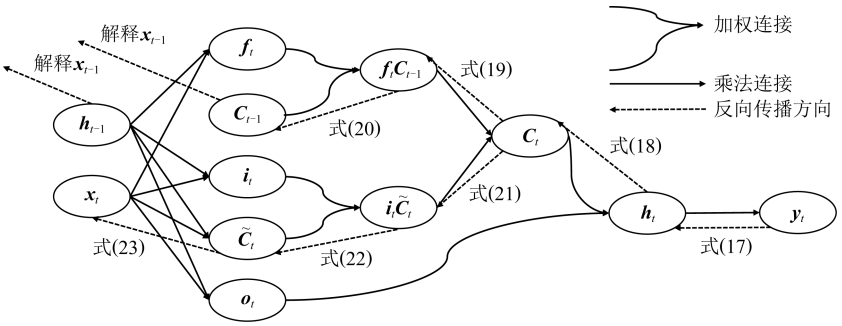

Fig. 6 The feedforward prediction path and the backpropagation path for interpreting its prediction results

图 6 前馈预测路径及解释预测结果的反向传播路径

的方法:通过为模型的每一个预测值 $\boldsymbol{y}$ 提供一个不确定性评分 $\boldsymbol{u}$(由模型输出),以减轻预测过程中的不透明性,增加模型的可解释性.不缺定性分为 2 种:

① 随机不确定性(或数据不确定性).由随机事件或观测中的固有噪声导致.由于神经网络的固定权值,因此输入 $\boldsymbol{x}$ 的噪声会传播给输出 $\boldsymbol{y}$.用 D 表示噪声,则模型的输出为

$$\begin{aligned}\hat{\boldsymbol{y}}_t&=\boldsymbol{y}_t^D+\sigma^D\varepsilon_t,\\ \varepsilon_t&\sim N(0,\boldsymbol{I}),\end{aligned}\tag{24}$$

其中,σ^D 表示 $\boldsymbol{y}_t^D$ 的偏差值,由模型输出.由于 $\hat{\boldsymbol{y}}$ 的期望很难确定,因此使用蒙特卡洛方法近似,最大化期望对数似然相当于最小化二元交叉熵,最终的损失函数为

$$L=L_{\mathrm{BCE}}(\boldsymbol{y},\hat{\boldsymbol{y}})+\alpha L_1+\beta L_2,\tag{25}$$

其中,L_1,L_2 均为优化计算的正则项,由超参数 α,β 平衡权重.

② 认知不确定性(或模型不确定性).由数据不足导致,这种不确定性可以通过使用更多的数据来降低.针对深度学习模型的不确定性建模的贝叶斯方法是假设这些模型的权重不是固定的,而是从分布中取样的.最后的预测是通过综合所有可能的权重得到的.对于分类任务,预测正确的概率可以近似为

$$p_c=\frac{1}{S}\sum_{t=t_1}^{t_T}Softmax(\boldsymbol{y}_t^D).\tag{26}$$

其中,c 表示预测正确的题目,S 表示取样总数.不确定性可以用熵概括:

$$H(p_c)=-\sum_c p_c\times\log p_c.\tag{27}$$

2.1.3 小 结

本节详细介绍了 DLKT 领域针对可解释性问题的改进方法,主要分为 Ante-hoc 和 Post-hoc.前者对问题的解决程度稍显不足,无论是注意力机制还是自解释模型都只能提高某一部分的可解释性.而后者试图解释整个模型,但其方法本身对部分条件要求较高,如 LRP 需要预测函数有较高的梯度.

2.2 针对长期依赖问题的改进

针对长期依赖问题的改进模型中,根据方法的不同,可以分为 LSTM 的扩展模型和基于自注意力机制的模型.

2.2.1 基于 LSTM 的扩展模型

1) Hop-LSTM

Abdelrahman 等人[47]使用了 Hop-LSTM 来进一步扩大 LSTM 的序列学习容量.Hop-LSTM 是一种改进的 LSTM,可以根据隐藏单元之间的相关性进行跳跃连接.如图 7 所示,Hop-LSTM 所基于的顺序关系为 $q_1\leftarrow q_4$,$q_2\leftarrow q_3$,$q_5\leftarrow q_3$,…,$q_4\leftarrow q_t$,其中 $\boldsymbol{f}_t$ 的计算参考式(8).可以看到,LSTM 单元依据顺序关系跳跃连接.

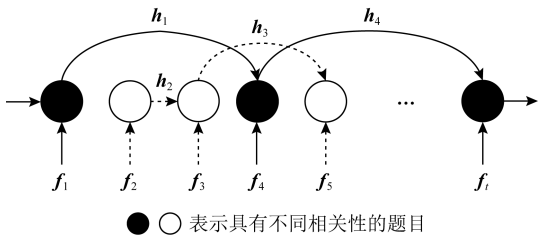

Fig. 7 Hop-LSTM schematic diagram

图 7 Hop-LSTM 示意图

2) NKT(neural KT)

有研究指出,RNN 的层层叠加可以减轻 LSTM 中长期依赖关系的学习困难[48].基于这个结论,Sha 等人[49]提出了 NKT 模型,设计了一种 2 层堆叠的 LSTM,并使用残差连接减小训练难度.实验证明,这种叠加的 LSTM 可以有效扩大序列学习的容量.

2.2.2 基于 Transformer 的模型

自注意力机制(self-attention mechanism)是注意力机制的一个变体,由 Lin 等人[50]提出.后来,

Vaswani 等人[51]使用自注意力机制代替了 RNN 搭建了整个模型框架,提出了 Transformer 模型.由于 Transformer 模型不依赖 RNN 框架,因此不存在长期依赖问题.Transformer 模型最初用于机器翻译任务,取得了很好的效果.后来,有研究者将其用于知识追踪领域,获得了媲美基于 RNN 的 DLKT 模型的效果,且不存在长序列依赖问题.

1) SAKT(self attention KT)

Pandey 等人[52]率先在知识追踪领域使用了 Transformer 模型,提出了 SAKT 模型,其结构如图 8 所示:

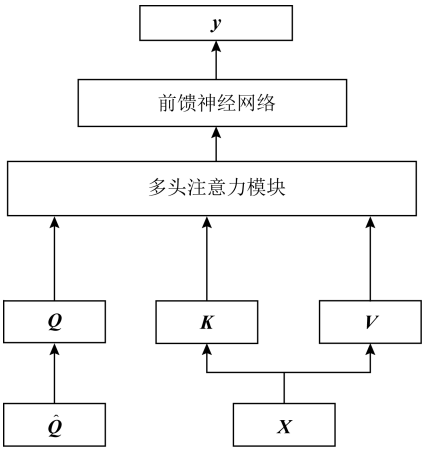

Fig. 8 Architecture for SAKT model

图 8 SAKT 模型结构图

在 Transformer 中,计算注意力所使用的 $\boldsymbol{Q}$,$\boldsymbol{K}$,$\boldsymbol{V}$ 这 3 个参数由输入序列乘以不同的权重矩阵所得.而在 SAKT 中,$\boldsymbol{Q}$ 和 $\boldsymbol{K}$,$\boldsymbol{V}$ 分别由题目的嵌入序列和交互的嵌入序列投影得到.

$$\boldsymbol{Q}=\hat{\boldsymbol{Q}}\boldsymbol{W}^{\boldsymbol{Q}},\boldsymbol{K}=\hat{\boldsymbol{X}}\boldsymbol{W}^{\boldsymbol{K}},\boldsymbol{V}=\hat{\boldsymbol{X}}\boldsymbol{W}^{\boldsymbol{V}}, \tag{28}$$

其中,$\hat{\boldsymbol{Q}}$ 为题目嵌入矩阵,$\hat{\boldsymbol{X}}=\boldsymbol{X}\oplus\boldsymbol{P}$,由交互嵌入矩阵 $\boldsymbol{X}$ 与位置嵌入矩阵 $\boldsymbol{P}$ 连接得到.SAKT 不是时序模型,所以需要位置嵌入以保留输入序列的位置信息.

注意力被计算 h 次,使得模型能够在不同的表示子空间中学习相关信息,并将 h 次的结果连接,称为多头注意力(multi-head attention, MHA):

$$\boldsymbol{H}_{\mathrm{MHA}}=[\boldsymbol{H}_1\oplus\boldsymbol{H}_2\oplus\cdots\oplus\boldsymbol{H}_h]\boldsymbol{W}^{\mathrm{MHA}},$$
$$\boldsymbol{H}_i=Attention(\boldsymbol{Q}_i,\boldsymbol{K}_i,\boldsymbol{V}_i)=Softmax\left(\frac{\boldsymbol{Q}_i\boldsymbol{K}_i^{\mathrm{T}}}{\sqrt{d}}\right)\boldsymbol{V}_i, \tag{29}$$

其中,d 为嵌入向量的维度.最终,通过一个前馈神经网络层和一个 *Sigmoid* 激活的线性层得到预测结果:

$$\boldsymbol{F}=ReLU(\boldsymbol{H}_{\mathrm{MHA}}\boldsymbol{W}_1+\boldsymbol{b}_1)\boldsymbol{W}_2+\boldsymbol{b}_2,$$
$$\boldsymbol{y}_t=Sigmoid(\boldsymbol{F}_t\boldsymbol{W}+\boldsymbol{b}). \tag{30}$$

2) SAINT(separated self-attention neural KT)

Choi 等人[53]认为 SAKT 模型的注意力层太浅,且 $\boldsymbol{Q}$,$\boldsymbol{K}$,$\boldsymbol{V}$ 的计算方法缺乏经验支持,并提出了 SAINT 模型以解决这 2 个问题.

如图 9 所示,SAINT 主要由编码器(encoder)和解码器(decoder)两部分组成.编码器的输入为题目的嵌入矩阵 $\hat{\boldsymbol{Q}}=(\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_t)$,$\boldsymbol{q}$ 由题目嵌入和位置嵌入连接得到,解码器以编码器的输出以及回答嵌入矩阵 $\boldsymbol{A}=(\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_t)$ 为输入.编码器与解码器都是多头注意力模块和前馈神经网络的组合,不同之处在于解码器叠加了 2 个注意力模块,以进一步捕捉题目与回答之间的复杂关系.

$$\boldsymbol{F}_{\mathrm{E}}=Encoder(\hat{\boldsymbol{Q}}),$$
$$\boldsymbol{F}_{\mathrm{D}}=Decoder(\boldsymbol{A},\boldsymbol{F}_{\mathrm{E}}). \tag{31}$$

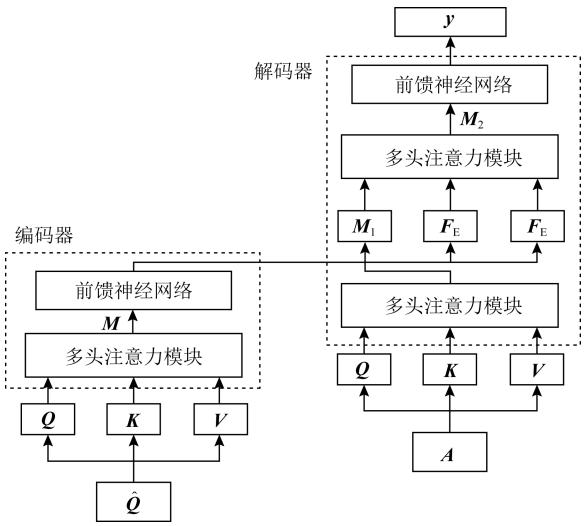

Fig. 9 Architecture for SAINT model

图 9 SAINT 模型结构图

编(解)码器的内部实现与 SAKT 相同,最终,解码器的输出 $\boldsymbol{F}_{\mathrm{D}}$ 经过一个 *Sigmoid* 激活的线性层得到最终的预测结果.

在 SAINT 中,编(解)码器共叠加了 4 层,实验结果证明,多次叠加的注意力层可以有效增大 AUC 的面积,获得更好的预测性能.

3) DKT+Transformer

Pu 等人[54]改进了 Transformer 的结构,在其中加入了题目的结构信息和答题的时间信息.改进后的 Transformer 模型结构与 SAINT 大致相同,都由编码器和解码器组成,主要区别在于自注意力的计算.

$$\boldsymbol{F}_{\mathrm{E}}=Encoder(\hat{\boldsymbol{X}},\boldsymbol{t}'),$$
$$\boldsymbol{F}_{\mathrm{D}}=Decoder(\hat{\boldsymbol{Q}},\boldsymbol{t}',\boldsymbol{F}_{\mathrm{E}}),\tag{32}$$

其中,$\boldsymbol{t}'=(t_1,t_2,\cdots,t_n)$为时间戳信息,注意力机制的计算为:

$$\boldsymbol{H}_{ij}=\frac{\boldsymbol{Q}_j\boldsymbol{K}_i+\boldsymbol{b}\times(t_j-t_i)}{\sqrt{d}}\boldsymbol{V}_i,\tag{33}$$

其中,$\boldsymbol{b}\times(t_j-t_i)$表示时间间隔偏置,目的是通过$\boldsymbol{x}_j$与$\boldsymbol{x}_i$之间的间隔时间来调整注意力权重,在计算时,使$i\leqslant j$以保证不会学习后面时间的权重.

2.2.3 小 结

本节详细介绍了DLKT领域针对长期依赖问题的解决方法,主要分为基于LSTM的方法与基于Transformer的方法.前者在一定程度上扩展了RNN序列学习的长度,但是并没有从根本上解决问题.后者不存在长期依赖问题,但是也丧失了RNN对序列建模的能力,位置嵌入对序列信息的影响更是需要深入研究.

2.3 针对缺少学习特征问题的改进

针对缺少学习特征问题的改进中,根据特征添加方式的不同,可以分为嵌入方式、损失函数、新结构3种方法.

2.3.1 嵌入方式

嵌入方式指将学习特征添加到模型的输入嵌入向量中,或作为额外的计算因子嵌入到计算过程中的方式.

1) DKT-FE(DKT with feature engineering)

Zhang等人[55]采用特征工程(feature engineering)的方式,使用人工分析选择特征并离散化.如图10所示,处理后的特征(答题时间、答题次数和第1个动作(首先尝试解答还是直接寻求帮助))在经过one-hot编码后与交互嵌入$\boldsymbol{x}_t$连接为$\boldsymbol{x}'_t$.然后利用自动编码器(auto-encoder)[56]降维并作为LSTM的输入,以减少模型的训练时间.Suragani等人[57]做了相同的工作.

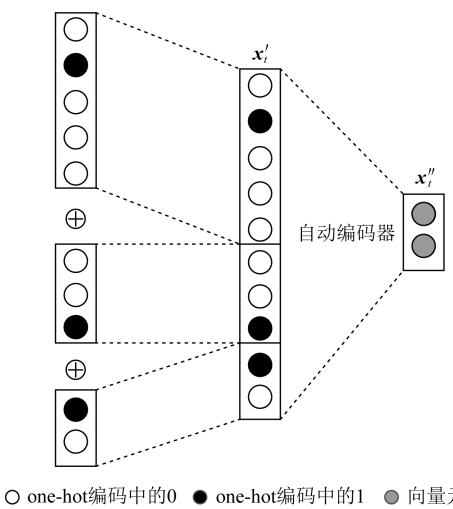

Fig. 10 Concatenate feature vectors and reduce dimensions

图10 连接特征向量并降维

2) DKT-DT(DKT with decision trees)

Yang等人[58]认为DKT-FE中使用人工分析特征的方式无法离散化很大的特征,且会带来人为的误差并提出了DKT-DT模型解决这个问题.DKT-DT模型使用了分类与回归树(classification and regression trees, CART)对one-hot编码的额外特征(标题、答题次数、答题时间)$\boldsymbol{f}_t$进行预处理.CART通过最小化交叉熵学习了系列分类规则,在时刻t,$\boldsymbol{f}_t$经过CART预处理后得到一个融合了多个特征信息的$\boldsymbol{f}'_t$,$\boldsymbol{f}'_t$进一步与其他学习交互信息连接,作为LSTM的输入:

$$\boldsymbol{x}'_t=[\boldsymbol{x}_t\oplus(\boldsymbol{f}'_t,a_t)].\tag{34}$$

3) DKVMN-DT(DKVMN with decision trees)

Sun等人[59]基于DKVMN模型添加特征,提出了DKVMN-DT(DKVMN with decision trees)模型.其使用了与DKT-DT相同的方法,不再赘述.

4) DKT+forgetting

Nagatani等人[60]在DKT模型中加入了遗忘特征.作者用3点信息衡量学生的遗忘情况,如图11所示,分别是相同题目时间间隔、相邻题目时间间隔和题目历史练习次数.

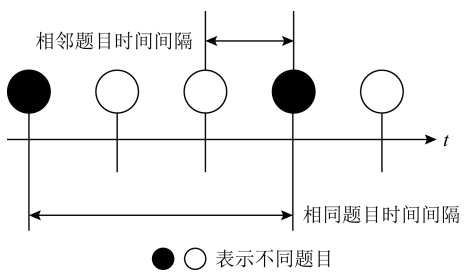

Fig. 11 Three measures of forgetting

图11 衡量遗忘的3种因素

使用one-hot编码衡量遗忘的3种因素,然后连接,得到一个multi-hot编码的遗忘因子$\boldsymbol{c}_t$.如图12所示,遗忘因子作为额外的特征,分别与交互嵌入$\boldsymbol{x}_t$和隐藏状态$\boldsymbol{h}_t$整合.

$$\boldsymbol{x}_t^{\mathrm{c}}=Aggregate^{\mathrm{in}}(\boldsymbol{x}_t,\boldsymbol{c}_t),$$
$$\boldsymbol{h}_t^{\mathrm{c}}=Aggregate^{\mathrm{out}}(\boldsymbol{h}_t,\boldsymbol{c}_{t+1}).\tag{35}$$

5) EERNN(exercise enhanced RNN)

Su 等人[40]关注到题目文本描述中包含的丰富信息,并使用双向 LSTM 提取文本描述的语义特征.如所图 13 所示,题目的文本描述被表示为单词序列并使用 Word2Vec[61]将转化为向量序列($\boldsymbol{w}_1$,$\boldsymbol{w}_2$,…,$\boldsymbol{w}_m$),作为双向 LSTM 的输入.然后,训练之后,连接前向状态和后向状态,经过一个逐元素的最大池化操作,得到最终的语义表示 $\boldsymbol{e}$.语义表示作为题目的嵌入,与答案组合成新的交互嵌入,作为 LSTM 的输入.

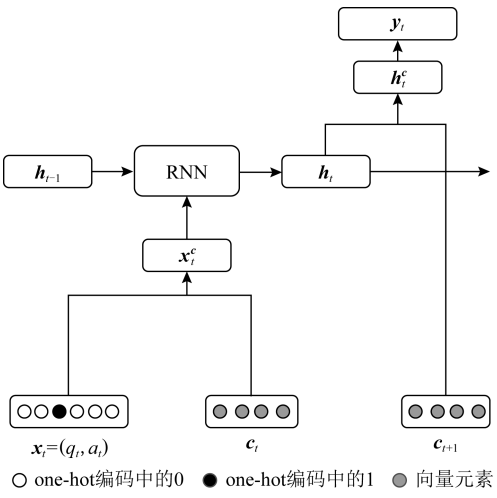

Fig. 12 The addition of forgetting features

图 12 遗忘特征的添加

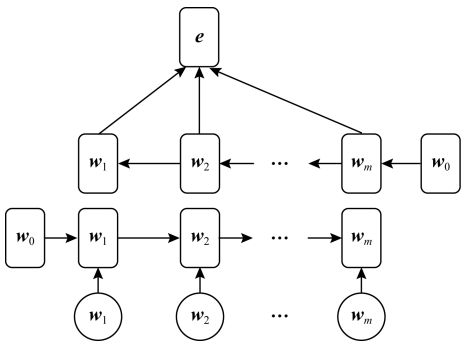

Fig. 13 The bidirectional LSTM is used to extract semantic features

图 13 使用双向 LSTM 提取语义特征

6) AKT(adaptable KT)

Cheng 等人[62]在其模型 AKT 中使用了与 EERNN 模型[27]相同的文本特征提取方法,并进一步从提出的语义特征中挖掘出学生的猜测(掌握了 KC 却没有答对题目)和失误(没掌握 KC 却答对了题目)行为.

在 AKT 中,猜测 $\boldsymbol{g}_t$ 和失误 $\boldsymbol{s}_t$ 分别用单层神经网络建模:

$$\boldsymbol{s}_t=S(\boldsymbol{e}_t),\boldsymbol{g}_t=G(\boldsymbol{e}_t). \tag{36}$$

$\boldsymbol{s}_t$ 与 $\boldsymbol{g}_t$ 进一步与学习交互嵌入 $\boldsymbol{x}_t$ 连接,经过 LSTM 得到隐藏状态 $\boldsymbol{h}_t$.然后将 $\boldsymbol{s}_t$,$\boldsymbol{g}_t$,$\boldsymbol{h}_t$ 三者结合得到知识状态 $\boldsymbol{h}'_t$:

$$\begin{aligned}\boldsymbol{h}_t&=f_{\text{LSTM}}(\boldsymbol{s}_t\oplus\boldsymbol{g}_t\oplus\boldsymbol{x}_t),\\ \boldsymbol{h}'_t&=(1-\boldsymbol{s}_t)\odot\boldsymbol{h}_t+\boldsymbol{g}_t\odot(1-\boldsymbol{h}_t),\end{aligned} \tag{37}$$

其中,f_{LSTM}代表 LSTM 模型,$\odot$表示逐元素相乘.最后,模型根据知识状态做出预测.

此外,AKT 还利用了迁移学习的思想,通过额外的自适应层一定程度上解决了数据稀疏问题.相比纳入专家知识[63],这种方法具有更好的泛化性.

7) EHFKT(exercise hierarchical feature enhanced KT)

Tong 等人[5]在其模型 EHFKT 中使用了 BERT[64]从题目的文本描述中提取了知识分布、语义特征和题目难度等信息.如图 14 所示,EHFKT 首先使用 BERT 生成文本描述的嵌入向量,然后经过知识分布提取系统(knowledge distribution extraction system, KDES)、语义特征提取系统(semantic feature extraction system, SFES)和题目难度提取系统(difficulty feature extraction system, DFES)分别生成知识分布、语义特征和题目难度.3 个特征连接之后经过 LSTM 的输入作出预测.

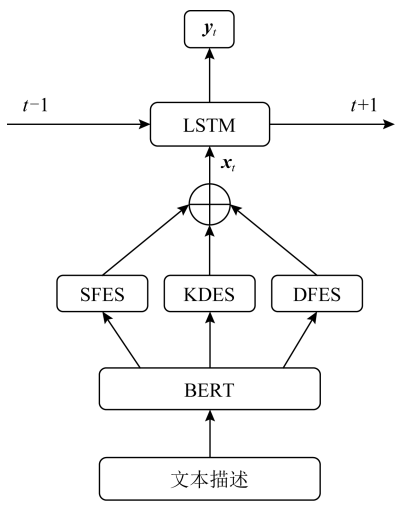

Fig. 14 Architecture for EHFKT

图 14 EHFKT 模型结构图

8) LSTMCQ(LSTM based contextualized Q-matrix)

Huo 等人[65]在其模型 LSTMCQ 中提出了一种带有上下文信息的题目编码方法.具体来说,首先

由领域专家手动创建一个矩阵 $\boldsymbol{Q}$,矩阵 $\boldsymbol{Q}$ 中存储了题目与 KC 的对应关系.迭代更新矩阵 $\boldsymbol{Q}$ 以附加 KC 的权重信息,且同一个问题的所有权重相加为 1,记为权重矩阵 $\boldsymbol{Q}$.最后,加入学生 n 在题目 m 上的表现信息 r_{nm}:

$$\delta_k(q_{mk}\neq 0)=\frac{1}{MN}\sum_{n=1}^{N}\sum_{m=1}^{M}r_{nm}q_{mk}, \tag{38}$$

其中,N,M 分别为学生和题目的数量.q_{mk} 为题目 m 中 KC k 所占的权重(即权重矩阵 $\boldsymbol{Q}$ 第 m 行第 k 列的值),δ_k 表示学生在 k 上的平均表现.将 δ_k 与 q_{mk} 相乘就得到了矩阵 $\boldsymbol{CQ}$(contextualized Q-matrix)对应的值:

$$\boldsymbol{CQ}_{mk}=\delta_k\times q_{mk}. \tag{39}$$

矩阵 $\boldsymbol{CQ}$ 包含了上下文信息,其每一行表示一个题目,将矩阵 $\boldsymbol{CQ}$ 的每一行单独提取出来,即为带有上下文信息的题目编码.

9) DKT-DSC(DKT with dynamic student classification)/DSCMN(dynamic student classification on memory networks)

Minn 等人[66-67]提出了一种根据能力分类学生,并将分类后的学生分组训练的方法,相当于在模型的输入中隐式地嵌入了学生能力信息.应用于 DKT 与 DKVMN,分别称为 DKT-DSC 和 DSCMN.

具体来说,首先通过设置一个时间间隔将交互序列分段,在每一个时间间隔,计算学生每个 KC 的答对率和答错率,并将两者之间的差值转化为数据向量,以代表学生的能力:

$$\begin{aligned}correct(x_j^z)&=\sum_{i=1}^{z}\frac{q_j==1}{N_{j,i}},\\ incorrect(x_j^z)&=\sum_{i=1}^{z}\frac{q_j==0}{N_{j,t}},\\ r(x_j^z)&=correct(x_j^z)-incorrect(x_j^z),\\ \boldsymbol{d}_i^z&=(r(x_1^z),r(x_2^z),\cdots,r(x_n^z)).\end{aligned} \tag{40}$$

其中,z 表示时间间隔,$N_{j,i}$ 表示前 i 个时间间隔内 q_j 被回答的次数.$\boldsymbol{d}_i^z$ 被解释为学生 i 的学习能力.

然后,用 K 均值(K-means)算法将学生按能力分组:

$$cluster(i,z)=\arg\min_{C\in[1,m]}\sum_{l=1}^{m}\sum_{\boldsymbol{d}_i^{z-1}\in C_l}\|\boldsymbol{d}_i^{z-1}-\boldsymbol{\mu}_l\|^2, \tag{41}$$

其中,$\boldsymbol{\mu}_l$ 为第 l 组(共 m 组)学生 C_l 的学习能力的均值.最后,将学生数据按照不同的分组放入模型中训练.

2.3.2 损失函数限制

损失函数限制指将额外的学习特征作为限制条件,编码到损失函数中的方式.

1) Colearn

Chaudhry 等人[68]关注到了学习交互系统中学生的提示获取(hint-taking)行为,并将其作为知识追踪的子任务,提出了一个多任务模型 Colearn. Colearn 基于 DKVMN 模型,所不同的是,Colearn 在更新知识状态矩阵 $\boldsymbol{M}^{\text{value}}$ 时,额外添加了 g_t(指示是否使用了提示),新的输入(q_t,a_t,g_t)用 one-hot 编码为向量 $\boldsymbol{f}_t$.Colearn 的输出有 2 个,答对的概率 $\boldsymbol{y}$ 和请求提示的概率 $\boldsymbol{y}_g$:

$$\begin{aligned}\boldsymbol{y}_t&=Sigmoid(\boldsymbol{W}^{\mathrm{T}}\cdot\boldsymbol{f}_t+\boldsymbol{b}),\\ \boldsymbol{y}_{g,t}&=Sigmoid(\boldsymbol{W}_g^{\mathrm{T}}\cdot\boldsymbol{f}_t+\boldsymbol{b}_g).\end{aligned} \tag{42}$$

对应地,模型的损失函数也由知识追踪任务和提示获取任务 2 部分组成,并用超参数 α,β 平衡权重:

$$L=\alpha\sum_{t=t_1}^{t_T}L_{\mathrm{BCE}}(p_t,a_{t+1})+\beta\sum_{t=t_1}^{t_T}L_{\mathrm{BCE}}(p_{g,t},g_{t+1}). \tag{43}$$

2) PDKT-C(prerequisite-driven DKT with constraint modeling)

Chen 等人[69]将 KC 之间的先后序关系引入了知识追踪模型,提出了 PDKT-C 模型.以 $P(m_{i,k,t}=1)$ 表示学生 i 在时刻 t 掌握 KC k 的概率,假设 k_1,k_2 存在先后序关系,将 KC 之间的关系建模为有序对,可以表示为

$$P(m_{i,k_2,t_2}=1)\leqslant P(m_{i,k_1,t_1}=1). \tag{44}$$

这个有序对很自然地说明了一个事实:一个知识点越先进,学习起来就越困难.在式(45)的约束下,经过正则化后,PDKT-C 模型的损失函数为

$$\begin{aligned}&\max_{\Theta}\sum_i\sum_{t=t_1}^{t_T}\log P(f_{i,t}\mid s_i,\Theta)+\\ &\alpha\sum_i\sum_{k_1,k_2}\sum_{t_1}\sum_{t_2}\delta(*)\times[\log P(m_{i,k_1,t_1})-\\ &\qquad\log P(m_{i,k_2,t_2})],\end{aligned} \tag{45}$$

其中,s_i 表示学生 i 的练习序列,Θ 指代 GRU 的参数,$f_{i,t}$ 指示学生 i 在时刻 t 是否回答了问题.第 1 部分为最大化似然函数.当 $f_{i,t_1}=f_{i,t_2}$ 时,$\delta(*)=1$,否则为 0,α 为平衡先决关系权重的超参数.

3) DKT-S(DKT with side information)

Wang 等人[70]提出了 DKTS 模型,通过在加入一个用来捕获题目之间关系的 Side Layer,将题目之间的关系纳入学生知识状态建模.

Wang 等人[70]认为,具有类似 KC 的问题是内在相关的,这种内在关系可以用图表示,其中节点是题目,若 2 个题目包含同样的 KC,则它们之间存在一条边.这个题目-题目图用邻接矩阵 $\boldsymbol{A}$ 表示.基于这种直觉,Side Layer 没有使用嵌入层,而是使用了图嵌入算法如 LINE[71]和 Node2Vec[72]以得到保持题目关系的表示.此外,Side Layer 中还包含 1 个正则化项 $L_1=\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{y}$,其中 $\boldsymbol{L}$ 是矩阵 $\boldsymbol{A}$ 的拉普拉斯矩阵.正则项的设计来源于这样一个直觉:如果 1 对题目包含相似的知识点,则学生答对这 2 个题目的概率也应该是相似的.DKTS 的损失函数包含 2 部分,即交叉熵损失和 Side Layer 的正则项,使用超参数 α 控制正则项的权重:

$$L=\sum_{t=t_1}^{t_T}L_{\mathrm{BCE}}(p_t,a_{t+1})+\alpha L_1. \tag{46}$$

4) DHKT(deep hierarchical KT)

Wang 等人[73]关注到了题目之间的层次结构关系,提出了 DHKT 模型.其通过 KC 嵌入和题目嵌入之间的内积铰链损失(hinge loss)来对层次关系建模.题目 $\boldsymbol{e}_i$ 与 KC $\boldsymbol{k}_j$ 之间的铰链损失定义为

$$L_{\mathrm{hinge}}^{i,j}=\begin{cases}\max(0,1-\boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{k}_j), & c_{i,j}=1,\\ \max(0,1+\boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{k}_j), & c_{i,j}=0.\end{cases} \tag{47}$$

其中,$c_{i,j}=1$ 表示 $\boldsymbol{e}_i$ 与 $\boldsymbol{k}_j$ 相关,反之无关.

在模型的训练过程中,同时最小化预测损失和铰链损失:

$$L=\sum_{t=t_1}^{t_T}L_{\mathrm{BCE}}(p_t,a_{t+1})+\alpha\sum_i\sum_j L_{\mathrm{hinge}}^{i,j}. \tag{48}$$

5) qDKT(question-centric DKT)

包含同样 KC 的题目之间存在着差异,解决这些题目对知识状态的贡献也是不同的.为了区分这些差异,Sonkar 等人[74]提出了 qDKT 模型,使用正则项对题目之间的差异建模:

$$diff(\boldsymbol{c})=\sum_i\sum_j\mathbf{1}(i,j)\cdot(c_i-c_j)^2, \tag{49}$$

其中,$\boldsymbol{c}=(c_1,c_2,\cdots,c_n)$包含了所有 n 个题目的正确率(通过数据集计算),如果 q_i,q_j 包含同一个知识点,则 $\mathbf{1}(i,j)=1$,否则为 0.这个正则项被加到模型的损失函数中作为限制:

$$L=\sum_{t=t_1}^{t_T}L_{\mathrm{BCE}}(p_t,a_{t+1})+\alpha\cdot diff(\boldsymbol{c}). \tag{50}$$

2.3.3 新结构

新结构指通过使用新的模型结构,将额外学习特征纳入模型计算过程中的方式.

1) GKT(graph based KT)

Nakagawa 等人[75]提出了 GKT 模型,通过将 KC 间的关系表示为 1 个有向图,知识追踪任务转化为了图神经网络(graph neural network, GNN)中的时间序列节点级分类问题.有向图 $G=(V,E,\boldsymbol{A})$由节点集(表示 KC)$V=\{v_1,v_2,\cdots,v_N\}$、边集(表示 KC 之间的关系)$E\subseteq V\times V$ 和邻接矩阵(定义关系的权重值)$\boldsymbol{A}\in\mathbb{R}^{N\times N}$定义.

在 GKT 中,假设学生在每个时刻 t 对每个知识点 v_i 都有独立的知识状态 $\boldsymbol{s}^t=\{\boldsymbol{s}_{i\in V}^t\}$.当学生回答与概念 v_i 相关的题目时,更新 v_i 及与 v_i 邻接的知识点所对应的知识状态 $\boldsymbol{s}_i^t$ 和 $\boldsymbol{s}_{j\in N_i}^t$.其中,$N_i$ 表示与 v_i 邻接的知识点的集合.因此,对于节点 v_i,首先要将其与邻接节点 $j\in N_i$ 的知识状态聚合:

$$\boldsymbol{s}_t'^t=\begin{cases}[\boldsymbol{s}_l^t,\boldsymbol{a}^t\boldsymbol{E}_x], & l=i,\\ [\boldsymbol{s}_l^t,\boldsymbol{E}_k(l)], & l\neq i,\end{cases} \tag{51}$$

其中,$\boldsymbol{E}_x,\boldsymbol{E}_k$ 分别为交互和 KC 嵌入矩阵,$\boldsymbol{E}_k(l)$表示 $\boldsymbol{E}_k$ 的第 l 行.根据图结构更新知识状态:

$$\begin{aligned}\boldsymbol{m}_l^{t+1}&=\begin{cases}f_{\mathrm{self}}(\boldsymbol{s}_l'^t), & l=i,\\ f_{\mathrm{neighbor}}(\boldsymbol{s}_i'^t,\boldsymbol{s}_l'^t), & l\neq i,\end{cases}\\ \hat{\boldsymbol{m}}_l^{t+1}&=G_{\mathrm{ea}}(\boldsymbol{m}_l^{t+1}),\\ \boldsymbol{s}_l^{t+1}&=G_{\mathrm{GRU}}(\hat{\boldsymbol{m}}_l^{t+1},\boldsymbol{s}_l^t).\end{aligned} \tag{52}$$

其中,f_{self}为多层感知机,G_{ea}为 DKVMN 模型中的删除-添加机制,G_{GRU}为 GRU 模型,f_{neighbor}为一个基于图结构信息向邻接节点传播的函数.

2) CKT(convolutional KT)

Shen 等人[76]提出了 CKT 模型,率先在知识追踪领域使用了卷积神经网络.如图 15 所示,在 CKT 中,使用 1 维卷积操作从矩阵 $\boldsymbol{Q}$ 中提取学习率特征,

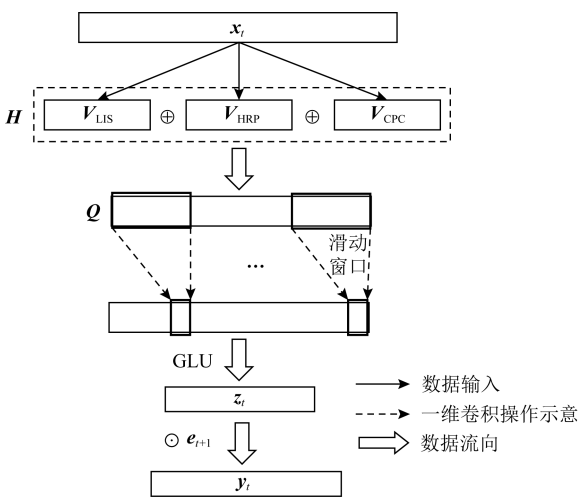

Fig. 15 Architecture of CKT model

图 15 CKT 模型结构图

滑动窗口将 d 个连续的学习交互映射为单个输出元素,并使用 GLU[77]作为非线性函数,在卷积层的输出上实现一个简单的选通机制,控制知识在学习过程中是否会被遗忘.最终的输出 $\boldsymbol{z}_t$ 作为学生的知识状态,用于知识追踪的预测任务.

$$\boldsymbol{y}_t = Sigmoid(\boldsymbol{z}_t \cdot \boldsymbol{e}_{t+1}). \tag{53}$$

矩阵 $\boldsymbol{Q}$ 由学习交互序列(learning interaction sequence, LIS)、历史相关表现(historical relevant performance, HRP)和 KC 的正确率(concept-wised percent correct, CPC)在合并之后经过 GLU 单元[77]组成:

$$\begin{aligned}\boldsymbol{H} &= \boldsymbol{V}_{\mathrm{LIS}} \oplus \boldsymbol{V}_{\mathrm{HRP}} \oplus \boldsymbol{V}_{\mathrm{CPC}},\\ \boldsymbol{Q} &= (\boldsymbol{H} \cdot \boldsymbol{W}_1 + \boldsymbol{b}_1) \otimes (\boldsymbol{H} \cdot \boldsymbol{W}_2 + \boldsymbol{b}_2).\end{aligned} \tag{54}$$

其中,$\otimes$表示逐元素的乘法,$\boldsymbol{V}_{\mathrm{HRP}}$,$\boldsymbol{V}_{\mathrm{CPC}}$计算为

$$\begin{aligned} r_t(i) &= Masking(\boldsymbol{e}_i \cdot \boldsymbol{e}_t),\\ w_t(i) &= Softmax(r_t(i)),\\ \boldsymbol{V}_{\mathrm{HRP},t}(t) &= \sum_{i=1}^{t-1} w_t(i)\boldsymbol{x}_i,\\ \boldsymbol{V}_{\mathrm{CPC},t}(q^k) &= \frac{1}{count(q^k)} \sum_{i=0}^{t-1} a_i^{q^k} == 1,\end{aligned} \tag{55}$$

其中,$Masking$ 为掩码操作,目的是排除后续时刻的学习交互,q^k 为与 KC k 相关的题目,$count(q^k)$ 表示 q^k 被回答的次数.

3) SKVMN (sequential key-value memory networks)

Abdelrahman 等人[47]在 SKVMN 模型中使用三角隶属度函数(triangular membership function)来计算 KC 之间的相关性:

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right), \tag{56}$$

其中,a,c 确定三角形的"脚",b 确定三角形的"峰".

相关性用来进一步计算题目间的顺序依赖关系,并作为 Hop-LSTM 的约束,控制信息在其上的跳跃连接,如图 7 所示.

4) DeepFM(deep factorization machines)

Vie[78]率先将 DeepFM 算法[79]应用到知识追踪领域,其优势在于对稀疏特征的添加与利用.DeepFM 由 FM 和 DNN(deep neural network)组成,前者的输出为

$$y_{\mathrm{FM}} = \sum_{i=1}^{N} w_i f_i + \sum_{1 \leqslant i \leqslant j \leqslant N} x_i x_j \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle, \tag{57}$$

其中,w_i 为偏置值,N 表示特征数量,f_i 表示第 i 个特征,$\boldsymbol{v}_i$ 表示系数矩阵 $\boldsymbol{V}$ 的第 i 维向量,$\langle\rangle$表示向量点积.

DNN 是一个 N 层的前馈神经网络,其输出为

$$\begin{aligned} \boldsymbol{e}^0 &= (\boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \cdots, \boldsymbol{v}_{i_c}),\\ \boldsymbol{e}^{n+1} &= ReLU(\boldsymbol{W}^n \boldsymbol{e}^n + \boldsymbol{b}^n), n \in [0, N],\\ \boldsymbol{y}_{\mathrm{DNN}} &= ReLU(\boldsymbol{W}^N \boldsymbol{e}^N + \boldsymbol{b}^N),\end{aligned} \tag{58}$$

其中,c 表示类别.DeepFM 的预测由 2 部分组合得到:

$$\boldsymbol{y} = Sigmoid(y_{\mathrm{FM}} + \boldsymbol{y}_{\mathrm{DNN}}). \tag{59}$$

5) KTM-DLF(knowledge tracing machine by modeling cognitive item difficulty and learning and forgetting)

Gan 等人[80]提出了一种结合学习者能力、认知项目难度、学习和遗忘等特征的建模方法,并使用 KTM 在高维中嵌入这些特征,所提出的模型称为 DKM-DLF.

具体来说,Gan 等人[80]认为,题目的难度包括 3 个方面:题目所包含的 KC 的难度、学习者的知识状态、题目的特点.综合这 3 个因素,题目难度被概括为认知项目难度(cognitive item difficulty, CID).设 β_k 为 KC k 的固有难度,δ_j 指代题目特点,$KC(j)$ 为题目 j 中包含的所有知识点,θ 为偏置值.则对于学生 i 在时间 t 作答的题目 j 来说,其 CID 为

$$\begin{aligned} d(i,j,t) = &\underbrace{\delta_j}_{\text{Part-1}} + \underbrace{\sum_{k \in KC(j)} \beta_k}_{\text{Part-2}} +\\ &\underbrace{\theta_m \Psi_{i,j,t} + \theta_n \left[\frac{\sum_{k \in KC(j)} \Psi_{i,k,t}}{|KC(j)|}\right]}_{\text{Part-3}}. \end{aligned} \tag{60}$$

上述 3 个因素分别对应 Part-1,2,3.其中,$\Psi \in [0,c]$代表难度,共 $c+1$ 个等级,用之前交互中的回答错误率计算:

$$\Psi_{i,v,t|v=\{j,k\}} = \begin{cases} \left[\frac{|a_{i,v}==0|_{0:t}}{|N_{i,v}|_{0:t}} \cdot (c-1)\right], & |N_{i,v}|_{0:t} \geqslant 5,\\ c, & |N_{i,v}|_{0:t} < 5, \end{cases} \tag{61}$$

其中,$|N_{i,v}|_{0:t}$表示学生 i 直到时刻 t 回答过的题目或 KC.学生的学习特征通过其在相同题目 $\Phi_{i,j,t}$ 和包含相同 KC 的不同题目 $\Phi_{i,k,t}$ 上的表现计算:

$$l(i,j,t) = \Phi_{i,j,t} + \sum_{k \in KC(j)} \Phi_{i,k,t}. \tag{62}$$

无论回答的正确与否,都有助于知识的获得:

$$\begin{aligned} \Phi_{i,v,t|v=\{j,k\}} = \sum_{tw=1}^{T} \{&\theta_{v,3tw+1} \log(1 + W_{i,v,tw}) +\\ &\theta_{v,3tw+2} \log(1 + F_{i,v,tw}) -\\ &\theta_{v,3tw+3} \log(1 + A_{i,v,tw})\}, \end{aligned} \tag{63}$$

其中,$W_{i,v,tw}$ 与 $F_{i,v,tw}$ 分别指代答对次数和答错次数,$A_{i,v,tw}$ 表示学生 i 在时间窗口 tw 的尝试次数,$tw|_{0:T}$是跨度不断增大的时间间隔.

学习某个知识点的间隔越长,遗忘的可能性越大,将遗忘行为定义为

$$f(i,j,t)=\theta_{j,j}\mathrm{e}^{\Delta_{j,j}}+\theta_{k,k}\sum_{k\in KC(j)}\mathrm{e}^{\Delta_{k,k}}+\theta_{j,j-1}\mathrm{e}^{\Delta_{j,j-1}}. \tag{64}$$

其中,e 为自然对数的底,Δ 为衡量遗忘的因素,有 3 个部分,与 DKT+forgetting(图 11)中提到的略有不同,具体如图 16 所示:

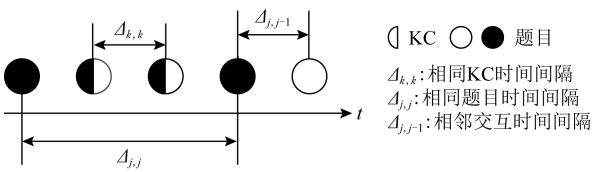

Fig. 16 Three measures of forgetting

图 16 衡量遗忘的 3 因素

设嵌入维度为 0,用 $\alpha_{i,j}$ 表示学生 i 在时刻 t 的能力,KTM-DLF 模型可以表示为

$$Sigmoid(P(Y_{i,j,t}=1))=\alpha_{i,t}-d(i,j,t)+l(i,j,t)-f(i,j,t). \tag{65}$$

6) DynEMb

Xu 等人[81]结合了矩阵分解和 RNN,提出了 DynEmb 模型,用矩阵分解做嵌入,用 RNN 对学习过程建模,DynEmb 主要分为 2 部分:.

① QuestionEmb.给出学习交互,用矩阵分解学习题目嵌入矩阵 $\boldsymbol{Q}$ 和学生嵌入矩阵 $\boldsymbol{S}$:

$$\arg\min_{\boldsymbol{Q},\boldsymbol{S},\boldsymbol{b},\boldsymbol{c}}\sum_{t=t_1}^{t_T}L_{\mathrm{BCE}}(Sigmoid(\boldsymbol{Q}_{q_t}\cdot\boldsymbol{S}_{s_t}+\boldsymbol{b}_{q_t}+\boldsymbol{c}_{s_t}),a_{t+1})+\lambda(\|\boldsymbol{Q}\|_{\mathrm{F}}^2+\|\boldsymbol{S}\|_{\mathrm{F}}^2), \tag{66}$$

其中,$\boldsymbol{b}$,$\boldsymbol{c}$ 分别为题目和学生的偏置项,λ 为平衡参数.

② StudentDyn.使用 RNN 生成动态的学生嵌入矩阵 $\boldsymbol{S}_t$.在时刻 $t-1$,RNN 的输入为题目嵌入 $\boldsymbol{Q}_{q_{t-1}}$ 和向量 $(a_{t-1},1-a_{t-1})^{\mathrm{T}}$ 的克罗内克积(Kronecker product),RNN 的隐藏状态就是学生 s_t 的动态嵌入 $\boldsymbol{S}_{s_t,t}$.DynEmb 预测下一时刻答对题目的概率:

$$\boldsymbol{y}_t=Sigmoid(\boldsymbol{Q}_{q_t}\cdot\boldsymbol{S}_{s_t,t}+\boldsymbol{b}_{q_t}). \tag{67}$$

7) BDKT(Bayesian neural network DKT)

Li 等人[82]提出了 BDKT 模型,使用贝叶斯神经网络来对丰富的学习特征建模.BDKT 的参数设置为分布形式:

$$P(W|X,Y)=\frac{p(W)p(Y|X,W)}{p(Y|X)}, \tag{68}$$

其中,X,Y 分别为输入和输出,$p(W)$ 为参数 W 的先验.由于 $P(W|X,Y)$ 在训练集 (X,Y) 上的概率分布是复杂的,难以用归一化常数处理,也不能直接计算模型参数的后验分布.BDKT 使用了变分推断(variational inference)解决这个问题,变分分布 $q(W)\sim N(W|\mu,\sigma^2)$ 被用来近似真正的后验分布 $P(W|X,Y)$,$N(\mu,\sigma^2)$ 表示数学期望为 μ、方差为 σ^2 的正态分布.p,q 之间距离为

$$KL(q\parallel p)=\sum_{i=1}^{n}q(W)\log\frac{q(W)}{p(W\mid X,Y)}=E_q[\log q(W)]-E_q[\log p(W)]-E_q[\log p(Y|X,W)]+E_q[\log p(Y|X)], \tag{69}$$

其中,$E_q[\log p(Y|X)]$ 为常量,$E_q[\log p(Y|X,W)]$ 可以从训练集中得到.BDKT 的损失由训练损失和 KL 项决定:

$$L=\hat{y}-y+E_q[\log q(W)]-E_q[\log p(W)]. \tag{70}$$

8) Q-Embedding

Nakagawa 等人[83]提出了无需 KC 标签信息的 Q-Embedding 模型,可以自动学习题目与 KC 的嵌入.

Q-Embedding 模型的结构如图 17 所示,其中,$\boldsymbol{P}$ 为需要学习的题目-KC 矩阵,$\boldsymbol{u}_t$ 与 $\boldsymbol{v}_t$ 为额外的隐藏层,长度分别为 $2N'$ 和 N',N' 为自定义的 KC 个数,$\boldsymbol{u}_t$,$\boldsymbol{v}_t$,$\boldsymbol{P}$ 的定义为

$$\begin{aligned}\boldsymbol{u}_t&=[\boldsymbol{x}_{t_{\mathrm{pos}}}\boldsymbol{P}\oplus\boldsymbol{x}_{t_{\mathrm{neg}}}\boldsymbol{P}],\\ \boldsymbol{v}_t&=Sigmoid(\boldsymbol{W}_{hv}\boldsymbol{h}_t+\boldsymbol{b}_v),\\ \boldsymbol{P}&=Sigmoid(\boldsymbol{W}_{xu}).\end{aligned} \tag{71}$$

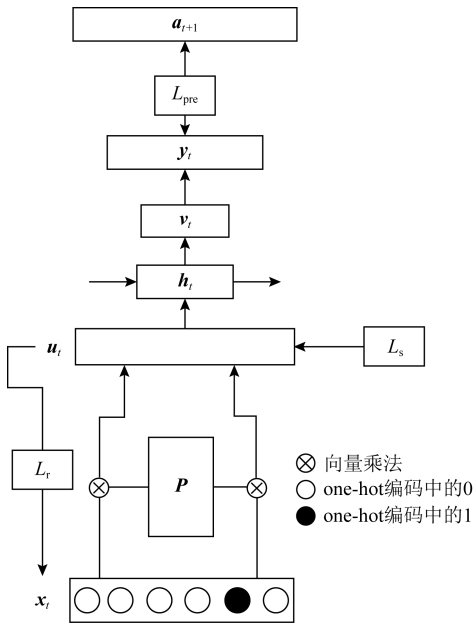

Fig. 17 Architecture of Q-Embedding model

图 17 Q-Embedding 模型结构图

式(71)中,$\boldsymbol{x}_{t_{\text{pos}}}$,$\boldsymbol{x}_{t_{\text{neg}}}$分别表示$\boldsymbol{x}_t$的前后2部分,$\boldsymbol{h}_t$为RNN的隐藏状态.使用2个额外的损失函数对模型进行训练:

$$\begin{aligned}\boldsymbol{x}'_t &= Sigmoid(\boldsymbol{W}_{vy}\boldsymbol{u}_{t_{\text{pos}}}+\boldsymbol{b}_y),\\ L_{\text{ref}} &= \sum_{t=t_1}^{t_T} L_{\text{BCE}}(\boldsymbol{x}'^{\mathrm{T}}_t\boldsymbol{q}_t, a_{t+1}),\\ L_{\text{spa}} &= \sum_{t=t_1}^{t_T}(0.5-|\boldsymbol{u}_t-0.5|),\end{aligned} \tag{72}$$

其中,L_{ref}为对题目空间和KC空间的重构正则化,旨在反映以下假设:从学生对KC空间对应的每个KC的理解中,可以估计出学生对题目的回答情况.L_{s}为题目嵌入矩阵的稀疏正则化,目的是在训练模型后二值化时使$\boldsymbol{P}$接近$\boldsymbol{0}$或$\boldsymbol{1}$,并抑制信息丢失.Q-Embedding的损失函数分为3部分,预测的交叉熵L_{pre},L_{ref},L_{spa},用超参数α,β,γ平衡权重:

$$L=\alpha L_{\text{pre}}+\beta L_{\text{ref}}+\gamma L_{\text{spa}}. \tag{73}$$

2.3.4 小 结

本节详细介绍了DLKT领域针对缺少学习特征问题的改进方法,主要分为嵌入方式、损失函数限制和新结构.其中,嵌入方式的最为直观:通过添加更多信息,借助深度学习的特征提取能力,使模型自主地建立特征间的联系.损失函数将额外信息作为一种限制条件,使模型向指定方向优化.新结构方式则充分利用了其他网络结构的特点,带来了许多优点(如预防过拟合[82]、减轻数据稀疏问题[62]等),但新结构的迁移应用研究不够深入,其可能存在的缺点值得更进一步研究.

3 DLKT模型对比与分析

第2节详细介绍了DKT模型的改进方法,按照改进侧重点的不同,分为可解释问题的改进、长序列依赖问题的改进、添加特征的改进和其他方面的改进.本节归纳了各类改进方法中所使用的数据集,同时也对DLKT在各公开数据集上的表现做了对比和分析.

3.1 数据集介绍

近年来,用作DLKT模型评估的公开数据集主要有9个,其简述和下载链接如表2所示:

Table 2 DLKT Domain Public Datasets Summaries, Download Links and Models that Use Them

表2 DLKT领域公开数据集简述、下载链接及使用其的模型

数据集	简述	下载链接
Synthetic	DKT模型使用的模拟数据集,它模拟了2 000名虚拟学生,他们回答了来自5个虚拟知识点的50个问题.仅在此数据集中,所有学生回答问题的顺序相同.	https://github.com/chrispiech/DeepKnowledgeTracing/tree/master/data/synthetic
Static2011	来自一个大学级的工程静力学课程,具有333个学生在1 223个问题上的189 927个交互.	https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507
KDDCup2010	2010年KDD杯比赛开发数据集,具有574个学生在436个问题上的607 026个交互.	https://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp
EdNet	由Santa(一个人工智能导学系统)收集的大规模分层的学生活动数据集,包含784 309名学生的131 317 236个交互信息,是迄今为止发布的最大的公共交互教育系统数据集.	https://github.com/riiid/ednet
Junyi	来自Junyi Academy(一个在线教育网站),除EdNet外数据量最多的开源数据集.	https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=1198
ASSIST2009	来自ASSISTMENTS在线辅导系统,去掉重复记录之后,包含4 151个学生在110个问题上的325 673个交互.	https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010data/skill-builder-data-2009-2010
ASSIST2012	包含27 066个学生在45 716个问题上的2 541 201个交互.	https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-with-affect
ASSIST2015	包含19 840个学生在100个问题上的683 801个交互.	https://sites.google.com/site/assistmentsdata/home/2015-assistments-skill-builderdata
ASSIST2017	包含686个学生在102个问题上的942 816个交互.	https://sites.google.com/view/assistmentsdatamining/dataset?authuser=0

3.2 模型对比与性能概览

表3总结了各种模型所属的改进方向类别和其主要的改进方式.表4总结了使用公开数据集的DLKT模型的性能表现(以大多数论文都采用了的AUC指标为基准),表4中的数据皆来自于模型初始论文,取最大值.需要指出的是,深度学习模型受参数设置影响较大,且同一个模型在不同论文中的表现也存在较大差异,因此,表4中的数据参考价值大于实际意义.

Table 3 An Overview of DKT Model Improvement Methods

表 3 DKT 模型改进方法概览

方法名称	改进类别	改进方式
A-DKT[37]	可解释性问题的改进	计算 KC 之间的注意力
DKVMN[38]	可解释性问题的改进	计算题目与 KC 之间的注意力
DKVMN-CA[39]	可解释性问题的改进	改进 DKVMN,使其支持自定义的 KC 标签
EERNN[40]	可解释性问题的改进	计算隐藏状态之间的注意力
EKT[41]	可解释性问题的改进	结合 EERNN 和 DKVMN
KQN[42]	可解释性问题的改进	基于向量点积的自解释模型
Deep-IRT[43]	可解释性问题的改进	基于 IRT 理论,在 DKVMN 基础上构建的自解释模型
LRP[44]	可解释性问题的改进	使用分层相关性传播方法解释计算过程中的数值变化
不确定性评估[45-46]	可解释性问题的改进	使用不确定性评估方法减轻预测过程中的不透明性
Hop-LSTM[47]	长期依赖问题的改进	根据拓扑关系跳跃连接的 LSTM
NKT[48]	长期依赖问题的改进	双层堆叠 LSTM
SAKT[52]	长期依赖问题的改进	Transformer 编码器结构的迁移应用
SAINT[53]	长期依赖问题的改进	完整的 Transformer 模型的迁移应用
DKT+Transformer[54]	长期列依赖问题的改进	基于时序信息,改进了自注意力的计算方式
DKT-FE[55]	缺少学习特征问题的改进	使用特征工程,人工分析特征,特征与交互共同嵌入
DKT-DT[58]/DKVMN-DT[59]	缺少学习特征问题的改进	使用 CART 提取特征,特征与交互共同嵌入
DKT+forgetting[60]	缺少学习特征问题的改进	基于历史交互记录计算遗忘因子,遗忘因子与交互共同嵌入
EERNN[35]	缺少学习特征问题的改进	双向 LSTM 提取题目文本的语义特征,语义特征与交互共同嵌入
AKT[62]	缺少学习特征问题的改进	从题目文本中提取猜测和失误行为特征,特征用于隐藏状态的更新
EHFKT[5]	缺少学习特征问题的改进	使用 BERT 提取题目文本特征,特征用于隐藏状态的更新
LSTMCQ[65]	缺少学习特征问题的改进	融合上下文信息的交互嵌入方式
DKT-DSC[66]/DSCMN[67]	缺少学习特征问题的改进	将学生按能力分组进行训练以隐式添加学生能力特征
Colearn[68]	缺少学习特征问题的改进	模型额外输入提示获取概率,并添加额外的损失函数
PDKT-C[69]	缺少学习特征问题的改进	将题目的拓扑关系正则化,作为额外的损失函数
DKT-S[70]	缺少学习特征问题的改进	使用图嵌入算法嵌入拓扑关系,添加额外的损失函数
DHKT[73]	缺少学习特征问题的改进	用铰链损失建模题目间的差异,添加额外的损失函数
qDKT[74]	缺少学习特征问题的改进	将题目差异正则化,作为额外的损失函数
GKT[75]	缺少学习特征问题的改进	使用图神经网络添加题目拓扑关系
CKT[76]	缺少学习特征问题的改进	使用卷积神经网络提取特征
SKVMN[47]	缺少学习特征问题的改进	使用三角隶属度函数计算拓扑关系,并构造跳跃连接的 RNN
DeepFM[78]	缺少学习特征问题的改进	使用因式分解机构造并提取特征
KTM-DLF[80]	缺少学习特征问题的改进	利用 KTM 与 RNN,结合学生能力、题目难度、学习与遗忘行为的建模
DynEmb[81]	缺少学习特征问题的改进	矩阵分解与 RNN 的结合
BDKT[82]	缺少学习特征问题的改进	利用贝叶斯网络对丰富的学习特征建模,可以预防过拟合问题
Q-Embedding[83]	缺少学习特征问题的改进	端对端的 KT 模型,无需 KC 标签信息

Table 4 An Overview of DLKT Models

表 4 DLKT 模型 AUC 指标对比概览

%

方法	数据集								
	Simulated-5	Static2011	KDDCup2010	EdNet	Junyi	ASSIST09	ASSIST12	ASSIST15	ASSIST2017
DKT	75.00	80.20				86.00		72.52	
DKVMN	82.73	82.84				81.57		72.68	
DKT+	82.64	83.49				82.27		73.71	73.43
KQN	82.82	83.16				82.35		73.40	
Deep-IRT		82.98				81.65		72.88	

续表 4 %

方法	数据集								
	Simulated-5	Static2011	KDDCup2010	EdNet	Junyi	ASSIST09	ASSIST12	ASSIST15	ASSIST2017
AKT	81.25	86.06	81.85		**90.41**	84.61		72.82	71.81
CKT	82.90	83.00				82.50		73.50	71.50
DKT-FE		74.00				86.70			
DKT-DT					73.00	74.90			
DKVMN-DT						89.70			
SKVMN	**84.00**	84.85			82.67	83.63		74.84	
SAKT	83.20	85.30				84.8		85.40	73.40
qDKT		83.40				76.20			77.00
DKT-DSC			81.00			**91.00**	**87.00**	**87.00**	
DSCMN			**86.00**			81.20	78.5	71.00	
BDKT			75.20			65.00			
GKT			76.9			72.3			
Q-Embedding			80			76.00			
DKT+TF		**94.70**	78.40						**80.6**
SAINT				**78.11**					
NKT						77.87			
PDKT-C						78.00			
DHKT		83.33				78.66	77.47		
DynEmb		83.63				73.90	73.60		
A-DKT						82.50			
LSTMCQP						73.88			

注:黑体值表示对应数据集上的最佳结果.

4 DLKT 模型的应用

在 DLKT 领域,除了对 DKT 模型的改进之外,还有许多研究致力于探索 DLKT 模型在教育领域的应用.如图 18 所示,除了主要用来预测学生下一次答对题目的概率,DLKT 模型还有许多其他应用,下面简单介绍这些应用.

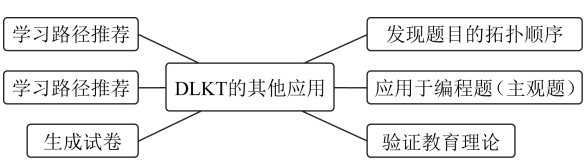

Fig. 18 Other applications of DLKT model

图 18 DLKT 模型的其他应用

4.1 发现题目的拓扑顺序

Zhang 等人[84]提出了一种基于规则的方法,利用 DKT 模型发现知识点之间的拓扑顺序.具体来说,在 DKT 模型中,由于一个题目对应一个知识点,因此,答对题目的概率可以视为掌握题目对应知识点的概率.将输出概率大于 0.5 的知识点视为掌握,拓扑顺序的发现分为 3 个步骤:

1) 确定偏序关系.若当前的知识点被掌握,则输出概率最高的知识点为先决知识点;若当前的知识点未被掌握,则输出概率最低的知识点为先决知识点.

2) 删除冗余连接.设 k_a,k_b,k_c 为存在 $k_a \rightarrow k_b$,$k_b \rightarrow k_c$,$k_a \rightarrow k_c$ 关系的 3 个 KC,删除其中的冗余连接,得到类似 $k_a \rightarrow k_b \rightarrow k_c$ 的有向无环图.

3) 用 Kahn's 算法[85]把有向无环图转化为拓扑顺序.

4.2 应用于编程题(主观题)

Wang 等人[86]将 LSTM 模型应用到编程题上,模型的输入为学生对单个编程题目的多次提交记录.提交的代码被表达成一个抽象语法树(abstract syntax tree, AST),然后利用递归神经网络对代码的 AST 进行向量化[87-88],向量化后的 AST 作为 LSTM 的输入,最后,模型预测学生能够成功解决同一个知识点的下一个编程题的概率.Wang 等人[86]的模型仅支持基于块的编程语言 Scratch,Swamy 等人[89]

扩展了他们的工作,提出了一种对编程语言没有限制的模型.在模型的输入方面,Swamy 等人使用了 scikit-learn 的默认标记方案对学生代码中的非字母数字字符做分词,然后计算词频-逆文档频率(term frequency-inverse document frequency, tf-idf)生成代码的向量表示.向量的长度为单词的总数,其中每一项为对应单词出现的次数.代码的向量化表示与 one-hot 编码的学生编号、问题编号、尝试次数编号组合,作为 LSTM 的输入,模型的预测结果为完成每个知识点对应题目的剩余尝试次数.

4.3 验证教育理论

Lalwani 等人[90]利用 DKT 模型验证改进的布鲁姆分类法(revised Bloom's taxonomy).改进的布鲁姆分类法将认知分为 6 个阶段,分别是记忆、理解、应用、分析、评估和创造.这 6 个阶段的复杂度逐渐递增,且前面的阶段是后面阶段的先决条件.Lalwani 等人[90]将验证过程抽象为研究掌握前面阶段的 KC 对掌握后面阶段知识点的影响.具体来说,通过模型的输出判断学生是否掌握 KC,然后计算不同阶段 KC 之间掌握程度的差异:

$$\begin{gathered}(i\to j)-(*\to j),(\hat{i}\to j)-(*\to j),\\(\hat{i}\to j)-(\bar{i}-j),(j\to i)-(*\to i),\\(j\to i)-(i\to i),(j\to i)-(\hat{i}\to i),\end{gathered}\tag{74}$$

其中,i,j 分别代表掌握 k_i,k_j 的概率,且 i 的分类阶段在 j 之前,$*$ 表示未掌握 KC 的初始状态.$\hat{i}$ 表示分类阶段在 i 之前的知识点,包括 i.$\bar{i}$ 表示分类阶段在 i 之前的知识点,不包括 i.$i\to j$ 表示在掌握 i 的前提下掌握 j 的概率.

在后续的工作中,Lalwani 等人[91]在原始的 DKT 模型中加入了学生练习之间的时间间隔作为特征,将改进后的模型命名为 DKT-t 通过比较 DKT 与 DKT-t 模型的差异,研究时间间隔对预测的影响,并进一步利用 DKT-t 模型追踪遗忘曲线.

4.4 学习路径推荐

Cai 等人[92]提出了一种利用知识追踪以及强化学习技术的学习路径推荐方法,并命名为 KT-KDM. KT-KDM 分为 2 部分,KTM 和 DKM,前者是一个基于 DKT+模型的知识追踪模型,后者通过预测知识需求水平(level of knowledge requirements, LKR)来获得 KTM 提供的学习者知识掌握程度,并推荐习题.

如图 19 所示,推荐过程被建模为一个采用强化学习方法的 Markov 决策模型.KDM 本质上是一个使用 KTM 建模的学习状态作为输入的预测网络,其输出为一个向量,其中每个元素表示每个习题的 LKR.然后使用随机加权函数选择一个习题进行推荐.奖励函数设计为相邻 2 个推荐的习题的掌握度之差,同时,为了避免模型反复推荐相同的高回报习题,对已经推荐的习题进行了惩罚.奖励函数为

$$r_t=\begin{cases}\dfrac{s_{k,t}-s_{k,t-1}}{n_{k,t}}, & s_{k,t}-s_{k,t-1}>0,\\ s_{k,t}-s_{k,t-1}, & s_{k,t}-s_{k,t-1}\leqslant 0,\end{cases}\tag{75}$$

其中,k 表示推荐的最后一个 KC,$s_{k,t}$ 表示 k 在时刻 t 的知识状态,$n_{k,t}$ 表示在 k 被推荐的次数.

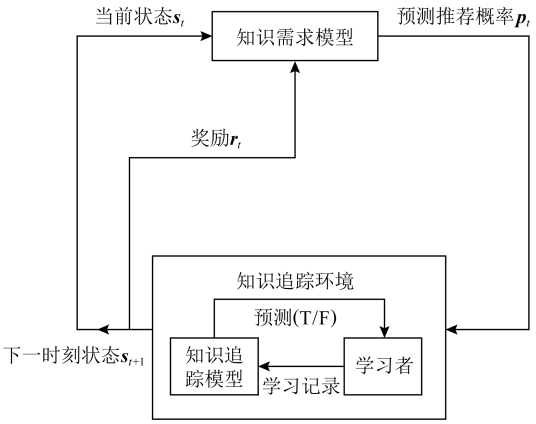

Fig. 19 Markov process model the recommendation learning path

图 19 学习路径推荐过程的 Markov 决策模型

4.5 STEM/Non-STEM 职业预测

Yeung 等人[93]将 DKT+模型预测的结果与从数据集中提取出的学生特征相结合,以预测学生是否会从事 STEM 类职业.

如图 20 所示,将 DKT+模型最后一次预测的结果 $\boldsymbol{y}_t$ 与学生的特征 $\boldsymbol{x}_{sp}$ 连接得到 $\boldsymbol{x}'=\boldsymbol{y}_t\oplus\boldsymbol{x}_{sp}$,作为机器学习模型的输入.机器学习模型学习 $\boldsymbol{x}'$ 与

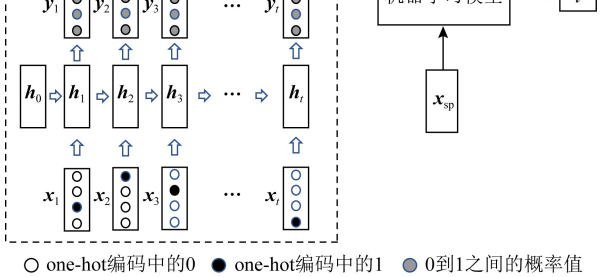

Fig. 20 Architecture of STEM/Non-STEM career prediction model

图 20 STEM/Non-STEM 职业预测模型结构

STEM 标签 $l\in\{0,1\}$ 的映射关系,并做出 STEM/Non-STEM 的职业预测,有 GBDT,LAD,LR,SVM 这 4 种实现方式.

4.6 生成试卷

Wu 等人[94]提出了一种利用知识追踪的试卷生成方法.具体来说,假设一张试卷包含 n_q 个题目与 n_k 个知识点,则试卷可以表示为一个矩阵 $\boldsymbol{E}\in\mathbb{R}^{n_q\times n_k}$.使用每种知识点出现的概率表示试卷中每种知识点的权重,用 $P(o_j)$ 表示 KC j 出现的概率:

$$P(o_j)=\frac{\sum_{i=1}^{n_q}E_{ij}}{\sum_{j=1}^{n_k}\sum_{i=1}^{n_q}E_{ij}},\tag{76}$$

其中,E_{ij} 表示 $\boldsymbol{E}$ 的第 i 行第 j 列.假设有 n_s 个学生,学生的对知识点的掌握程度可以表示为矩阵 $\boldsymbol{S}\in\mathbb{R}^{n_s\times n_k}$.知识掌握程度来自于深度知识追踪模型输出的答对题目的概率,由于一个题目对应一个知识点,所以答对题目的概率可以看作是对知识点的掌握程度.将题目中所有知识点的掌握程度相乘即为答对题目的概率,学生 l 答对题目 i 的概率 $P(a_{li}=1)$ 表示为

$$P(a_{li}=1)=\prod_{j=1}^{n_k}(S_{lj}\,|\,E_{lj}==1).\tag{77}$$

用 $Score_i$ 表示问题 i 的总分数,则学生 l 的试卷成绩可以表示为

$$r_l=\sum_{i=1}^{n_q}(r_{li}\times Score_i).\tag{78}$$

将 n_s 个学生的成绩联合,表示为 $R=\{r_1,r_2,\cdots,r_n\}$,用 $P(R)$ 表示 R 的分布.Wu 等人提出,合理的试卷有 2 个必要条件:

1) 卷中 KC 分布与课程中的接近,即 $\boldsymbol{\Phi}\sim\boldsymbol{W}$.其中,$\boldsymbol{\Phi}=(P(o_1),P(o_2),\cdots,P(o_{n_k}))$,$\boldsymbol{W}$ 中每一项表示对应 KC 的权重.

2) 每一组学生成绩的分布应该符合正态分布,即 $P(R)\sim N(\mu,\sigma^2)$,其中,$N(\mu,\sigma^2)$ 表示均值为 μ、标准差为 σ^2 的正态分布.

最后,使用动态规划算法和遗传算法更新试卷,使其符合上述 2 个必要条件.

5 总结与展望

本文聚焦教育大数据中的知识追踪,对该领域内基于深度学习方法的知识追踪模型进行了全面回顾和系统性的梳理.首先介绍了该领域的开创性工作 DKT,然后基于该工作,分针对可解释问题的改进、针对长期依赖问题的改进以及针对缺少学习特征问题的改进三大主要技术改进方向构建了技术演进脉络图、梳理了各模型的技术重难点、分析了各模型的优点和局限性.其中,可解释性问题作为深度学习领域普遍存在的问题,尚未得到有效解决,DLKT 也不例外,目前的方法都只能有限地提高可解释性.长期依赖问题在自注意力模型上得到了完美解决,但其需要额外的位置编码才可以维持序列学习能力.对缺少学习特征问题的研究占据了 DLKT 的主要部分,嵌入方式、损失函数限制和新结构 3 种方式各有优劣:嵌入方式直观但是过于依赖模型的学习能力,损失函数限制可以使指定模型优化的方向但需要大量的人工工作,新结构的使用带来了许多优点,但深入研究的缺乏可能使其中的缺点无法暴露.最后我们还整理了可供研究者使用的公开数据集,对比评估了各模型的性能表现和考察了该领域的主要应用.

基于深度学习的知识追踪因其优秀的性能而被广泛关注.对于目前发展迅速的线上教育,其产生的大量教育数据正好对应了深度学习模型对于数据量的需求,而无需标注数据的特性大大提高了数据的利用率,同时,深度学习框架的普及降低了模型构建的门槛.多种因素的共同作用,使得 DLKT 被广泛应用于在线教育平台,也使其成为教育数据挖掘领域新的研究热点.但是目前该领域尚在起步阶段,在实际应用中,仍有许多挑战和问题亟待解决.基于目前存在的挑战与问题,我们总结了 7 个可能的研究方向供研究者参考.

1) 现有 DLKT 模型大多使用二元变量来表示题目的回答情况,这种建模方式不适合分数值分布连续的主观题.Wang 等人[86]和 Swamy 等人[89]在处理学生的编程数据时,使用了学习者回答的连续快照作为回答情况的指示器,这提供了一种对主观题目建模的方式.而其他的对主观题目的建模方法仍有很大的研究前景.

2) 目前 DLKT 主要应用于在线教育平台,如何利用好在线平台所提供的大量学习轨迹信息,是研究的难点之一.Mongkhonvanit 等人[95]提供了一种对教学视频观看行为建模的方法,Huan 等人[96]则利用了鼠标轨迹信息.而其他学习特征信息的提取、建模亟需更多的研究.与此同时,特征的添加也是一大难点.对于以 RNN 为基础的 DLKT 模型来说,输入向量的长度会显著影响模型的训练速度.这就

需要使用降维方法减小向量的长度,或者采用其他的嵌入方式(如 LSTMCQ)融合更多特征而不增加向量长度.总而言之,学习特征信息的提取、建模、添加将会是 DLKT 实际应用中的重点研究方向.

3) DLKT 的优秀性能使利用其验证经典教育理论成为可能.如 Lalwani 等人[90]验证改进的布鲁姆分类与遗忘曲线.同时,已提出的教育理论也可以为建模提供指导,如 Gan 等人[80]结合了学习与遗忘理论.经典教育理论在 DLKT 领域的应用值得更多的研究者加以关注.

4) 利用 DLKT 模型构建知识图谱.DLKT 模型可以用来发现知识点之间的相互关系,构建出知识点关系图,这可以看作是简化的知识图谱.知识图谱作为当前人工智能时代最为主要的知识表现形式,如何扩展模型的知识结构发现能力,将知识点关系图扩展为知识图谱将会是未来的重点研究方向.

5) 目前的 DLKT 模型中仍存在许多不确定因素,现有的理论推断并不足以解释 DLKT 模型的训练过程.在基于 Transformer 的模型中,掩码机制被用来屏蔽后面时间的权重,这是为了防止未答的题目影响已答的题目.而 Xu 等人[97]使用双向 LSTM 以融合过去和未来的上下文序列信息.两者所依据的原理是相悖的,但都获得了性能提升.如何深入研究,以完整解释 DLKT 模型的训练过程,将会是未来的重点研究方向.

6) 目前 DLKT 主要使用 RNN 模型,许多研究已经证明了 RNN 的优越性.同时,Transformer 模型、GNN 模型也在知识追踪领域有着优秀的表现.而其他更多模型的应用仍亟需深度研究,对其他深度学习模型的应用将会是重要研究方向.

7) Transformer 相对于 RNN 的一大优势就是没有长期依赖问题,但目前基于 Transformer 的 DLKT 模型却并没有利用好这个优势,如 SAKT 和 SAINT,它们都将序列长度设置为 100,这个长度并没有超过 LSTM 的序列学习容量(200).同时,实验显示,位置编码的有无对最终的结果影响并不大.这似乎说明长期依赖与序列关系对 KT 任务的影响没有目前所认为的那么大,以此类推,各种学习特征对于 KT 任务的影响值得进一步研究.

作者贡献声明:刘铁园、陈威是综述的主要写作人,完成相关文献资料的收集和分析、论文初稿的写作和校对;常亮、古天龙是项目的构思者及负责人,指导论文写作.

参 考 文 献

[1] Li Feiming, Ye Yanwei, Li Xiaofei, et al. Application of knowledge tracing model in education: A review from 2008 to 2017 [J]. Distance Education in China, 2019(7): 86-91 (in Chinese)
(李菲茗, 叶艳伟, 李晓菲, 等. 知识追踪模型在教育领域的应用: 2008—2017 年相关研究的综述[J]. 中国远程教育, 2019(7): 86-91)

[2] Ha H, Hwang U, Hong Yongjun, et al. Deep trustworthy knowledge tracing [J]. arXiv preprint, arXiv: 1805.10768, 2018

[3] Pardos Z A, Bergner Y, Seaton D T, et al. Adapting Bayesian knowledge tracing to a massive open online course in edx [C] //Proc of the 6th Int Conf on Educational Data Mining(EDM). Worcester, MA: IEDMS, 2013: 137-144

[4] Wang Zhuo, Zhu Jile, Li Xiang, et al. Structured knowledge tracing models for student assessment on coursera [C] //Proc of the 3rd ACM Conf on Learning @ Scale(L@S). New York: ACM, 2016: 209-212

[5] Tong Hanshuang, Zhou Yun, Wang Zhen. Exercise hierarchical feature enhanced knowledge tracing [C] //Proc of the 21st Int Conf on Artificial Intelligence in Education (AIED). Berlin: Springer, 2020: 324-328

[6] 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

[7] Baker R S J D, Corbett A T, Aleven V. More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing [G] // LNCS 5091: Proc of Intelligent Tutoring Systems (ITS). Berlin: Springer, 2008: 406-415

[8] Pavlik P I, Cen H, Koedinger K R. Performance factors analysis: A new alternative to knowledge tracing [C] //Proc of the 14th Int Conf on Artificial Intelligence in Education (AIED). Berlin: Springer, 2009: 531-538

[9] Wauters K, Desmet P, Van Den Noortgate W. Adaptive item-based learning environments based on the item response theory: Possibilities and challenges [J]. Journal of Computer Assisted Learning, 2010, 26(6): 549-562

[10] Gong Yue, Beck J E, Heffernan N T. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures [G] //LNCS 6094: Proc of the 10th Int Conf on Intelligent Tutoring Systems (ITS). Berlin: Springer, 2010: 35-44

[11] Piech C, Bassen J, Huang J, et al. Deep knowledge tracing [C] //Proc of the 28th Int Conf on Neural Information Processing System(NeurIPS). Cambridge, MA: MIT, 2015: 505-513

[12] Pelanek R. Bayesian knowledge tracing, logistic models, and beyond: An overview of learner modeling techniques [J]. User Modeling and User Adapted Interaction, 2017, 27(3/4/5): 313-350

[13] Xu Pengfei, Zheng Qinhua, Chen Yaohua, et al. Research on learner modeling in education data mining [J]. Distance Education in China, 2018 (6): 5-11 (in Chinese)
(徐鹏飞, 郑勤华, 陈耀华, 等. 教育数据挖掘中的学习者建模研究[J]. 中国远程教育, 2018 (6): 5-11)

[14] Liu Hengyu, Zhang Tiancheng, Wu Peiwen, et al. A review of knowledge tracking [J]. Journal of East China Normal University: Natural Sciences, 2019(5): 1-15 (in Chinese)
(刘恒宇, 张天成, 武培文, 等. 知识追踪综述[J]. 华东师范大学学报: 自然科学版, 2019(5): 1-15)

[15] Hu Xuegang, Liu Fei, Bu Chenyang. Research advances on knowledge tracing models in educational big data [J]. Journal of Computer Research and Development, 2020, 57(12): 2523-2546 (in Chinese)
(胡学钢, 刘菲, 卜晨阳. 教育大数据中认知跟踪模型研究进展[J]. 计算机研究与发展, 2020, 57(12): 2523-2546)

[16] Zhang Nuan, Jiang Bo. Review progress of learner knowledge tracing [J]. Computer Science, 2021, 48(4): 213-222 (in Chinese)
(张暖, 江波. 学习者知识追踪研究进展综述[J]. 计算机科学, 2021, 48(4): 213-222)

[17] Khajah M, Lindsey R V, Mozer M C. How deep is knowledge tracing [C] //Proc of the 9th Int Conf on Educational Data Mining(EDM). Worcester, MA: IEDMS, 2016: 94-101

[18] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735-1780

[19] 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

[20] Candès E J, Wakin M B. An introduction to compressive sampling [J]. IEEE Signal Processing Magazine, 2008, 25(2): 21-30

[21] Xiong Xiaolu, Zhao Siyuan, Van Inwegen E G, et al. Going deeper with deep knowledge tracing [C] //Proc of the 9th Int Conf on Educational Data Mining(EDM). Worcester, MA: IEDMS, 2016: 545-550

[22] Mao Ye, Lin Chen, Chi Min. Deep learning Bayesian knowledge tracing: Student models for interventions [J]. Journal of Educational Data Mining, 2018, 10(2): 28-54

[23] Montero S, Arora A, Kelly S, et al. Does deep knowledge tracing model interactions among skills [C] //Proc of the 11th Int Conf on Educational Data Mining (EDM). Worcester, MA: IEDMS, 2018: 462-466

[24] King D R. Production implementation of recurrent neural networks in adaptive instructional systems [C] //Proc of the 22nd Int Conf on Human-Computer Interaction (HCI). Berlin: Springer, 2020: 350-361

[25] Wilson K H, Xiong Xiaolu, Khajah M, et al. Estimating student proficiency: Deep learning is not the panacea [C/OL] //Proc of the 27th Conf on Neural Information Processing Systems, Workshop on Machine Learning for Education. 2016 [2020-10-22]. http://www.rob-lindsey.com/papers/2016/nips.pdf:

[26] Doleck T, Lemay D J, Basnet R B, et al. Predictive analytics in education: A comparison of deep learning frameworks [J]. Education and Information Technologies, 2020, 25(3): 1951-1963

[27] Lalwani A, Agrawal S. Few hundred parameters outperform few hundred thousand [C] //Proc of the 10th Int Conf on Educational Data Mining(EDM). Worcester, MA: IEDMS, 2017: 448-453

[28] Wilson K H, Karklin Y, Han Bojian, et al. Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation [C] //Proc of the 9th Int Conf on Educational Data Mining(EDM). Worcester, MA: IDEMS, 2016: 539-544

[29] Ding Xinyi, Larson E C. Why deep knowledge tracing has less depth than anticipated [C] //Proc of the 12th Int Conf on Educational Data Mining(EDM). Worcester, MA: IDEMS, 2019: 282-287

[30] Ghosh A, Heffernan N, Lan A S. Context-aware attentive knowledge tracing [C] //Proc of the 26th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2020: 2330-2339

[31] Khandelwal U, He He, Qi Peng, et al. Sharp nearby, fuzzy far away: How neural language models use context [C] //Proc of the 56th Annual Meeting of the Association for Computational Linguistics(ACL). Stroudsburg, PA: ACL, 2018: 284-294

[32] Tang Gongbo, Müller M, Gonzales A R, et al. Why self-attention? A targeted evaluation of neural machine translation architectures [C] //Proc of the 2018 Conf on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA: ACL, 2018: 4263-4272

[33] Daniluk M, Rocktäschel T, Welbl J, et al. Frustratingly short attention spans in neural language modeling [C/OL] //Proc of the 5th Int Conf on Learning Representations(ICLR). 2017 [2020-10-22]. https://openreview.net/forum?id=ByIAPUcee

[34] Yeung C K, Yeung D Y. Addressing two problems in deep knowledge tracing via prediction-consistent regularization [C] //Proc of the 5th Annual ACM Conf on Learning at Scale(L@S). New York: ACM, 2018: 5:1-5:10

[35] Zhu Jia, Yu Weihao, Zheng Zetao, et al. Learning from interpretable analysis: Attention-based knowledge tracing [G] //LNCS 12164: Proc of the 21st Int Conf on Artificial Intelligence in Education (AIED). Berlin: Springer, 2020: 364-368

[36] Ji Shouling, Li Jinfeng, Du Tianyu, et al. Survey on techniques, applications and security of machine learning interpretability [J]. Journal of Computer Research and Development, 2019, 56(10): 2071-2096 (in Chinese)

(纪守领, 李进锋, 杜天宇, 等. 机器学习模型可解释性方法、应用与安全研究综述[J]. 计算机研究与发展, 2019, 56(10): 2071-2096)

[37] Liu Dong, Dai Huanhuan, Zhang Yunping, et al. Deep knowledge tracking based on attention mechanism for student performance prediction [C] //Proc of the 2nd Int Conf on Computer Science and Educational Informatization (CSEI). Piscataway, NJ: IEEE, 2020: 95-98

[38] Zhang Jiani, Shi Xingjian, King I, et al. Dynamic key-value memory networks for knowledge tracing [C] //Proc of the 26th Int Conf on World Wide Web (WWW). New York: ACM, 2017: 765-774

[39] Ai Fangzhe, Chen Yishuai, Guo Yuchun, et al. Concept-aware deep knowledge tracing and exercise recommendation in an online learning system [C/OL] //Proc of the 12th Int Conf on Educational Data Mining (EDM). 2019 [2020-10-22]. https://drive.google.com/file/d/1o8JRGKxow-d4QtM2LS2D8X2DPnq554Ix/view?usp=drive_open

[40] Su Yu, Liu Qingwen, Liu Qi, et al. Exercise-enhanced sequential modeling for student performance prediction [C] //Proc of the 32nd AAAI Conf on Artificial Intelligence (AAAI). Palo Alto, CA: AAAI, 2018: 2435-2443

[41] Huang Zhenya, Yin Yu, Chen Enhong, et al. EKT: Exercise-aware knowledge tracing for student performance prediction [J/OL]. IEEE Transactions on Knowledge and Data Engineering, 2019 [2020-10-22]. https://ieeexplore.ieee.org/abstract/document/8744302/

[42] Lee J, Yeung D Y. Knowledge query network for knowledge tracing: How knowledge interacts with skills [C] //Proc of the 9th Int Conf on Learning Analytics & Knowledge (LAK). New York: ACM, 2019: 491-500

[43] Yeung C K. Deep-IRT: Make deep learning-based knowledge tracing explainable using item response theory [C/OL] //Proc of the 12th Int Conf on Educational Data Mining (EDM). 2019 [2020-10-22]. https://drive.google.com/file/d/1B_vuJGNgdl7Wxdbh9Lz9nkM_GuxWAfGD

[44] Lu Yu, Wang Deliang, Meng Qinggang, et al. Towards interpretable deep learning models for knowledge tracing [G] //LNCS 12164: Proc of the 21st Int Conf on Artificial Intelligence in Education (AIED). Berlin: Springer, 2020: 185-190

[45] Ding Xinyi, Larson E C. Incorporating uncertainties in student response modeling by loss function regularization [J]. Neurocomputing, 2020, 409: 74-82

[46] Hu Qian, Rangwala H. Reliable deep grade prediction with uncertainty estimation [C] //Proc of the 9th Int Conf on Learning Analytics & Knowledge (LAK). New York: ACM, 2019: 76-85

[47] Abdelrahman G, Wang Qing. Knowledge tracing with sequential key-value memory networks [C] //Proc of the 42nd Int Conf on Research and Development in Information Retrieval (SIGIR). New York: ACM, 2019: 175-184

[48] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [C/OL] //Proc of the 3rd Int Conf on Learning Representations (ICLR). 2015 [2020-10-22]. http:// arxiv.org/abs/1409.0473

[49] Sha Long, Hong Pengyu. Neural knowledge tracing [G] //LNCS 10512: Proc of Int Conf on Brain Function Assessment in Learning (BFAL). Berlin: Springer, 2017: 108-117

[50] Lin Zhouhan, Feng Minwei, Santos C N, et al. A structured self-attentive sentence embedding [C/OL] //Proc of the 5th Int Conf on Learning Representations (ICLR). 2017 [2020-10-22]. https://openreview.net/forum?id=BJC_jUqxe

[51] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C] //Proc of the 30th Int Conf on Neural Information Processing Systems (NeurIPS). Cambridge, MA: MIT Press, 2017: 6000-6010

[52] Pandey S, Karypis G. A self-attentive model for knowledge tracing [C] //Proc of the 12th Int Conf on Educational Data Mining (EDM). Worcester, MA: IDEMS, 2019 [2020-10-22]. https://drive.google.com/file/d/18d_X6AXkPMhiHFQ2POarstVbX_7oMdFM

[53] Choi Y, Lee Y, Cho J, et al. Towards an appropriate query, key, and value computation for knowledge tracing [C] //Proc of the 7th ACM Conf on Learning @ Scale (L@S). New York: ACM, 2020: 341-344

[54] Pu Shi, Yudelson M, Ou Lu, et al. Deep knowledge tracing with transformers [C] //Proc of the 21st Int Conf on Artificial Intelligence in Education (AIED). Berlin: Springer, 2020: 252-256

[55] Zhang Liang, Xiong Xiaolu, Zhao Siyuan, et al. Incorporating rich features into deep knowledge tracing [C] //Proc of the 4th ACM Conf on Learning at Scale (L@S). New York: ACM, 2017: 169-172

[56] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks [J]. Science, 2006, 313(5786): 504-507

[57] Suragani G, Pothuraju L N, Reddi K S, et al. Enhancing deep knowledge tracing (DKT) model by introducing extra student attributes [C/OL] //Proc of the 5th Int Conf for Convergence in Technology (I2CT). 2019 [2020-10-22]. https://ieeexplore.ieee.org/abstract/document/9033598

[58] Yang Haiqin, Cheung L P. Implicit heterogeneous features embedding in deep knowledge tracing [J]. Cognitive Computation, 2018, 10(1): 3-14

[59] Sun Xia, Zhao Xu, Ma Yuan, et al. Multi-behavior features based knowledge tracking using decision tree improved DKVMN [C/OL] //Proc of the ACM Turing Celebration Conf. 2019 [2020-10-22]. https://dl.acm.org/doi/abs/10.1145/3321408.3322847

[60] Nagatani K, Zhang Qian, Sato M, et al. Augmenting knowledge tracing by considering forgetting behavior [C] //Proc of the Int World Wide Web Conf. New York: ACM, 2019: 3101-3107

[61] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C] //Proc of the 26th Int Conf on Neural Information Processing Systems (NeurIPS). Cambridge, MA: MIT, 2013: 3111-3119

[62] Cheng Song, Liu Qi, Chen Enhong. Domain adaption for knowledge tracing [J]. arXiv preprint, arXiv: 2001.04841, 2020

[63] Tato A, Nkambou R. Some improvements of deep knowledge tracing [C] //Proc of the 31st Int Conf on Tools with Artificial Intelligence (ICTAI). Piscataway, NJ: IEEE, 2019: 1520-1524

[64] Devlin J, Chang Mingwei, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding [C] //Proc of the 2019 Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). Stroudsburg, PA: ACL, 2019: 4171-4186

[65] Huo Yujia, Wong D F, Ni L M, et al. Knowledge modeling via contextualized representations for LSTM-based personalized exercise recommendation [J]. Information Sciences, 2020, 523: 266-278

[66] Minn S, Yu Yi, Desmarais M C, et al. Deep knowledge tracing and dynamic student classification for knowledge tracing [C] //Proc of the Int Conf on Data Mining (ICDM). Piscataway, NJ: IEEE, 2018: 1182-1187

[67] Minn S, Desmarais M C, Zhu Ferida, et al. Dynamic student classification on memory networks for knowledge tracing [G] //LNCS 11440: Proc of the Pacific-Asia Conf on Knowledge Discovery and Data Mining (PAKDD). Berlin: Springer, 2019: 163-174

[68] Chaudhry R, Singh H, Dogga P, et al. Modeling hint-taking behavior and knowledge state of students with multi-task learning [C/OL] //Proc of the 11th Int Conf on Educational Data Mining (EDM). 2018 [2020-10-22]. http://educationaldatamining.org/files/conferences/EDM2018/papers/EDM2018_paper_100.pdf

[69] Chen Penghe, Lu Yu, Zheng V W, et al. Prerequisite-driven deep knowledge tracing [C] //Proc of the Int Conf on Data Mining (ICDM). Piscataway, NJ: IEEE, 2018: 39-48

[70] Wang Zhiwei, Feng Xiaoqin, Tang Jiliang, et al. Deep knowledge tracing with side information [G] //LNCS 11626: Proc of the Int Conf on Artificial Intelligence in Education (AIED). Berlin: Springer, 2019: 303-308

[71] Tang Jian, Qu Meng, Wang Mingzhe, et al. LINE: Large-scale information network embedding [C] //Proc of the 24th Int Conf on World Wide Web (WWW). New York: ACM, 2015: 1067-1077

[72] Grover A, Leskovec J. Node2Vec: Scalable feature learning for networks [C] //Proc of the 22nd Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2016: 855-864

[73] Wang Tianqi, Ma Fenghong, Gao Jing. Deep hierarchical knowledge tracing [C/OL] //Proc of the 12th Int Conf on Educational Data Mining (EDM). 2019 [2020-10-22]. https://drive.google.com/file/d/1wlW6zAi-l4ZAw8rBA_mXZ5tHgg6xKL00

[74] Sonkar S, Waters A E, Lan A S, et al. qDKT: Question-centric deep knowledge tracing [C] //Proc of the 13th Int Conf on Educational Data Mining (EDM). 2020 [2020-10-22]. https://educationaldatamining.org/files/conferences/EDM2020/papers/paper_35.pdf

[75] Nakagawa H, Iwasawa Y, Matsuo Y. Graph-based knowledge tracing: Modeling student proficiency using graph neural network [C] //Proc of the Int Conf on Web Intelligence (WI). Piscataway, NJ: IEEE, 2019: 156-163

[76] Shen Shuanghong, Liu Qi, Chen Enhong, et al. Convolutional knowledge tracing: Modeling individualization in student learning process [C] //Proc of the 43rd Int Conf on Research and Development in Information Retrieval (SIGIR). New York: ACM, 2020: 1857-1860

[77] Dauphin Y N, Fan A, Auli M, et al. Language modeling with gated convolutional networks [C] //Proc of the 34th Int Conf on Machine Learning (ICML). Cambridge, MA: JMLR, 2017: 933-941

[78] Vie J J. Deep factorization machines for knowledge tracing [C] //Proc of the 13th Workshop on Innovative Use of NLP for Building Educational Applications @ NAACL-HLT. Stroudsburg, PA: ACL, 2018: 370-373

[79] Guo Huifeng, Tang Ruiming, Ye Yunming, et al. DeepFM: A factorization-machine based neural network for CTR prediction [C] //Proc of the 26th Int Joint Conf on Artificial Intelligence (IJCAI). Palo Alto, CA: AAAI, 2017: 1725-1731

[80] Gan Wenbin, Sun Yuan, Peng Xian, et al. Modeling learner's dynamic knowledge construction procedure and cognitive item difficulty for knowledge tracing [J]. Applied Intelligence, 2020, 50(11): 3894-3912

[81] Xu Liangbei, Davenport M A. Dynamic knowledge embedding and tracing [C] //Proc of the 13th Int Conf on Educational Data Mining (EDM). 2020 [2020-10-22]. https://educationaldatamining.org/files/conferences/EDM2020/papers/paper_79.pdf

[82] Li Donghua, Jia Yanming, Jian Zhou, et al. Deep knowledge tracing based on Bayesian neural network [C] //Proc of the Int Conf on Intelligent and Interactive Systems and Applications (IISA). Berlin: Springer, 2019: 29-37

[83] Nakagawa H, Iwasawa Y, Matsuo Y. End-to-end deep knowledge tracing by learning binary question-embedding [C] //Proc of the Int Conf on Data Mining Workshops (ICDMW). Piscataway, NJ: IEEE, 2018: 334-342

[84] Zhang Jiani, King I. Topological order discovery via deep knowledge tracing [C] //Proc of the 23rd Int Conf on Neural Information Processing (ICONIP). Berlin: Springer, 2016: 112-119

[85] Kahn A B. Topological sorting of large networks [J]. Communications of the ACM, 1962, 5(11): 558-562

[86] Wang L, Sy A, Liu L, et al. Deep knowledge tracing on programming exercises [C] //Proc of the 4th ACM Conf on Learning at Scale(L@S). New York: ACM, 2017: 201-204

[87] Piech C, Huang J, Nguyen A, et al. Learning program embeddings to propagate feedback on student code [C] //Proc of the 32nd Int Conf on Machine Learning (ICML). Cambridge, MA: JMLR, 2015: 1093-1102

[88] Socher R, Perelygin A, Wu J Y, et al. Recursive deep models for semantic compositionality over a sentiment treebank [C] //Proc of the 2013 Conf on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA: ACL, 2013: 1631-1642

[89] Swamy V, Guo A, Lau S, et al. Deep knowledge tracing for free-form student code progression [C] //LNCS 10948: Proc of the 19th Int Conf on Artificial Intelligence in Education (AIED). Berlin: Springer, 2018: 348-352

[90] Lalwani A, Agrawal S. Validating revised Bloom's taxonomy using deep knowledge tracing [C] //Proc of the 19th Int Conf on Artificial Intelligence in Education (AIED). Berlin Springer, 2018: 225-238

[91] Lalwani A, Agrawal S. What does time tell? Tracing the forgetting curve using deep knowledge tracing [C] //Proc of the 20th Int Conf on Artificial Intelligence in Education (AIED). Berlin: Springer, 2019: 158-162

[92] Cai Dejun, Zhang Yuan, Dai Bintao. Learning path recommendation based on knowledge tracing model and reinforcement learning [C] // Proc of the 5th Int Conf on Computer and Communications (ICCC). Piscataway, NJ: IEEE, 2019: 1881-1885

[93] Yeung C K, Yeung D Y. Incorporating features learned by an enhanced deep knowledge tracing model for stem/non-stem job prediction [J]. International Journal of Artificial Intelligence in Education, 2019, 29(3): 317-341

[94] Wu Zhengyang, He Tao, Mao Cenjie, et al. Exam paper generation based on performance prediction of student group [J]. Information Sciences, 2020, 532: 72-90

[95] Mongkhonvanit K, Kanopka K, Lang D. Deep knowledge tracing and engagement with MOOCs [C] //Proc of the 9th Int Conf on Learning Analytics & Knowledge (LAK). New York: ACM, 2019: 340-342

[96] Huan Wei, Li Haotian, Xia Meng, et al. Predicting student performance in interactive online question pools using mouse interaction features [C] //Proc of the 10th Int Conf on Learning Analytics & Knowledge (LAK). New York: ACM, 2020: 645-654

[97] Xu Bin, Yan Sheng, Yang Dan. BiRNN-DKT: Transfer Bi-directional LSTM RNN for knowledge tracing [C] //Proc of the 16th Int Conf on Web Information Systems and Applications (WISA). Berlin: Springer, 2019: 22-27

Liu Tieyuan, born in 1984. PhD candidate. Lecturer. Member of CCF. His main research interests include educational data mining, deep learning, machine learning.

刘铁园,1984 年生.博士研究生,讲师,CCF 会员.主要研究方向为教育数据挖掘、深度学习和机器学习.

Chen Wei, born in 1997. Master candidate. His main research interests include educational data mining, deep learning, machine learning.

陈 威,1997 年生.硕士研究生.主要研究方向为教育数据挖掘、深度学习和机器学习.

Chang Liang, born in 1980. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include data and knowledge engineering, formal methods, and trusted software.

常 亮,1980 年生.博士,教授,博士生导师,CCF 高级会员.主要研究方向为数据和知识工程、形式化方法和可信软件.

Gu Tianlong, born in 1964. PhD, professor, PhD supervisor. Senior member of CCF. His research interests include formal methods, trusted artificial intelligence and data mining.

古天龙,1964 年生.博士,教授,博士生导师,CCF 高级会员.主要研究方向为形式化方法、可信人工智能和数据挖掘.