Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

个性化学习路径推荐—量身打造专属于你的学习Plan

硕士研究生 杨晓楠

2024年04月14日

问题回溯

- 总结反思
  - 公式讲解过于抽象，缺乏图形化和流程化的辅助表述
  - 语气稍显平淡，听感不佳
  - 问题回答重点不突出，表述不清

- 相关内容
  - 2024.03.31 杨宗源：《LLM的强化学习》
  - 2023.09.03 杨晓楠：《认知诊断技术及其研究》
  - 2022.03.28 周瑾洁：《从赋能学习到知识追踪》
  - 2022.03.28 门元昊：《强化学习基础与实战》

内容提要

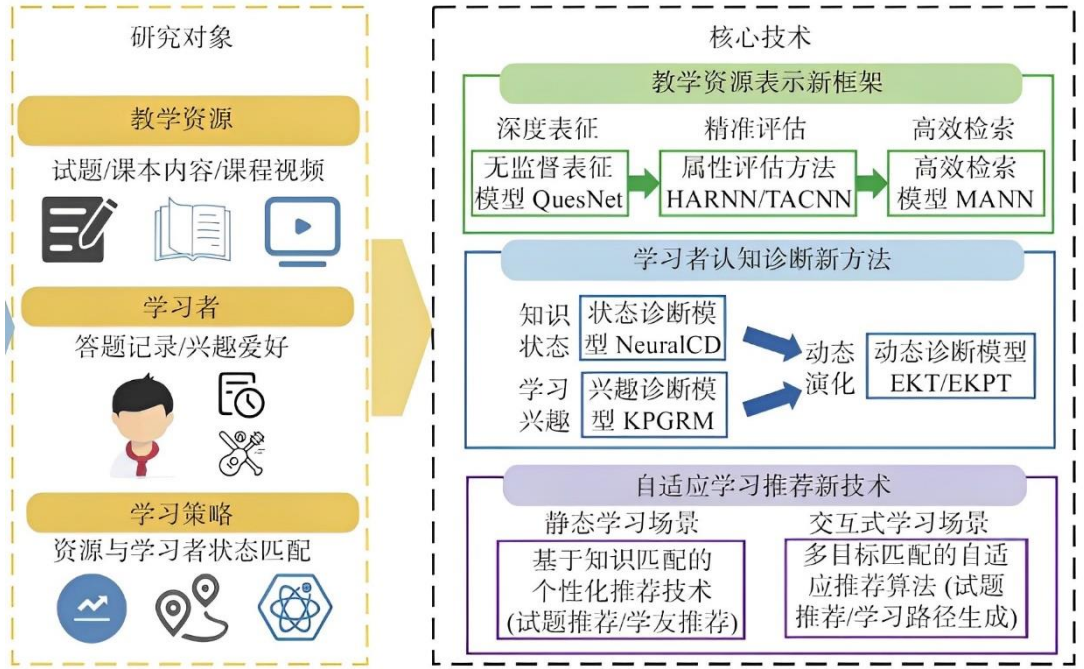
- 预期收获
- 题目内涵解析
- 研究背景与意义
- 研究历史与现状
- 知识基础
- 算法原理
  - GEHRL
  - cDQN-PathRec
- 特点总结与工作展望
- 参考文献

预期收获

- 预期收获
  - 1. 了解个性化学习路径推荐的基本概念和研究方法
  - 2. 理解强化学习技术在学习路径推荐的应用原理
  - 3. 了解个性化学习路径推荐的发展前景

目标内涵

- 研究目标
  - 以个性化学习策略为研究对象，面向学生的学习路径推荐任务
  - 结合知识图谱、强化学习、知识追踪、认知诊断等技术
  - 为每个学生量身定制学习路径，以最小的学习成本最大限度地完成学习目标



- 内涵解析
  - 个性化学习：基于个体学习者的特征、需求和背景，为其量身定制学习体验的教育理念和实践
  - 学习路径：以达到学习目标为导向，学生在学习过程中一系列学习活动和学习资源的有序组合

背景意义

- 研究背景
  - 在线教育的蓬勃发展为学习者提供了更加便捷、高效的学习环境
  - 传统教学模式忽略了学习者知识背景、学习能力的差异及学习目标的多样性，并不能充分满足学习者的个性化需求

- 研究意义
  - 深入探索学习者的动机、学习风格等因素对学习成效的影响，为教育实践提供了理论指导
  - 自动识别学习者特征，高效分配学习资源，为每一位学习者规划合理的学习路径，提高学习效率，具有重要的应用价值

研究历史

2017 Nabizadeh等人采用深度优先搜索算法寻找满足要求的所有个性化学习路径，然后以时间限制为基础进行剪枝操作。

2018 Shu等人使用CNN从学习资源的文本信息中挖掘潜在因素，并将文本信息转化为学习材料的特征进行推荐。

2019 Xia等人利用马尔可夫决策过程进一步考虑了前后路径节点的关联关系，根据其同伴的启发互动地规划合适的学习路径。

2019 Liu等人提出基于认知结构增强的自适应学习框架，应用Actor-Critic算法依次为不同的学习者识别正确的学习项目。

2020 Shi等人提出基于多维知识图谱框架的学习路径推荐模型，根据学习者的目标学习对象生成并推荐定制的学习路径。

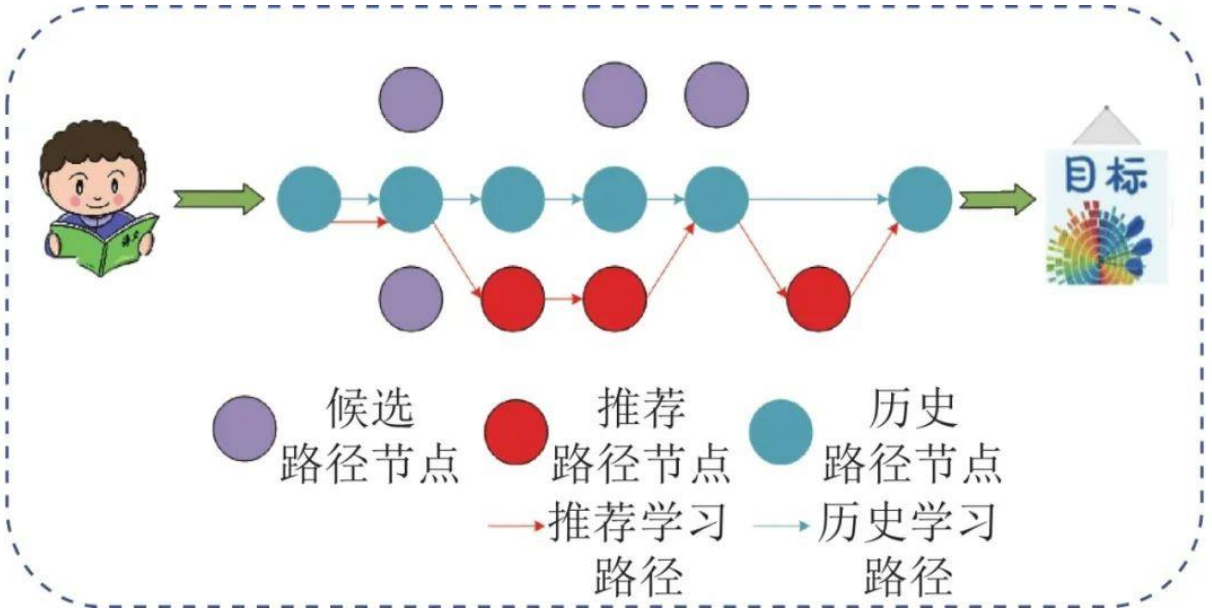
2021 Elshani等人基于遗传算法从学习资源顺序的角度切入，设定合适的选择算子与变异算子完成最优个性化学习路径的推荐任务。

2023 Ren等人开发了一个基于自关注网络的2层多目标运动推荐框架，以捕捉学生知识获取的变化，从而提供定制化的运动推荐。

2023 Bai等人提出了一种基于知识图谱的细粒度、多上下文感知的学习路径推荐模型,使用多维知识图加强全局背景下知识的远程关系。

知识基础 个性化学习路径推荐

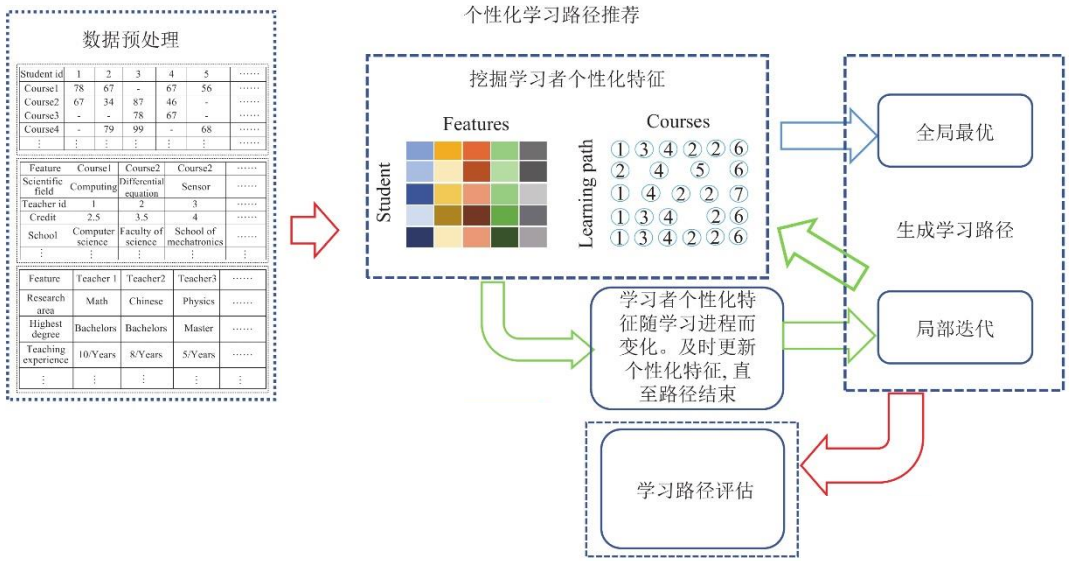
- 基本概念
  - 基于学习者的学习能力、知识背景、学习目标等的差异性，利用技术手段和数据分析，为学习者量身定制一条切实可行、高效科学的学习路径
- 基本要素
  - 学习者s：描述学习者个性化特征
  - 学习目标g：需要完成的学习任务
  - 学习资源i：由知识点映射组成的有序向量
    - 类型：文本、图片、视频等
    - 粒度：课程、章节、知识单元、知识点



为什么学？学什么？怎么学？

知识基础 个性化路径推荐

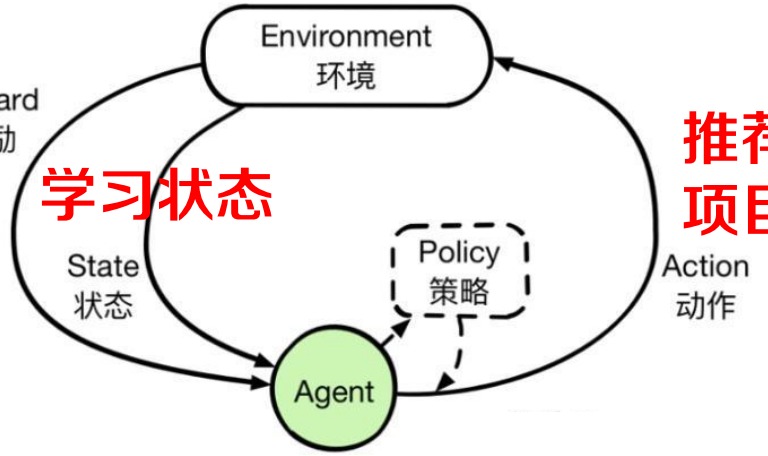
- 基本框架
  - 学习者个性化特征挖掘
    - 个性化学习目标
    - 学习能力、学习背景
    - 学习的方式与风格
  - 学习路径生成
    - 全局最优路径：专注于某阶段学习后最终的学习成果
    - 局部迭代路径：专注于知识的掌握程度随着学习过程递进的变化
  - 学习路径评估
    - 线上评估：对比分析、案例分析、问卷调查、教育模拟器
    - 线下评估：基于信息论的测量标准、基于相似性、基于优秀学习者的学习路径



知识基础 强化学习

- 基本架构
  - 两个角色
    - Agent：作为决策者
    - 环境：Agent所处的外部系统
  - 三个要素
    - 动作：Agent的输出
    - 状态：Environment所描述对象的情况
    - 奖励：Agent的动作带来的实时收益

学习者模型



学习者反馈

学习状态

推荐的学习项目或路径

推荐模型

学习路径推荐任务中是如何对应的?

- 经典模型
  - DQN：用神经网络来学习Q值函数，通过最大化动作价值函数来优化策略
  - Actor-Critic：结合策略评估（Critic，评估动作）和策略改进（Actor，选择动作）
  - PPO：基于策略梯度，通过限制策略更新的大小来稳定训练

算法原理 GEHRL

GEHRL

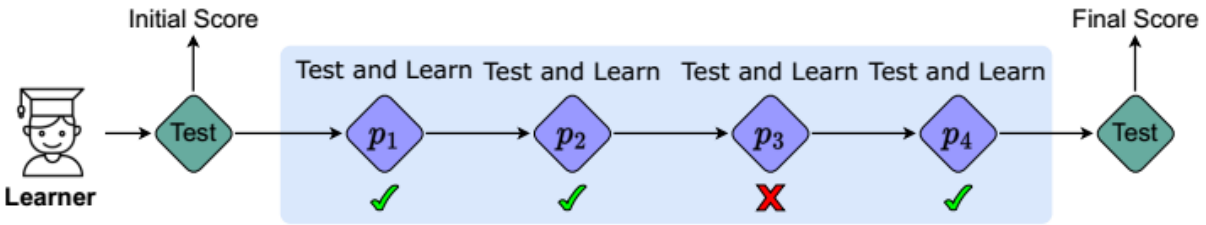
算法原理 TIPO

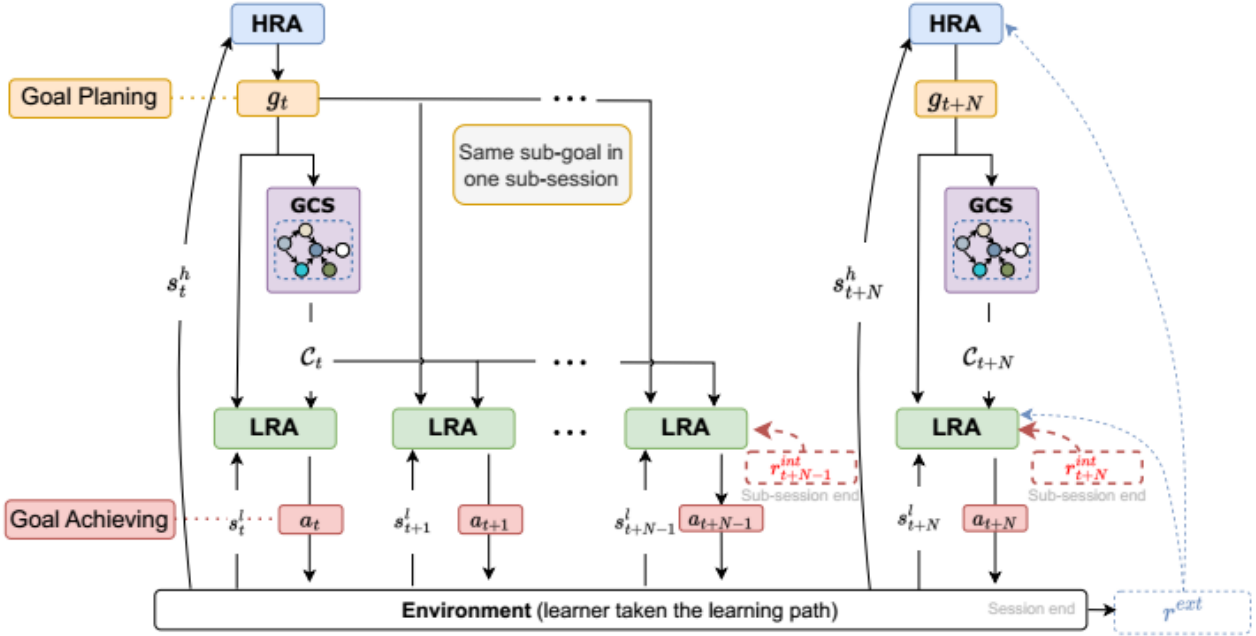
T	目标	利用分层强化学习提高学习路径推荐的准确性和有效性
I	输入	ASSITments2015数据集（学习交互记录*2.4M，学习项目*100） Junyi数据集（学习交互记录*21M，学习项目*835）
P	处理	1.基于高级代理的子目标选择 2.基于图模型的学习项目候选选择 3.基于低级代理的子目标实现
O	输出	个性化学习路径（长度为M的学习项目序列）

P	问题	缺乏合理目标规划：需要在较大的搜索空间中探索才能实现目标 目标实现效率低下：规划路径可能包含与目标无关的学习项目
C	条件	具有先决关系的项目集合；基于IRT和DKT构建的模拟器
D	难点	如何对目标达成做出合理的规划；如何避免推荐与目标无关的学习项目
L	水平	CIKM 2023 CCF B

算法原理 算法原理图

- 定义：
  - 项目集：$I = \{p_1, p_2, \dots, p_K\}$
  - 目标集：$G = \{g_1, g_2, \dots\}$
  - 学习路径：$P = \{p_1, p_2, \dots, p_M\}$
- 核心思想
  - 将目标规划为多个子目标顺序实现
- 算法步骤
  - 子目标选择
    - 为低级代理提供子目标
  - 子目标实现
    - 为学习者推荐项目以实现子目标
  - 学习项目候选选择





算法原理 子目标选择

- 高级状态编码
  - 高级状态：包含学习者历史学习记录信息和学习目标

$$s_t^h = Concat(h_t^H, h_t^G)$$

- 高级推荐代理(HRA)
  - 目标：为低级代理提供合适的子目标
  - Actor：将高级状态输入到一个全连接层，输出学习项的概率分布，并从中采样子目标

$$g_t \sim \pi^h\left(g_t \middle| s_t^h; \theta^h\right) = Softmax(FC\left(s_t^h\right))$$

  - Critic：将高级状态输入到一个全连接层，给出状态的估计返回值

$$V^h\left(s_t^h; \phi^h\right) = FC(s_t^h)$$

算法原理 子目标实现

- 低级状态编码
  - 低级状态：包含学习者历史学习记录信息、学习目标和子目标

$$s_t^l = Concat(h_t^H, h_t^G, h_t^g)$$

- 低级推荐代理(LRA)
  - 目标：进行学习项目推荐以实现特定子目标
- 更新策略
  - 高级代理每个子会话获得一次转换，低级代理每步获得一次转换
  - 在相同的交互之后，低级代理可以比高级代理训练更多的数据
  - 对于高级代理，低级代理是环境的一部分，影响高级代理面临的状态转换

算法原理 学习项目候选选择

- 基于子树的候选选择
  - 构建以学习项目为节点的有向知识图
  - 提取子目标节点的前提节点，作为低级代理的动作候选

$$C^{subtree} = prerequisite(subgoal)$$

- 基于图嵌入的候选选择
  - 使用node2vec生成对学习项目和子目标节点嵌入
  - 在嵌入空间中提取离子目标更近的top C个相关项目

$$C^{emb} = topC_{item \in I}(-\| emb(subgoal) - emb(item) \|_2^2)$$

算法原理 奖励机制

- 外部奖励
  - 评估学习者对整体目标的掌握程度
  - 高级代理的奖励等同外部奖励，对于每个时间步，奖励设置为

$$r_t^h = r_t^{ext} = \begin{cases} r_{session}^{ext} & if\ t\ is\ the\ last\ time\ step \\ 0 & otherwise \end{cases}$$

- 内部奖励
  - 评估学习者对子目标的掌握程度
  - 利用DKT的预测结果来估计学习者的掌握水平，当预测大于阈值，认为学习者已经达到了子目标

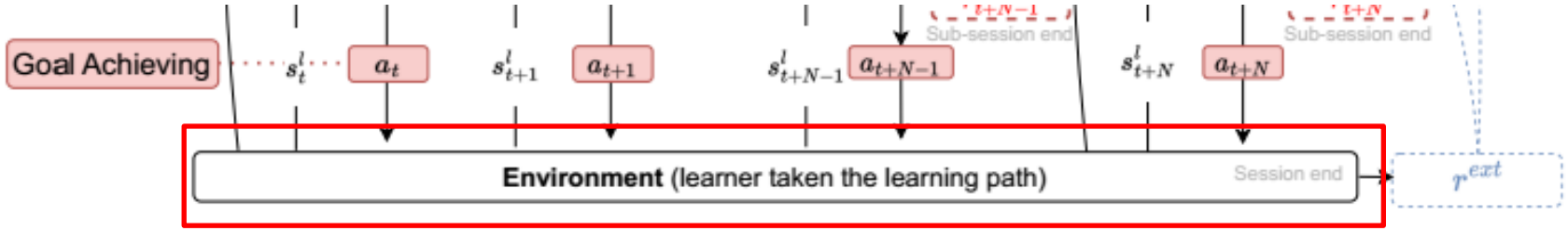
$$r_t^{int} = \begin{cases} 1, & DKT[subgoal] > \delta_{thre} \\ 0 & otherwise \end{cases}$$

  - 低级代理的奖励表示为内部奖励和外部奖励的组合

$$r_t^l = \alpha r_t^{int} + \beta r_t^{ext}$$

算法原理 环境模拟

- 教育模拟器
  - 问题：如何分析未包含在某个项目序列中的项目是否可以正确回答
  - 目标：模拟在某一学习项目上的表现，用作评估学习路径或训练代理的环境
  - 基于知识结构的模拟器（KSS）
  - 学习者在知识项目上的表现是基于项目反应理论（IRT）来测量的
  - 基于知识进化的模拟器（KES）
    - 采用DKT来模拟学习者的知识状态变化和在学习项目上的表现
    - 使用整个数据集来训练模拟器的DKT，其中的50%训练计算内部奖励的DKT



实验设计 数据集及对比方法

- 数据集
  - Junyi（超过2100万学习记录）
    - 包含项目先决条件
  - ASSITments2015（超过242万学习记录）
    - 未包含项目先决条件，自行构建
- 评价指标
  - 学习目标的提升度

Dataset	Junyi	ASSISTments2015
#exercises	835	100
#learners	525,061	69,675
#records	21,460,249	2,420,200
attempts per question	16.42	5.49
positive label rate	54.38%	73.17%

$$E_P = \frac{E_{end} - E_{start}}{E_{sup} - E_{start}}$$

- 对比方法
  - 非强化学习算法：KNN、GRU4Rec
  - 强化学习算法：DQN、SAC、Actor-Critic、 CB、 RLTutor、 CSEAL

实验设计 性能对比实验

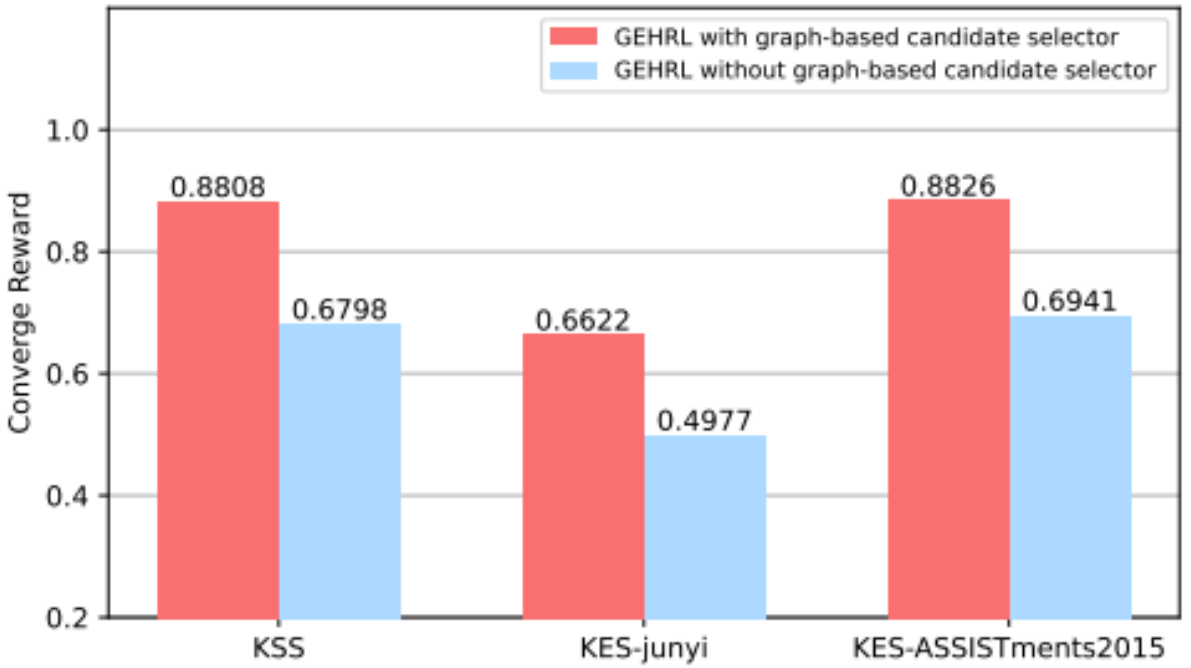
- 实验结果
  - GEHRL-EB优于所有基线，表明利用层次强化学习进行学习路径推荐中的有效性
  - 随着Step的增大，强化学习方法对比非强化学习方法的优势增加
  - 在KES-junyi中出现负值，分析由于模拟器中的DKT遇到未训练项目，预测不稳定
  - GEHRL-EB和GEHRL-ST的结果表明以合理的方式约束动作空间可以获得更优解

		KNN	GRU4Rec	DQN	SAC	Actor-Critic	CB	RLTutor	CSEAL	GEHRL-ST	GEHRL-EB
KSS	Step = 5	0.1607	0.1792	0.2627	0.2714	0.2385	0.1355	0.2947	0.3313	0.3062	**0.3784**
	Step = 10	0.4227	0.4133	0.2701	0.3692	0.4811	0.3610	0.5108	0.5175	0.6089	**0.6241**
	Step = 20	0.3851	0.1905	0.3855	0.2665	0.5051	0.3891	0.6227	0.6833	0.8333	**0.8808**
KES-junyi	Step = 5	0.0102	-0.0974	-0.1069	-0.4647	0.2852	0.1147	-0.0797	0.2401	-0.0342	**0.2868**
	Step = 10	-0.1205	-0.1227	-0.1386	-0.2873	0.1259	0.1691	-0.1037	0.3071	0.0269	**0.3463**
	Step = 20	0.1405	0.0011	0.2105	0.2140	0.3183	0.3128	-0.1306	0.3942	0.5626	**0.6622**
KES-ASSIST15	Step = 5	0.3544	0.3592	0.4214	0.5459	0.4327	0.4986	0.5623	0.5621	0.5751	**0.5858**
	Step = 10	0.2586	0.2433	0.5230	0.4817	0.5913	0.5394	0.7069	0.6954	0.6707	**0.7508**
	Step = 20	0.1644	0.1025	0.3675	0.1543	0.7282	0.6413	0.8713	0.8015	0.8352	**0.8826**

实验设计 消融实验

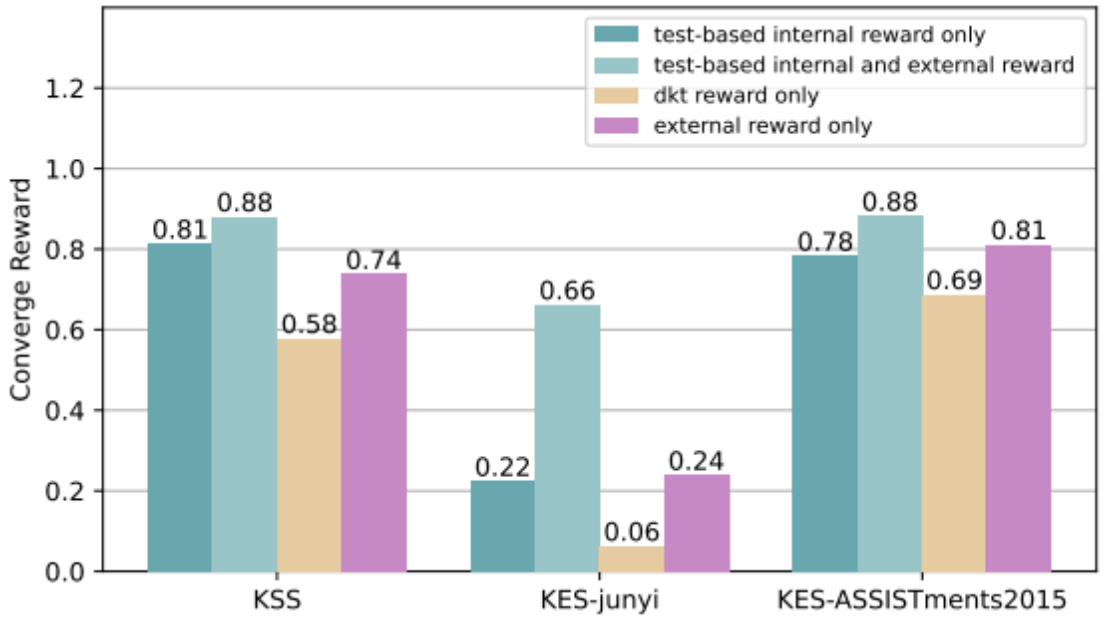
- 基于图的候选选择器的影响
  - 基于图形的候选选择器的框架具有更好的性能
  - 可以减少RL代理的搜索空间，更容易找到合适的策略

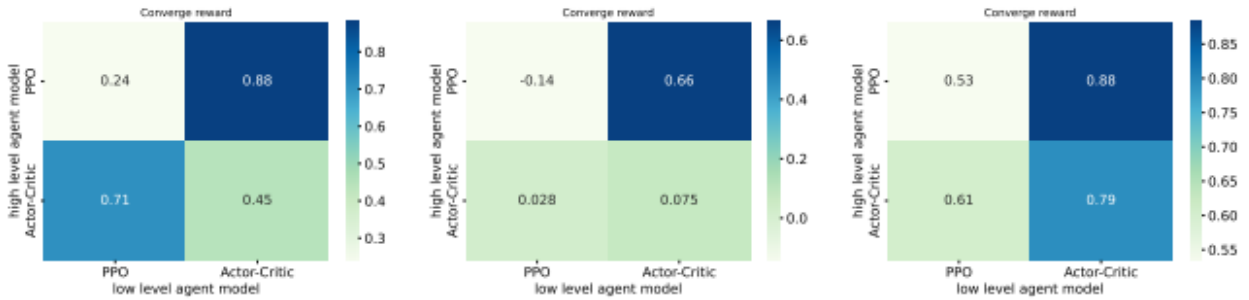
	Total space	CSEAL	Ours
KSS	10^{10}	3.4×10^{6}	5.9×10^{4}
KES-junyi	835^{10}	5.3×10^{15}	5.9×10^{14}
KES-ASSIST15	100^{10}	5.5×10^{10}	1×10^{10}



实验设计 消融实验

- 不同奖励及代理模型的影响
  - 单一内部奖励或外部奖励的训练都不能达到较好的效果，这证明低级代理应该考虑子目标和学习目标的提高
  - 先进的高级策略与简单的低级策略相结合，低级代理可以在一个会话中获得更多的训练样本，因此性能最好

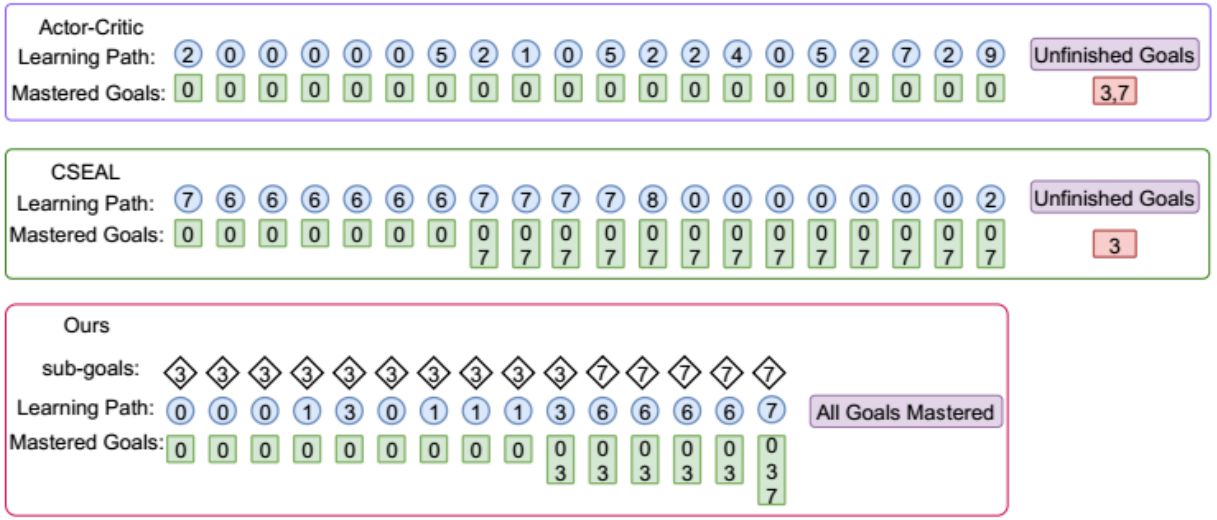


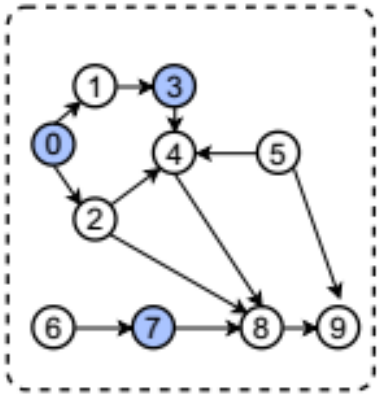


(a) KSS (b) KES-junyi (c) KES-ASSIST

实验设计 案例研究

- 案例任务
  - 基于KSS模拟器，20步内为学习者推荐学习路径，学习目标为0、3和7
- 实验结果
  - Actor-Critic：推荐路径不合理，未达成目标3和目标7
  - CSEAL：包含与目标无关的学习项目的低效路径，未达成目标3
  - GEHRL：更短的推荐路径实现了所有学习目标





算法原理 cDQN-PathRec

cDQN-PathRec

算法原理 TIPO

T	目标	基于强化学习利用多行为建模提高学习路径推荐的准确性和有效性
I	输入	课程知识图谱和学生交互日志（交互记录*58k，学习项目*545）
P	处理	1.多行为Transformer进行学习者建模 2.计算top k的学习资源候选 3.级联DQN结合加权奖励机制进行学习路径推荐
O	输出	个性化学习路径（长度为M的学习项目序列）

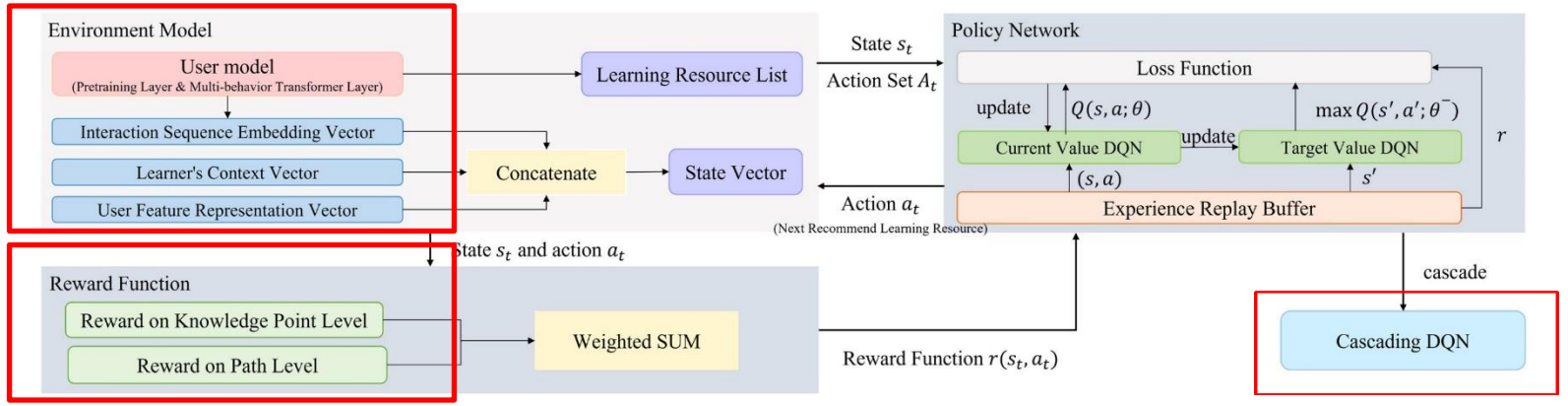
P	问题	现有方法未考虑在线学习者不同学习行为与学习资源之间的潜在相关性
C	条件	存在学习者不同学习行为记录
D	难点	建模不同行为之间的时间信息、语义信息和高阶关系
L	水平	KBS 2024 SCI 1区

算法原理 算法原理图

- 学习行为
  - 输入、查看、标记、提交
  - 代表用户与学习资源交互的不同程度的意愿

Learning behavior	Operator count	Count	Percentage
Enter	7	68,121	52.7%
Watch	7	12,918	10.0%
Tag	6	45,499	35.2%
Submit	5	2663	2.1%

- 算法框架
  - 构建多行为Transformer建模多用户行为（环境、状态）
  - 级联DQN推荐学习项目（代理、动作）
  - 知识点级和路径级奖励加权结合（奖励）



算法原理 多用户行为建模

- 多行为序列模式挖掘
  - 考虑序列中两个学习资源之间的时间间隔，采用桶式机制相对位置编码
  - 桶函数表示为

$$f(j-i) = h(b(j-i)$$

$$b(j-i) = \begin{cases} j-i & j \geq i \\ i-j+\frac{1}{2}N_B & j < i \end{cases}$$

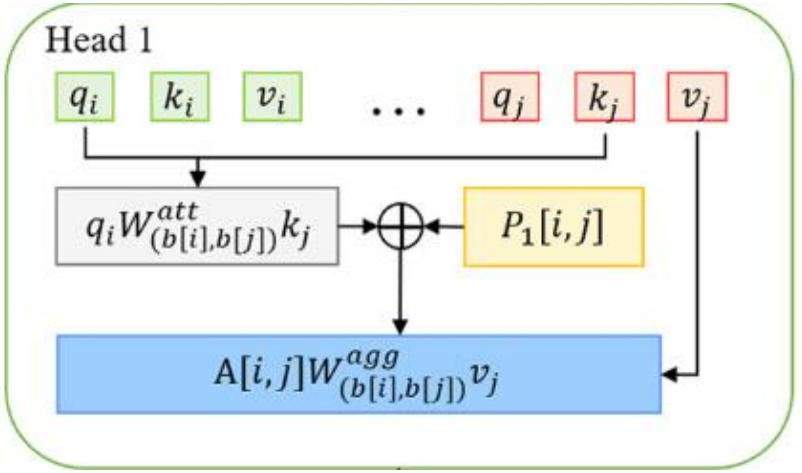
  - 同时，考虑序列中对应学习资源的相似性，使用余弦相似度计算学习资源嵌入向量之间的距离

$$sim(i,j) = \frac{emb_i \cdot emb_j}{\| emb_i \| \cdot \| emb_j \|}$$

$$\mathrm{P}[k,i,j] = f_{(\mathrm{b}[i],\mathrm{b}[j])}(j-i) \cdot sim(i,j)$$

算法原理 多用户行为建模

- 多行为多头自注意力机制
  - 使用特定于行为的线性投影计算查询、键和值
  - 考虑行为异质性，纳入行为相关权重来计算注意力得分



  - 将计算的序列模式表示分数与注意分数融合，得到最终的注意权重
  - 在聚合价值信息时，通过纳入行为相关权重来考虑行为异质性
  - 使用层归一化连接多头自注意机制和MLP层

$$\begin{cases} \mathbf{G}^{(l)} = \text{LayerNorm}(\mathbf{H}^{(l-1)} + MBMA(\mathbf{H}^{(l-1)}, \mathbf{b}, \mathbf{P}^{(l)})) \\ \mathbf{H}^{(l)} = \text{LayerNorm}(\mathbf{G}^{(l)} + B - MLP(\mathbf{G}^{(l)}, \mathbf{b})) \end{cases}$$

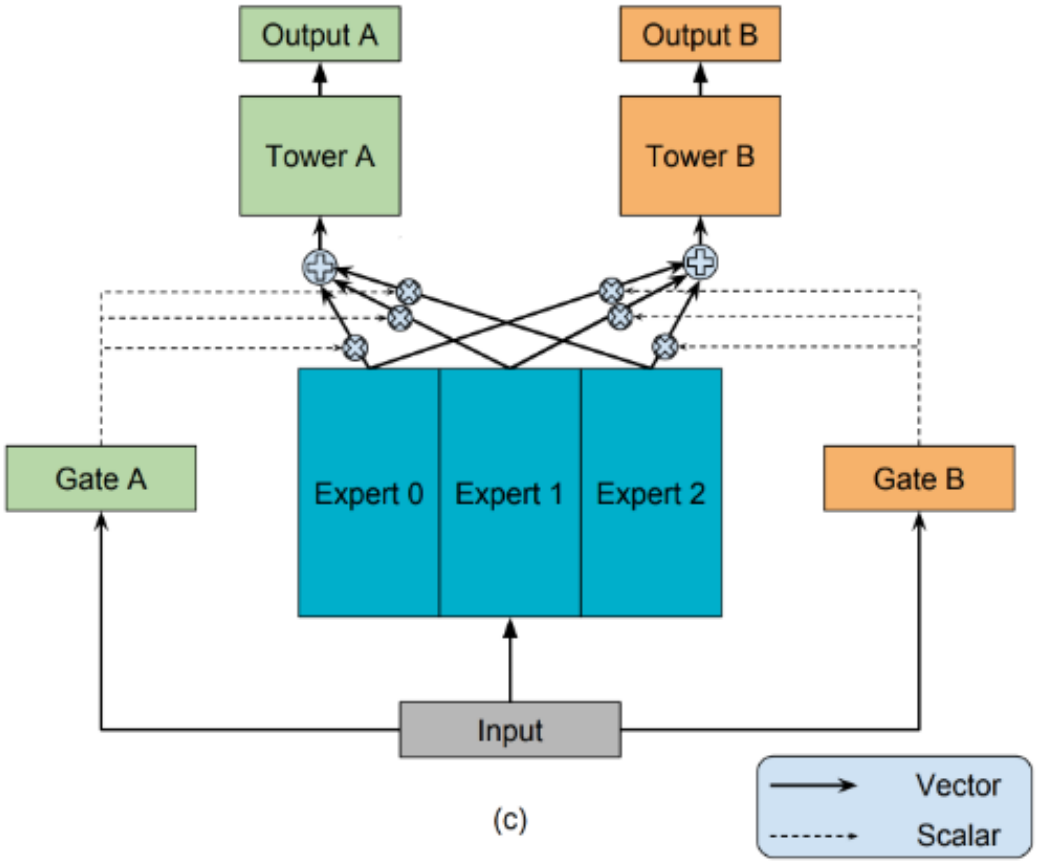
算法原理 多用户行为建模

- 特定行为的多层感知器
  - 从用户的多行为交互序列中捕获提取有效的多行为交互信息
  - 采用MMoE模型，为每种行为都单独使用一个门控网络计算得到不同的专家权重

$$\sigma_i = g_k(\mathrm{H}^{(L)}[i])^{\mathrm{T}} E_k(\mathrm{H}^{(L)}[i])$$

$$g_k(x) = softmax(W_g^k x)$$

$$E_k(x) = [e_{k,1}(x), \ldots, e_{k,n_b}(x), e_{s,1}(x), \ldots, e_{s,n_s}(x)]$$



算法原理 多用户行为建模

- 学习者状态表示
  - 学习语境：基于特定的时间阈值和交互频率限制

Features and constraints	Average number of interactions with learning resources < 2	Average number of interactions with learning resources ≥ 2
Time to exam > 7 days	initial learning	daily review
Time to exam ≤ 7 days	pre-exam learning	pre-exam review

  - 结合语境向量、交互序列嵌入向量和用户特征表示向量

$$s_t = Concat(v_{scene}, v_{seq}, v_{user})$$

- 学习资源候选
  - 使用softmax函数预测交互概率，并以此选择top k的学习资源

$$p_i(v) = Softmax(\sigma_i \cdot E^T)$$

算法原理 奖励函数

- 奖励函数
  - 同时考虑路径推荐的短期局部收益和长期全局收益

$$r(s_t, a_t) = (1-\alpha) \cdot r_{unit}(s_t, a_t) + \alpha \cdot r_{seq}(s_t, a_t)$$

  - 知识点级奖励函数
    - 衡量每一次交互所推荐的学习资源的准确性
    - 计算预测和真实学习资源的嵌入表示向量之间的余弦相似度

$$r_{unit}(s_t, a_t) = \frac{emb_{predict} \cdot emb_{actual}}{\| emb_{predict} \| \cdot \| emb_{actual} \|} \in (0,1]$$

  - 路径级奖励函数
    - 衡量整体交互所推荐的学习资源的准确性
    - 使用BLEU度量计算两个序列内子序列的相似度

$$r_{seq}(s_t, a_t) = \exp\left(\sum_{i=1}^{m} \log \frac{\sum_k^i \sum_j^{i-k} equal\left(seq_{actual}[j:j+k], seq_{predict}[j:j+k]\right)}{\sum_k^i \sum_j^{i-k} num(seq_{actual}[j:j+k])}\right)$$

算法原理 级联DQN

- 级联DQN
  - 减少计算时间，平衡每一步与整个路径的最优化，提高学习路径推荐的效率
  - 结合多个深度Q网络，共同求解路径上每一步的最优动作策略

$$
\begin{cases}
a_1{}^* = \underset{a_1}{\operatorname{argmax}}\left\{ Q_1^*(s, a_1) := \underset{a_{2:m}}{max} Q^*(s, a_{1:m}) \right\} \\
a_2{}^* = \underset{a_2}{\operatorname{argmax}}\left\{ Q_2^*(s, a_1^*, a_2) := \underset{a_{3:m}}{max} Q^*(s, a_{1:m}) \right\} \\
\cdots \\
a_m{}^* = \underset{a_m}{\operatorname{argmax}}\left\{ Q_m^*(s, a_{1:m-1}^*, a_m) := Q^*(s, a_{1:m}) \right\}
\end{cases}
$$

  - 最终得到由一系列学习资源组成的序列

$$[a_1{}^*, a_2{}^*, \ldots, a_m{}^*]$$

实验设计 数据集

- 数据集
  - MOOPer（超过45万学习记录）
    - 包含c&c++、Computer Science、Big Data Science三门课程，均具有全面的课程知识图谱
  - XJTU（超过12万学习记录）
    - 包含JAVA、NET、OS三门课程，均具有全面的课程知识图谱

Count	XJTU online learning dataset			MOOPer dataset		
	JAVA	NET	OS	C&C++	CS	BD
Learning resources	91	87	166	50	145	108
KG entities	363	338	496	114	306	263
Learners	586	708	751	5789	9450	10,667
Learning logs	19,651	54,248	55,298	99,364	201,830	157,673
Average length of learning paths	13.63	14.62	15.41	34.26	20.57	14.90
Maximum length of learning paths	132	122	199	175	114	87
Medium length of learning paths	6	7	7	30	15	10
Sparsity	85.11%	83.27%	89.69%	64.92%	84.91%	85.96%

实验设计 对比方法及评价指标

- 对比方法
  - LPG、ALPR、LPR、KGALPR
  - LSTMPR
- 评价指标
  - Precision, Recall和F1 score
  - 基于POMDP算法重新定义Precision, Recall来衡量推荐路径的效果
  - 路径Precision

$$Precision = \frac{|LCS(Path_{pred}, Path_{actual})|}{|Path_{pred}|}$$

  - 路径Recall

$$Recall = \frac{|LCS(Path_{pred}, Path_{actual})|}{|Path_{actual}|}$$

实验设计 对比实验

- 实验结果
  - cDQN-PathRec算法在三个指标上均达到了最佳性能
  - 所有指标的值都明显较低，这是由于学习路径评价指标的计算考虑了学习资源的顺序

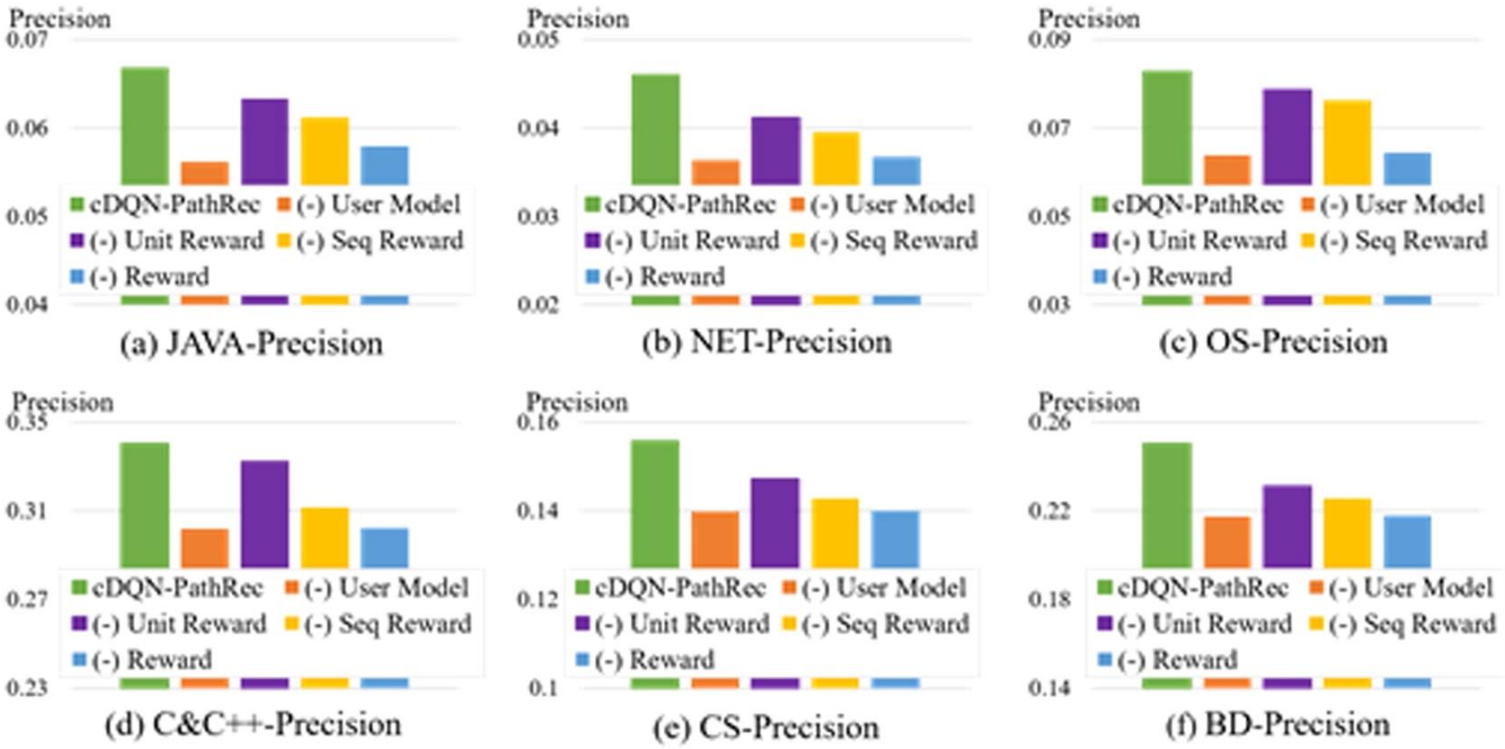
Algorithms	JAVA			NET			OS		
	Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Precision*	*Recall*	*F1*
LPG	0.0368	0.0072	0.0120	0.0338	0.0051	0.0089	0.0123	0.0048	0.0069
ALPR	0.0374	0.0022	0.0111	0.0401	0.0026	0.0098	0.0127	0.0044	0.0065
KGALPR	0.0452	0.0076	0.0130	0.0426	0.0063	0.0110	0.0163	0.0053	0.0080
LPR	0.0476	0.0057	0.0102	0.0454	0.0087	0.0146	0.0188	0.0052	0.0081
LSTMPR	0.0556	0.0081	0.0117	0.0462	0.0091	0.0123	0.0194	0.0038	0.0052
cDQN-PathRec	**0.0668**	**0.0128**	**0.0215**	**0.0828**	**0.0162**	**0.0271**	**0.0460**	**0.0095**	**0.0157**

TABLE V. EXPERIMENTAL RESULTS ON MOOPER DATASET

Algorithms	C&C++			CS			BD		
	Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Precision*	*Recall*	*F1*
LPG	0.1599	0.0239	0.0416	0.0947	0.0209	0.0342	0.1365	0.0218	0.0376
ALPR	0.1682	0.0235	0.0412	0.0955	0.0211	0.0346	0.1376	0.0224	0.0385
KGALPR	0.1846	0.0255	0.0448	0.1026	0.0233	0.0380	0.1437	0.0257	0.0436
LPR	0.2179	0.0261	0.0466	0.1198	0.0243	0.0404	0.1666	0.0278	0.0476
LSTMPR	0.2223	0.0279	0.0417	0.1294	0.0258	0.0350	0.1706	0.0294	0.0410
cDQN-PathRec	**0.3406**	**0.0540**	**0.0932**	**0.1558**	**0.0312**	**0.0520**	**0.2506**	**0.0449**	**0.0762**

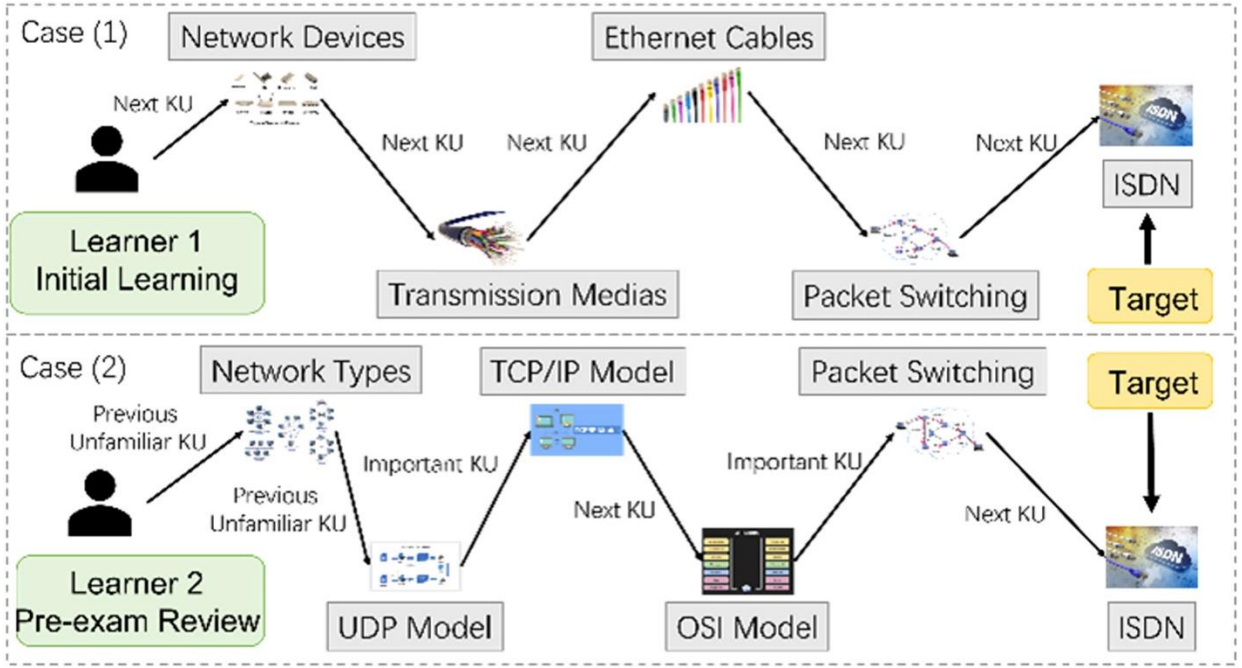
实验设计 消融实验

- 实验结果
  - 去除用户行为模型的效果下降最为显著，提出的多行为模型能够准确捕捉学习者学习过程中的时间序列关系和跨行为语义关联
  - 删除路径级奖励函数对模型的影响更大，推断总体规划奖励更能反映模型的收敛方向



实验设计 案例研究

- 案例任务
  - 针对相同的学习目标“ISDN”，分别对处于初始学习场景的学习者1和处于考前复习学习场景的学习者2进行推荐
- 实验结果
  - 对于学习者1，模型给出了如何直接向目标学习的推荐；对于学习者2，模型给出了包含更多相关学习资源的推荐
  - 基于学习者的学习场景不同状态进行适应性的推荐



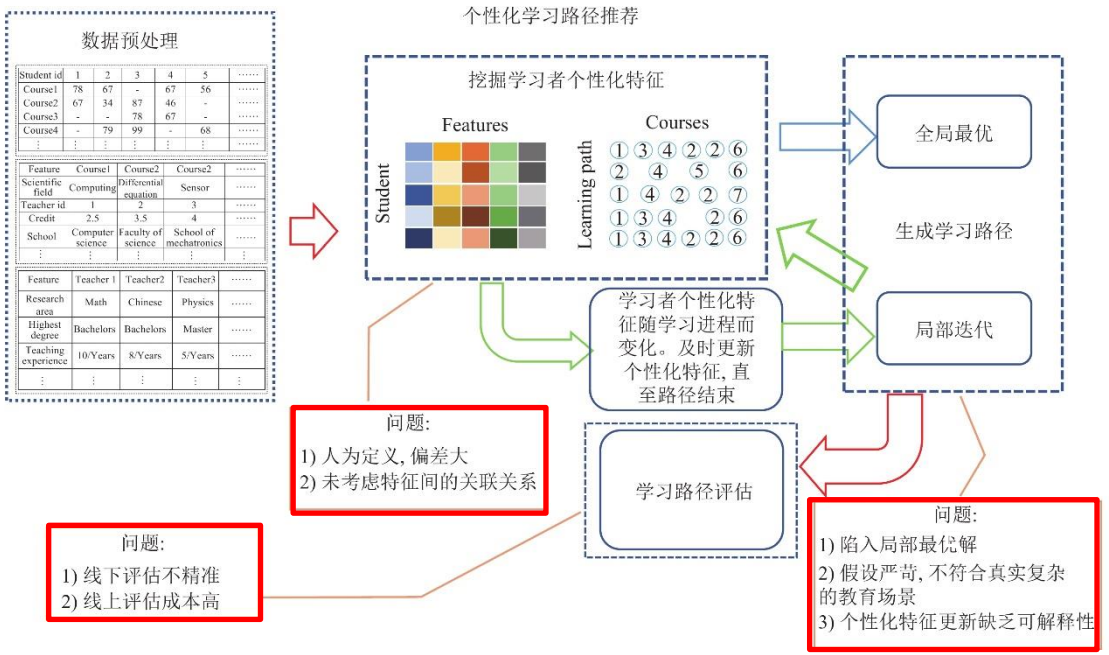
特点总结与未来展望

特点总结

- 算法优势
  - GEHRL
    - 利用分层强化学习将学习路径推荐分为子目标选择和子目标实现，可以规划多个目标的实现顺序，或者将一个困难的目标分解成多个目标逐一实现
    - 开发了基于图的候选选择器来约束低级代理的行为，确保实现目标的路径不包含不相关的学习项目，可以更有效地实现目标
  - cDQN-PathRec
    - 利用了基于知识图的多行为Transformer架构，全面地对用户状态建模，包括知识背景、学习风格、设置和偏好等因素，增强了模型的适应性
    - 结合具有两级奖励函数的级联深度Q网络确保模型收敛于整体和局部最优的平衡

未来展望

- 算法劣势
  - GEHRL
    - 未探索概念与练习之间的复杂关系
    - 奖励简单等价于问题回答正确概率
  - cDQN-PathRec
    - 未深入挖掘学习语境和用户特征
    - 奖励函数依赖于既定的学习项目序列，未考虑学习者的知识状态变化



- 未来展望
  - 深入挖掘特征间的关联关系，同时增强个性化推荐的可解释性
  - 最小的学习成本（步长、搜索空间）最大化推荐路径的效果
  - 从学习者和项目两个角度开发更合理、更全面、更高效的反馈评估模型

参考文献

[1] Li Q, Xia W, Yin L, et al. Graph Enhanced Hierarchical Reinforcement Learning for Goal-oriented Learning Path Recommendation[C]//Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023: 1318-1327.

[2] Ma D, Zhu H, Liao S, et al. Learning Path Recommendation with Multi-behavior User Modeling and Cascading Deep Q Networks[J]. Knowledge-Based Systems, 2024: 111743.

道德经

知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者，寿。

谢谢！