Hierarchical Reinforcement Learning for Course Recommendation in MOOCs

Jing Zhang, Bowen Hao, Bo Chen, Cuiping Li, Hong Chen
(Renmin University)

Jimeng Sun (Georgia Institute of Technology)

Course Recommendation

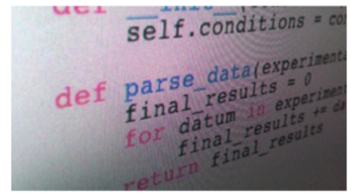
学堂在线
xuetangx.com

人人都能学计算机：计算机科学入门与Python编程

来自于：哈维穆德学院 | 分类：计算机(502)

相关课程

Related courses



计算机科学和Python编程导论
（自主模式）

Python程序设计

MyCS：计算机科学入门

edX

coursera

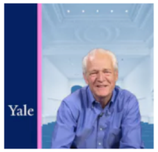
UDACITY

Problem Definition

Input: Historical enrolled courses of a user before *t*

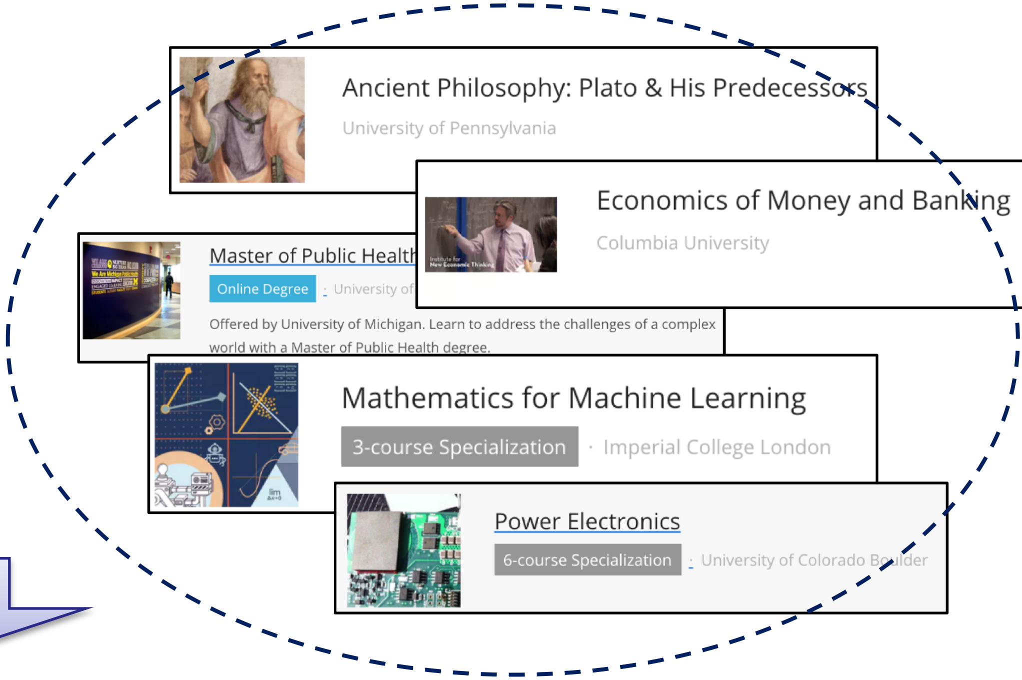
Machine Learning
Course · Stanford University

Information Systems
Specialization · University of Minnesota



Introduction to Classical Music
Course · Yale University

Candidate Courses:



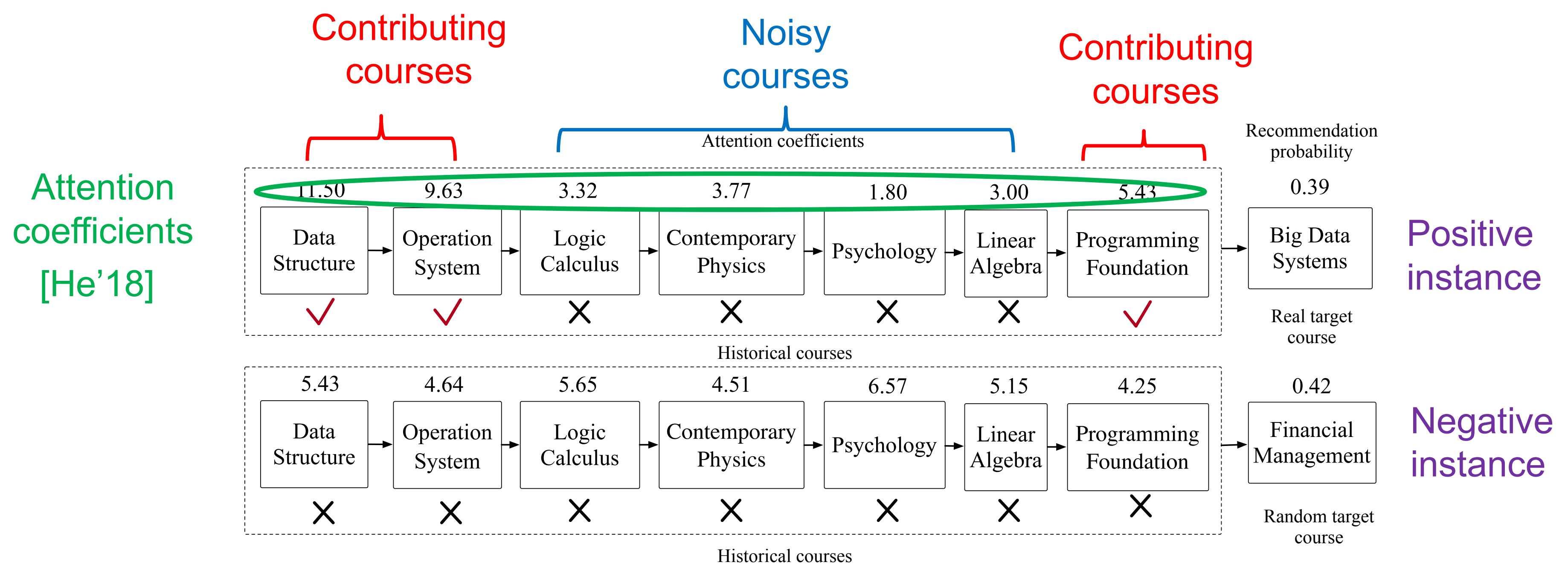
Output:

Most possible courses to be enrolled at *t*+1

Mathematics for Machine Learning
3-course Specialization · Imperial College London

Ancient Philosophy: Plato & His Predecessors
University of Pennsylvania

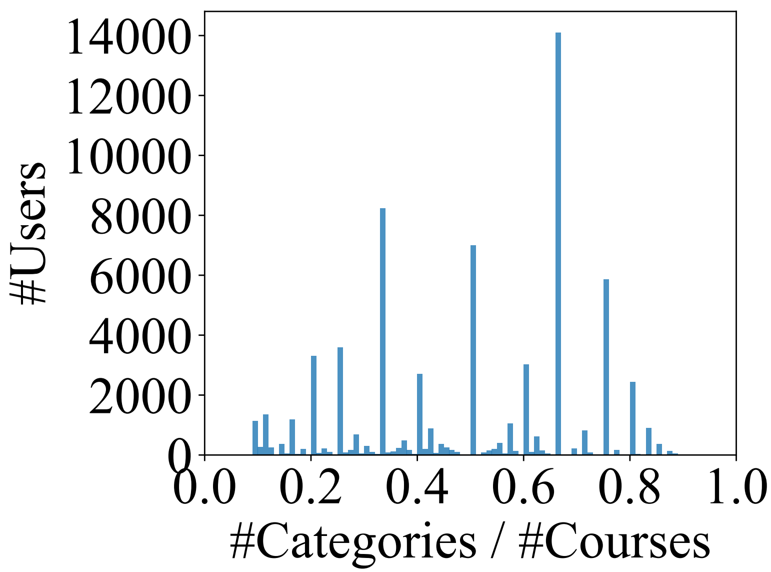
Challenges



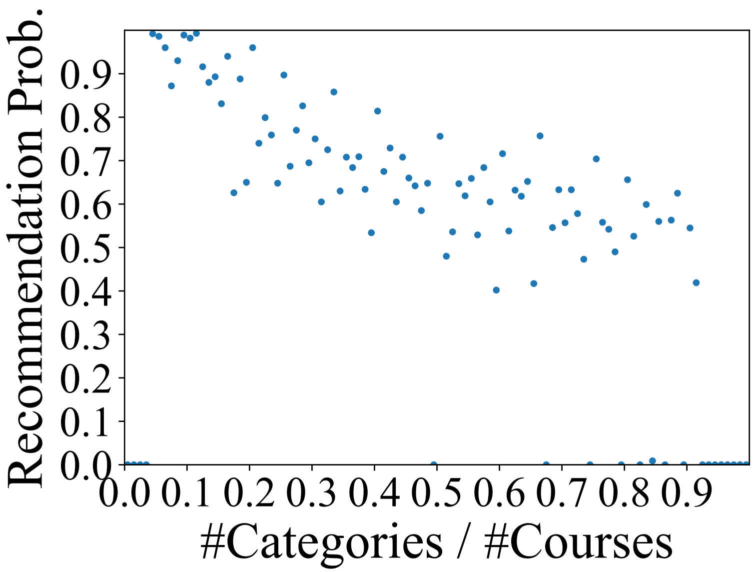
- Users' enrolled courses are usually diverse. The contributing courses may be diluted by noisy courses
- Even if no courses can contribute in predicting a random target course, each historical course will still be assigned an attention coefficient.

Data Analysis

Users	Courses	Categories	User-course pairs	Time
82,535	1,302	23	458,454	2016.10.1-2018.3.31



A large number of users enrolled diverse courses



The recommendation performance based on the diverse profiles is impacted

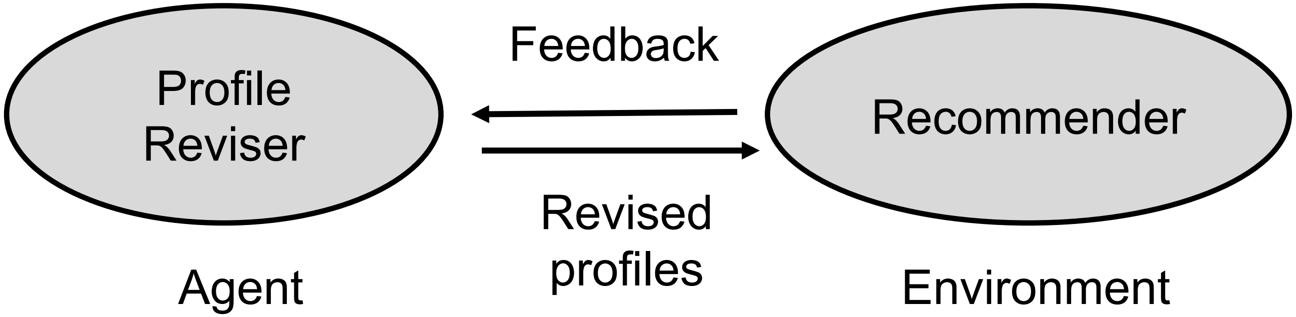
(A bigger #categories/#courses indicates the user is more distractive)

Idea to Deal with the Challenges

- Revise the user profiles by removing the noisy courses instead of assigning an attention coefficient to each of them.
  - But how to determine which courses should be removed?
  - Without the supervised information, can we automatically learn the pattern?

Solution

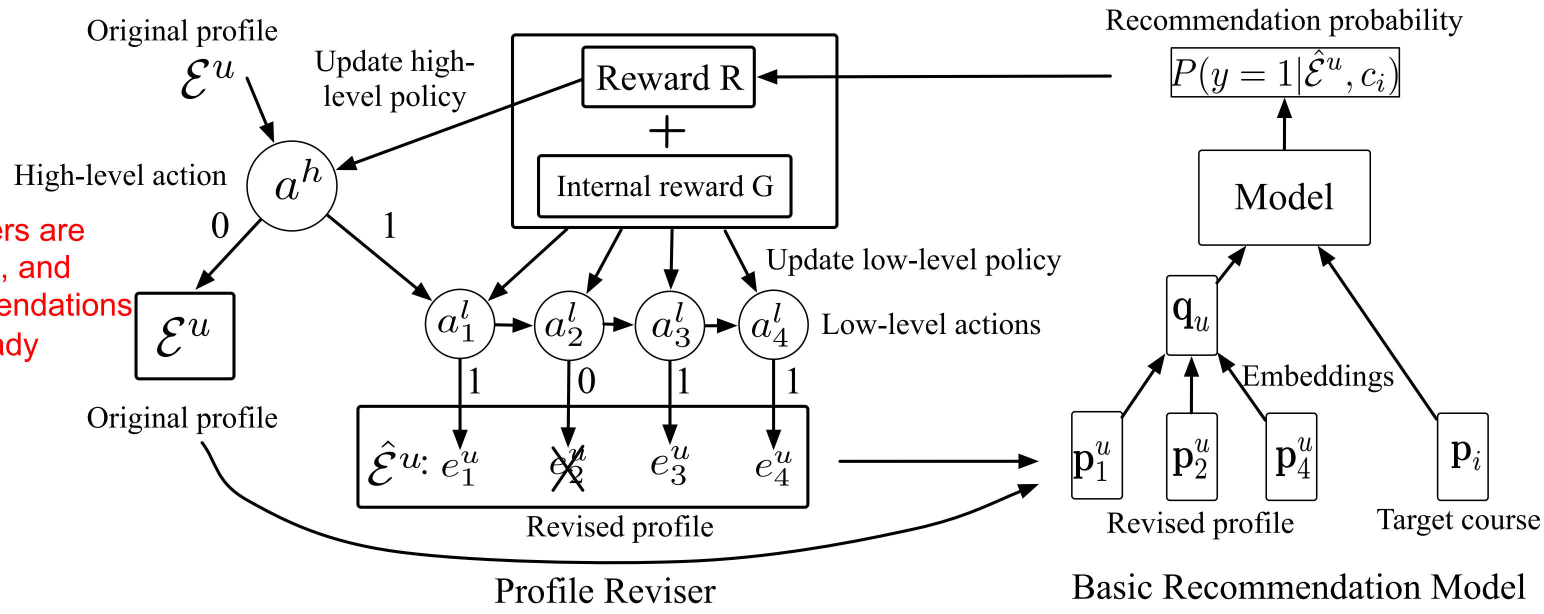
Revise the user profiles based on the feedbacks from the recommender



The revising process of a user profile: a sequential decision process

- The profile reviser (agent)
  - Revise the profile
  - Gets a delayed reward from the recommender
  - Update its policy
- The recommender (environment)
  - Update its parameters based on the profiles revised by the profile reviser

Framework



30% users are attentive, and recommendations are already correct.

- A hierarchical Markov Decision Process
  - The agent firstly performs a high-level task to determine whether to revise the whole profile or not
  - If it decides to revise, the agent performs a low-level task of multiples actions to determine whether to remove each historical course or not
  - The overall task is finished after the low-level task is finished or the high-level task decides to make no revision.

The High-level Task

- Determine whether to revise the whole profile ε^u or not
  - State:
    - The average cosine similarity between the embedding vectors of each historical course in ε^u and the target course c_i.
    - The average element wise product between the embedding vectors of each historical course in ε^u and the target course c_i.
    - The probability $P(y = 1|\varepsilon^u, c_i)$ of recommending c_i to user u by the basic recommender.

    (The lower recommendation probability is, more effort should be taken to revise the profile)
  - Action:
    - {Revise, keep}
  - Delayed reward:

Policy function: two-layer NN

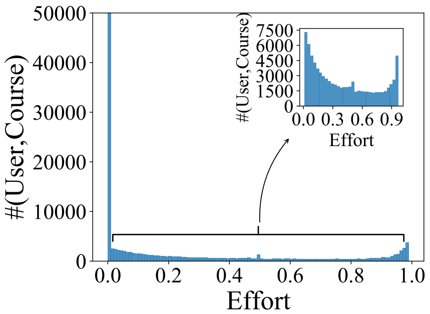
$$\begin{aligned} \mathbf{H}_t^l &= \mathrm{ReLU}(\mathbf{W}_1^l \mathbf{s}_t^l + \mathbf{b}^l), \qquad (3) \\ \pi(\mathbf{s}_t^l, a_t^l) &= P(a_t^l | \mathbf{s}_t^l, \Theta^l) \\ &= a_t^l \sigma(\mathbf{W}_2^l \mathbf{H}_t^l) + (1 - a_t^l)(1 - \sigma(\mathbf{W}_2^l \mathbf{H}_t^l)), \end{aligned}$$

The difference between the log-likelihood after and before the profile is revised.

$$R(a_t^l, \mathbf{s}_t^l) = \begin{cases} \log p(\hat{\mathcal{E}}^u, c_i) - \log p(\mathcal{E}^u, c_i). & \text{if } t = t_u; \\ 0 & \text{otherwise,} \end{cases}$$

The low-level Task

- Determine whether to remove a historical course $e_t^u \in \varepsilon^u$ or not
  - State:
    - The cosine similarity between the embedding vectors of the current historical course e_t^u and the target course c_i.
    - The element wise product between the embedding vectors of the current historical course e_t^u and the target course c_i.
    - Effort taken in the course.
  - Action:
    - {Remove, Keep}
  - Reward:
    - Add an internal reward
      - speed up local learning and does not propagate to the high-level.
    - $G(a_t^l, \boldsymbol{s}_t^l)$: calculate the average cosine similarity between each historical course and the target course after and before the profile is revised.



$$R(a_t^m, \mathbf{s}_t^m) + G(a_t^m, \mathbf{s}_t^m)$$

Effort: we calculate the ratio between the watch duration and the total duration of a video as the watch ratio, and use the maximal watch ratio of all the videos in a course to represent the effort taken by the user in the course

Objective Function

- Maximize the expected reward:

$$\Theta^* = \text{argmax}_\Theta \sum_{\tau} \boxed{P_\Theta(\tau;\Theta)} \boxed{R(\tau)},$$

A sequence of the sampled actions and the transited states

$\{s_1^l, a_1^l, s_2^l, \cdots, s_t^l, a_t^l, s_{t+1}^l, \cdots\}$

Sampling probability

The reward for the sampled sequence

- Update the policy network with policy gradient:

$$\nabla_\Theta = \frac{1}{m} \sum_{m=1}^{M} \nabla_\Theta \log \pi_\Theta(\mathbf{s}^m, a^m) R(a_t^m, \mathbf{s}_t^m)$$

Training Procedure

```
Pre-train the basic recommendation model;
Pre-train the profiler reviser by running Algorithm 2
  with the basic recommendation model fixed;
Jointly train the two models together by running
  Algorithm 2;
```

Algorithm 1: The Overall Training Process

Pre-train +joint train

Input: Training data $\{\mathcal{E}^1, \mathcal{E}^2, \cdots, \mathcal{E}^{|U|}\}$, a pre-trained basic recommendation model and a profile reviser parameterized by Φ^0 and Θ^0 respectively

```
Initialize Θ = Θ^0, Φ = Φ^0 ;
for episode l=1 to L do
    foreach E^u := (e^u_1, ··· , e^u_{t_u}) and c_i do
        Sample a high-level action a^h with Θ^h;
        if a^h = 0 then
            R(s^h, a^h) = 0
        else
            Sample a sequence of low-level actions
              {a^l_1, a^l_2, ··· , a^l_{t_u}} with Θ^l;
            Compute R(a^l_{t_u}, s^l_{t_u}) and G(a^l_{t_u}, s^l_{t_u}) ;
            Compute gradients by Eq. (5) and (6);
        end
    end
    Update Θ by the gradients;
    Update Φ in the basic recommendation model;
end
```

Algorithm 2: The Hierarchical Reinforcement Learning

Sample actions and transited states and get rewards

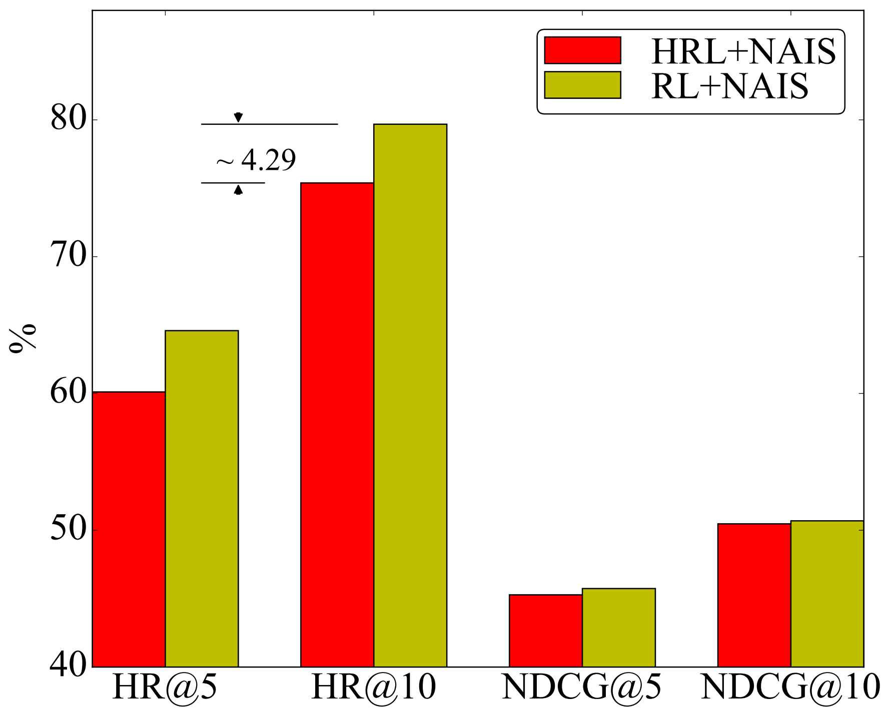
Update profile reviser and recommender

Experiment Results

Table 1: Recommendation performance (%).

	Methods	HR@5	HR@10	NDCG@5	NDCG@10
User-item model	BPR	46.82	60.73	34.16	38.65
	MLP	52.16	66.29	40.39	44.41
	FM	46.01	61.07	35.28	40.15
Item-item model, average	FISM	52.73	65.64	40.00	44.98
	GRU	52.07	68.63	38.92	46.30
Attention model	NAIS	56.42	69.05	43.73	47.82
	NASR	54.64	69.48	42.39	47.33
RL model	HRL+NAIS	**64.59**	**79.68**	**45.74**	**50.69**
	HRL+NASR	**59.05**	**74.50**	**47.51**	**52.73**

Compared with One-level RL



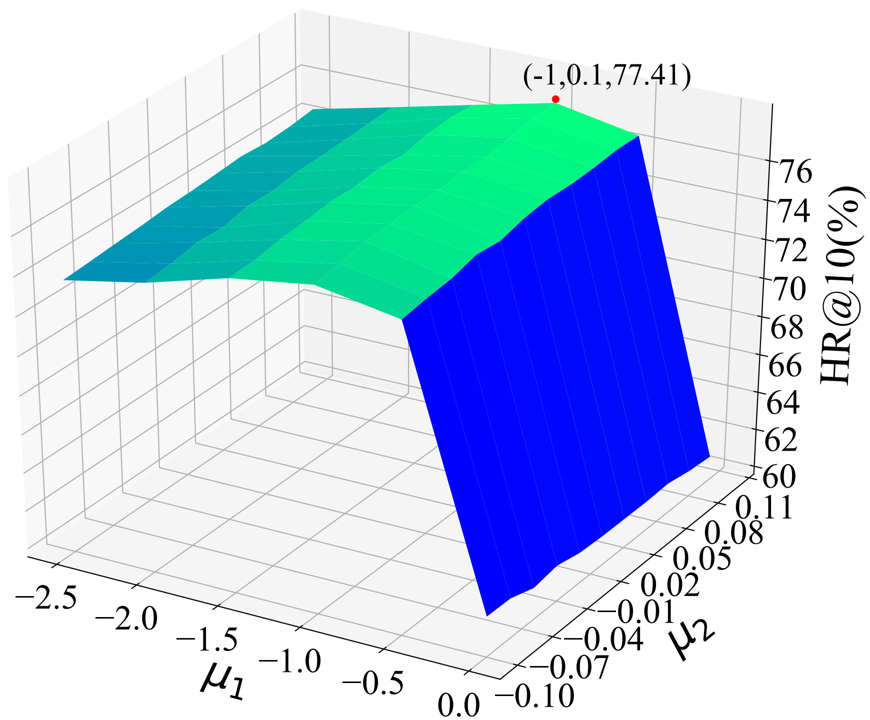
- The average #Categories/#Courses of the revised profiles:
  - Two-level RL: 0.73
  - One-level RL: 0.75
- The revised profiles by the two-level HRL are more consistent

The high-level task of the two-level RL:

- The average #Categories/#Courses of
  - the kept profiles: 0.57
  - the revised profiles: 0.69
- High-level task tends to keep more consistent profiles

(A bigger #categories/#courses indicate the user is more distractive)

Compared with Greedy Revision



The greedy reviser

- firstly decides to revise the whole profile if $P(y = 1|\varepsilon^u, c_i)$ < μ_1
- and then removes the course $e_t^u \in \varepsilon^u$ if its cosine similarity with c_i is less than μ_2

Compared with Attentions

Table 2: Case studies of the profiles revised by HRL+NAIS and the attention coefficients learned by NAIS.

Methods	Revised profile or the learned attentions	The target course
HRL+NAIS	~~Crisis Negotiation~~, ~~Social Civilization~~, Web Technology, C++ Program	Web Development
NAIS	Crisis Negotiation(29.61), Social Civilization(29.09), Web Technology(28.32), C++ Program(28.12)	Web Development
HRL+NAIS	Modern Biology, Medical Mystery, Biomedical Imaging, ~~R Program~~	Biology
NAIS	Modern Biology(37.79), Medical Mystery(37.96), Biomedical Imaging(37.62), R Program(37.84)	Biology
HRL+NAIS	~~Web Technology~~, Art Classics, ~~National Unity Theory~~, Philosophy	Life Aesthetics
NAIS	Web Technology(38.32), Art Classics(35.87), National Unity Theory(40.63), Philosophy(43.69)	Life Aesthetics

Conclusion

- We present the first attempt to solve the problem of course recommendation in MOOCs platform by a hierarchical RL model.
- The model jointly trains a profile reviser and a basic recommendation model, which enables the hierarchical RL model effectively to remove the noisy courses to the target course, and enables recommendation model to be improved on revised user profiles by an agent.
- The experimental results on a real dataset collected from XuetangX validate the effectiveness of the proposed model.

Thank you！