CIKM 2019
Beijing, China. Nov. 03-07, 2019
The 28th ACM International Conference on Information and Knowledge Management

Exploring Multi-Objective Exercise Recommendations in Online Education Systems

Zhenya Huang[1], Qi Liu[1], Chengxiang Zhai[2], Yu Yin[1], Enhong Chen[1], Weibo Gao[1], Guoping Hu[3]

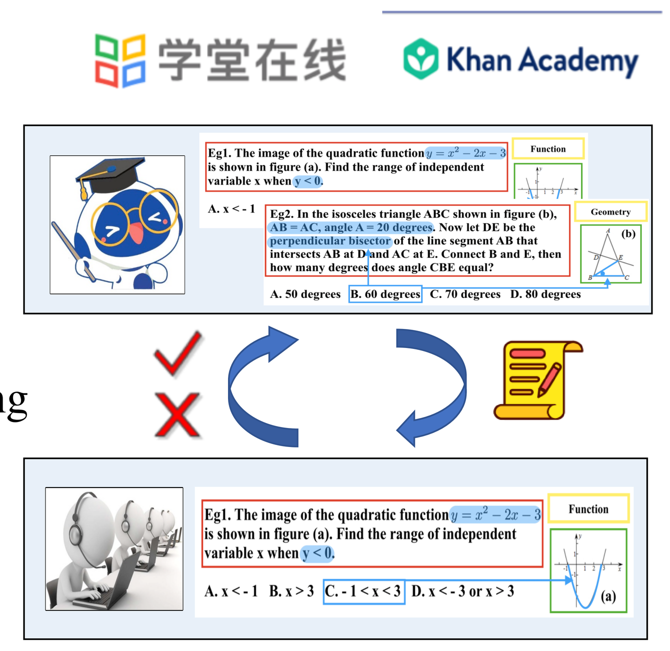
[1]Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China, [2]University of Illinois at Urbana-Champaign, [3]iFLYTEK Research

Introduction

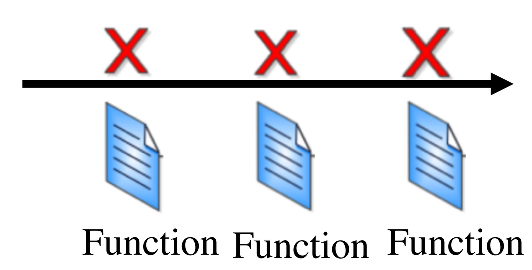
Online education systems become popular
- ➢Abundant learning materials
  - ➢E.g., exercise, course, video
- ➢Personalized learning on students' own paces

Recommender systems
- ➢Suggest suitable exercises instead of letting self-seeking
- ➢Interactive systems between agent vs. student

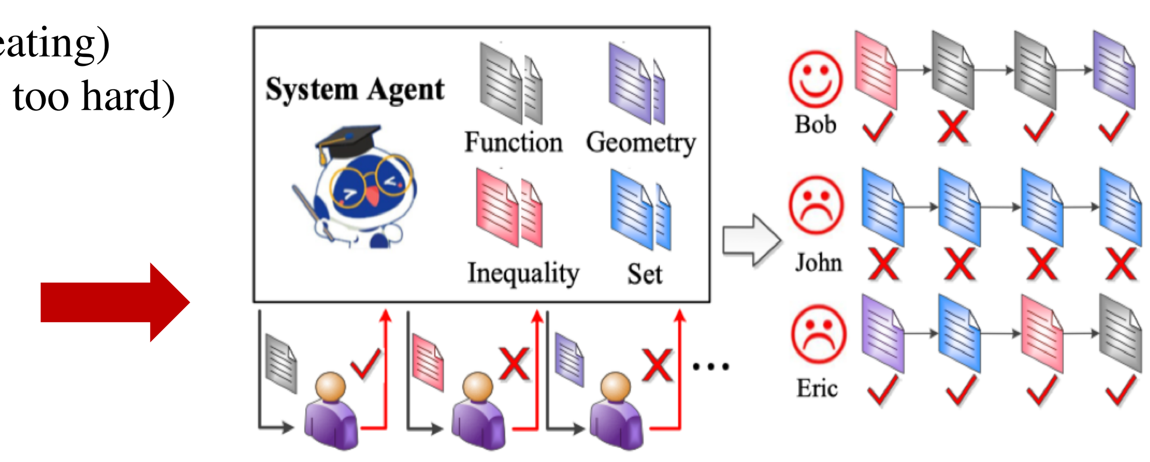
Key Problem
- ➢Design an optimal recommendation strategy that can recommend the best exercises at the right time



Existing recommendation for online learning
- ➢Basic idea
  - ➢Try to discover the weaknesses of students
  - ➢Recommending non-mastered exercises
- ➢Educational psychology
  - ➢Cognitive diagnosis, Q-learning
- ➢Data mining
  - ➢Content-based, Collaborative Filtering, Deep learning
- ➢Problem
  - ➢Single Objective (repeating)
  - ➢Lose interests (always too hard)



Multiple Objectives
- ➢Review & Explore
- ➢Difficulty Smoothness
- ➢Engagement

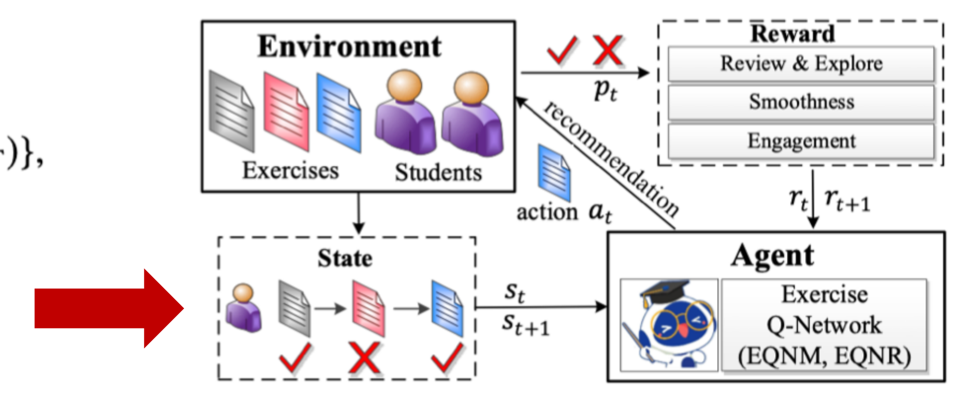


Challenges
- ➢How to define above objectives based on exercising trajectories
- ➢How to enable flexible recommendations with above objectives simultaneously?
- ➢Large space of exercise candidates

Problem Definition

Given
- ➢Student $u = \{(e_1, p_1), (e_2, p_2), \cdots, (e_T, p_T)\}$,
- ➢Exercise: triplet $e = \{c, k, d\}$
  - ➢Content c: word sequence
  - ➢Knowledge k: concept attribute
  - ➢Difficulty level d: error rate

Goal
- ➢Find the optimal exercises at each step for each student



Find an optimal policy $\pi : S \rightarrow A$ of recommending exercises to students, maximizing the multi-objective rewards

DRE Framework

Optimization Objective
- ➢Optimal action-value function

$$Q^*(s,a) = \mathbb{E}_{s'}[r + \gamma \max_{a'} Q^*(s',a')|s,a].$$

- ➢Compute all Q-values is infeasible
  - ➢Estimate and store all (s, a) pairs
  - ➢Update all Q-values
- ➢Solution
  - ➢EQN: as a non-linear function approximator θ

$$Q^*(s,a) \approx Q(s,a;\theta)$$

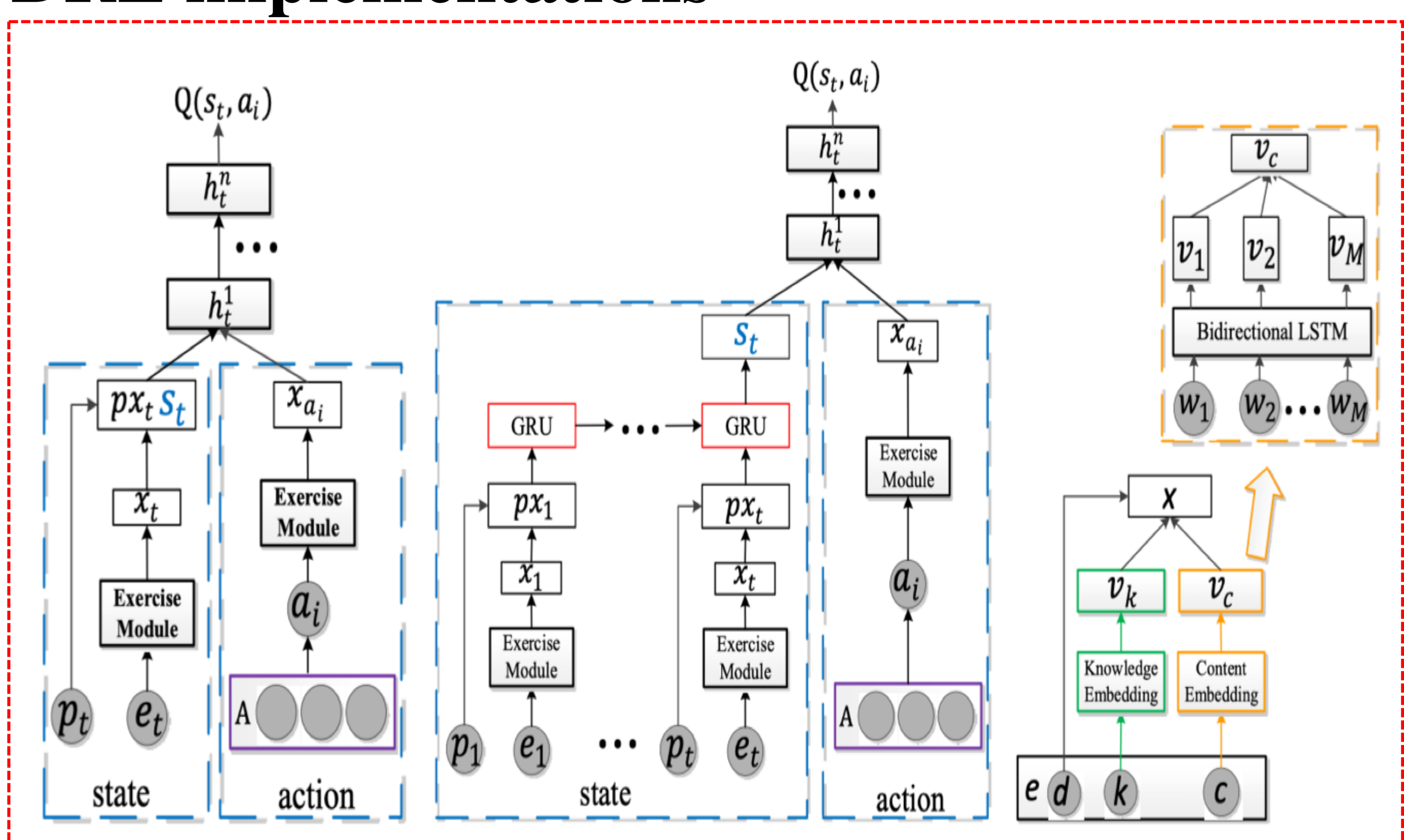
  - ➢Minimize the objective function to estimate this network approximator

$$L_t(\theta_t) = \mathbb{E}_{s,a,r,s'}[(y - Q(s,a;\theta_t))^2],$$
$$y = \mathbb{E}_{s'}[r + \gamma \max_{a'} Q(s',a';\theta_{t'})|s,a]$$

Algorithm 1: DRE Learning with Off-Policy Training

```
Initialize replay memory D with capacity Z;
Initialize action-value function Q with random weights.;
for u = 1, 2, ···, |U| do
    Randomly initialize state s_0;
    for t = 1, 2, ···, T do
        Observe state s_t = (e_t, p_t) in EQNM or
          s_t = {(e_1, p_1), ···, (e_t, p_t)} in EQNR;
        Execute action a_t (e_{t+1}) from off-policy π_o(s_t);
        Compute reward r_t according to p_{t+1} by Eq. (10);
        Set state s_{t+1} = (e_{t+1}, p_{t+1}) in EQNM or
          s_{t+1} = {(e_1, p_1), ···, (e_t, p_t), (e_{t+1}, p_{t+1})} in EQNR;
        Store transition (s_t, a_t, r_t, s_{t+1}) in D;
        Sample minibatch of transition (s, a, r, s') from D;
        y = { r                                terminal s'
            { r + γ max_a'(Q(s', a'); θ)       non-terminal s' ;
        Minimize (y − Q(s, a); θ)^2 by Eq. (3);
    end
end
```

DRE implementations



(a) EQNM with Markov property (b) EQNR with Recurrent manner (c) Exercise Module

Exercise Q-Network
- ➢Generate recommendation
  - ➢Implement network approximator θ
- ➢**Exercise Module**
  - ➢Goal: Learn exercise semantics
  - ➢Knowledge Embedding
  - ➢Content Embedding: Bi-LSTM
- ➢**Two implements**
  - ➢Goal: Learn student knowledge states
  - ➢Estimate Q value Q(s, a)
    - ➢EONM with Markov property $s_t = (e_t, p_t)$
    - ➢EQNR with Recurrent manner $s_t = \{(e_1, p_1), \cdots, (e_t, p_t)\}$

Multi-Objective Rewards
- ➢**Review & Explore**

$$r_1 = \begin{cases} \beta_1 & \text{if} \quad p_t = 0 \quad \text{and} \quad k_{t+1} \cap k_t = \emptyset, \\ \beta_2 & \text{if} \quad k_{t+1} \setminus \{k_1 \cup k_2 \cup \cdots \cup k_t\} \neq \emptyset, \\ 0 & \text{else.} \end{cases}$$

- ➢**Difficulty Smoothness**

$$r_2 = \mathcal{L}(d_{t+1}, d_t) = -(d_{t+1} - d_t)^2,$$

- ➢**Engagement**

$$r_3 = 1 - |g - \varphi(u, N)|, \quad \varphi(u, N) = \frac{1}{N} \sum_{i=t-N}^{t} p_i,$$

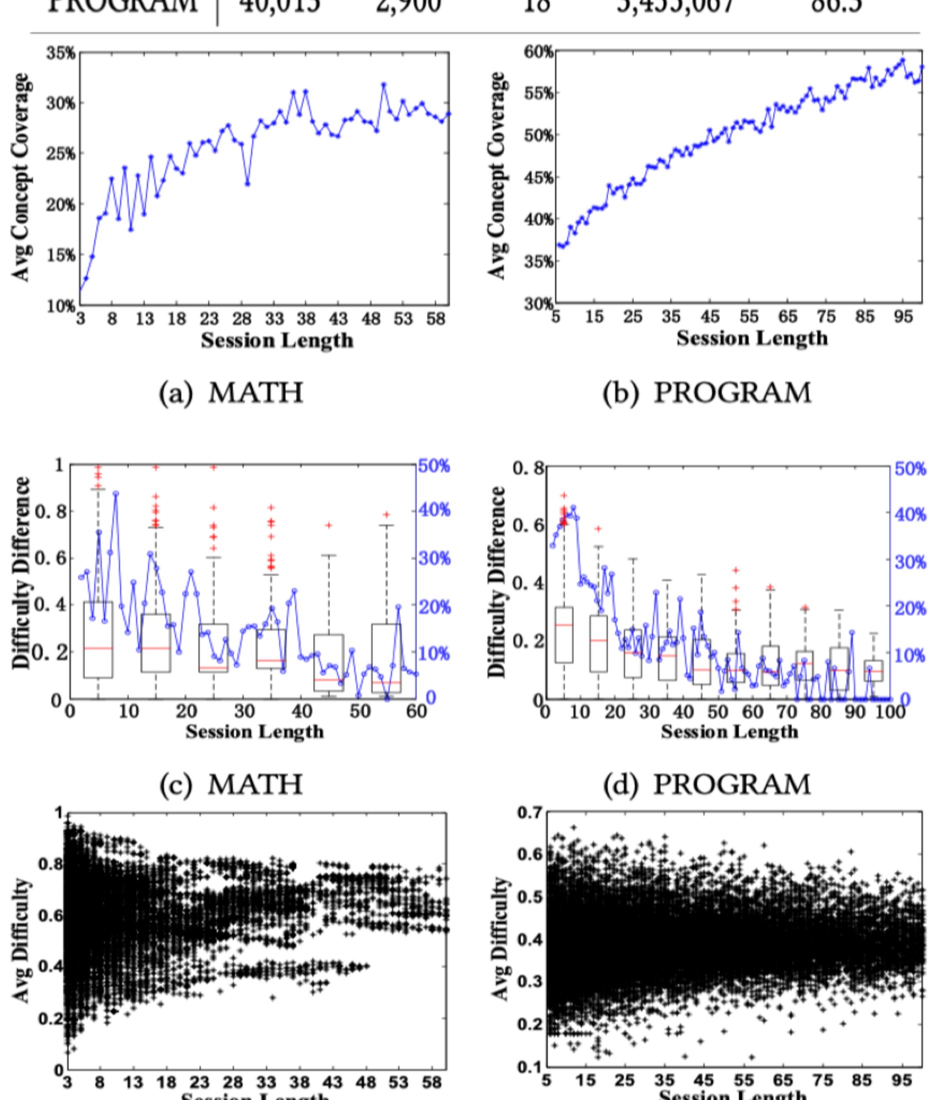
- ➢**Balancing**

$$r = \alpha_1 \times r_1 + \alpha_2 \times r_2 + \alpha_3 \times r_3, \quad \{\alpha_1, \alpha_2, \alpha_3\} \in [0, 1].$$

Datasets

Table 1: The statistics of the datasets.

Dataset	Num. Students	Num. Exercises	Num. Concepts	Num. records	Avg. records per student
MATH	52,010	2,464	37	1,272,264	24.5
PROGRAM	40,013	2,900	18	3,455,067	86.3



(a) MATH (b) PROGRAM (c) MATH (d) PROGRAM (e) MATH (f) PROGRAM

Offline Evaluation

Point-wise Recommendation

- ➢Evaluation on logged data
- ➢Ranking problem
  - ➢Provide an list at a particular time based on Q-values (related to performance) from bad to good (70%/30%)

(a) **MATH**

Methods	NDCG@10	NDCG@15	MAP@10	MAP@15	F1@10	F1@15
IRT	0.5065	0.6235	0.3373	0.4463	0.2100	0.3464
PMF	0.4900	0.5986	0.3155	0.4163	0.2016	0.3347
FM	0.5123	0.6279	0.3419	0.4507	0.2123	0.3489
DKT	0.5587	0.7033	0.3959	0.5486	0.2797	0.4634
DKVMN	0.5657	0.7112	0.4021	0.5581	0.2895	0.4747
DQN	0.5031	0.7001	0.3191	0.5296	0.2912	0.5178
DREM	**0.6114**	0.7773	**0.4355**	0.6353	0.3559	0.6033
DRER	0.6129	**0.7813**	0.4337	**0.6435**	**0.3676**	**0.6099**

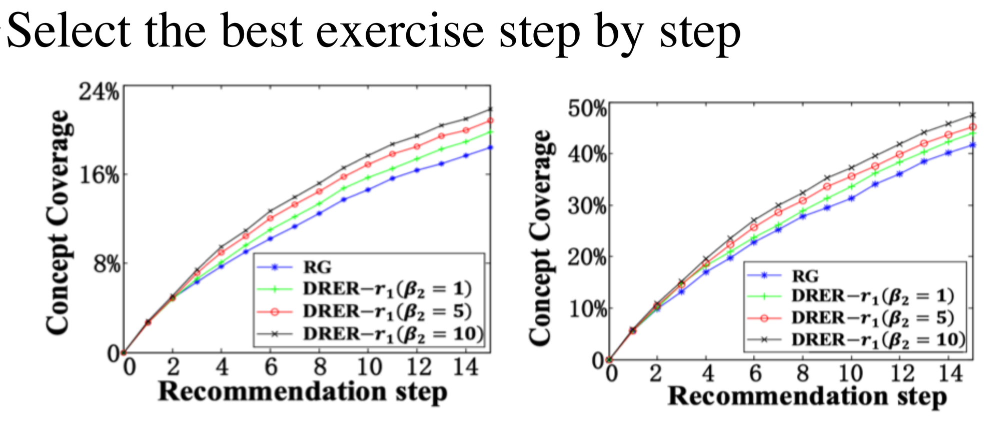
(b) **PROGRAM**

Methods	NDCG@10	NDCG@15	MAP@10	MAP@15	F1@10	F1@15
IRT	0.3369	0.4231	0.1852	0.2430	0.0879	0.1530
PMF	0.3330	0.4152	0.1810	0.2336	0.0842	0.1467
FM	0.3664	0.4456	0.2081	0.2617	0.0921	0.1567
DKT	0.3893	0.4924	0.2361	0.3197	0.1451	0.2445
DKVMN	0.3853	0.4889	0.2351	0.3226	0.1555	0.2620
DQN	0.3422	0.4901	0.1851	0.3095	0.1781	0.3266
DREM	0.4446	0.5638	0.2753	0.3834	0.1683	0.3325
DRER	**0.4538**	**0.5907**	**0.2802**	**0.4059**	**0.2091**	**0.3655**

Online Evaluation

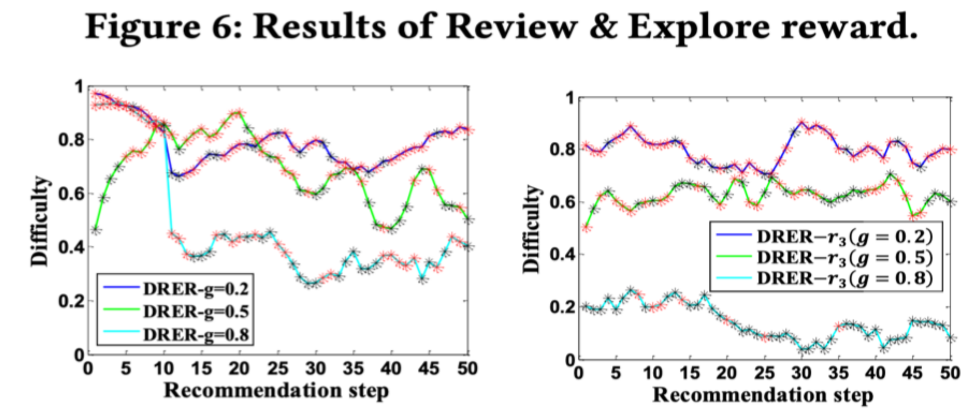
Sequence-wise Recommendation

- ➢Evaluation on simulated environment
- ➢Reward effectiveness
  - ➢Select the best exercise step by step



(a) MATH (b) PROGRAM

Figure 6: Results of Review & Explore reward.



(a) MATH (b) PROGRAM

Figure 7: Results of Smoothness vs. Engagement rewards.

- ✓ DRER with larger β_2 has faster coverage growth speed
- ✓The difficulty levels of recommendations do not vary dramatically in most cases
- ✓If we set g with lower value (0.2), DRER would recommend more difficult exercises