Class Planning using Reinforcement Learning

Julia Reisler
jreisler@stanford.edu

Rishu Garg
rishu@stanford.edu

Pete Rushton
prushton@stanford.edu

Shubo Yang
shuboy@stanford.edu

Wanyue Zhai
wzhai702@stanford.edu

1 Problem

At Stanford, students often plan their schedules manually, making decisions based on program requirements, prerequisites, number of units, and course offerings, among other factors. Selecting the optimal ordering of classes can be daunting as it is difficult to know ahead of time which combination of classes will be most useful and enjoyable. For example, several classes may satisfy the same requirement for a student, but students may learn more from certain ones if they are more interested in the topic or the teaching quality is better. As another example, students may prefer to finish the degree faster than the two year expectation, but overloading on units to accomplish this could negatively impact mental health during the terms.

In this project, we will model course scheduling as a constrained search problem. We also plan to incorporate reinforcement learning and student preferences, which will be unknown to the algorithm ahead of time. Finally, the course plan will be dynamic – it can be updated based on choices students made in previous quarters.

2 Behavior

The input to the system will be the constraints of the search problem and a reward (or cost) function. We will use UCS to output the optimal ordering of classes based on the constraints.

We will start this problem using a subset of courses from explorecourses and create fake rewards for each one. Later on in the project, we will scrape data from Carta and Axess to incorporate sentiment analysis and course ratings for the reward function. While we have an api which scrapes data from explorecourses, we will need to follow up about how to access data on Carta and Axess. To constrain the problem, we will only consider "extra curricular" courses and technical courses related to CS, EE, and ICME programs.

3 Evaluation

There are several parts of the system that we can evaluate. To evaluate the end-to-end component, we will survey a sample of Stanford students for their reported satisfaction with regard to the course recommendations generated by the baseline and by our own models, with the recommendation source being concealed from the students.

We can also evaluate sub-parts of the system. To evaluate our Q-learning strategy, we can compare the total reward from the learned policy to the optimal reward that we compute with Value Iteration. To evaluate our sentiment analysis component, we can analyze how the predicted scores for reviews of a class compare to the overall class rating.

4 Related Work

There are multiple ways that we can approach the scheduling problem. One possible approach is to use search algorithms. Earlier work shows that Tabu search can be used in resource constrained project scheduling problem (RCPSP) and educational project schedulings [1] [2]. Wu and Havens [3] applied two-phased mixed-initiative constraint reasoning algorithms on scheduling. Such algorithms use

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

dynamic backtracking to first construct a initial plan and then semi-systematic local search to support user interaction. These approaches allows flexibility in individual user preferences. More recently, Srisamutr et al. [4] used a genetic algorithm for course scheduling and planning for undergraduate students. They included course conditions such as the offering semester as well as personal conditions (historical GPA accumulation).

There are various problems that use reinforcement learning on scheduling, including dynamic task scheduling [5], business process management [6], and adaptive scheduling of educational activities [7]. However, we found no research that specifically tackles the problem of course scheduling using reinforcement learning. This may be due to the large amount of historical data needed [8] and the inability of reinforcement learning to learn long sequences of data [9]. The method from Bassen et al. [7] leveraged reinforcement learning by using reinforcement scheduling to maximize learning gains and minimize the number of items assigned.

5 Baselines and Oracles

Baselines: The modeling baseline involves applying unconstrained search algorithms on a simple, deterministic model. Here, UCS will be used to find the minimum cost (highest reward) path. Once we have introduced uncertainties to the problem, for example, preferences and extracurricular courses, we will take a reinforcement learning approach. The baselines in this case will be Q-learning with ϵ set to 0 and 1 for a case of exploitation-only and exploring-only, respectively.

Oracles: Since the problem is very subjective and involves uncertainties, there is not an evaluation metric such as accuracy. However, because of the subjectivity, handcrafting the course plans by different group members may serve as "oracles". The manual and individual plan is nearly optimal for that person, under requirements and preferences. Thus, comparing the handcrafted plans and considerations will give us a view of expected results.

6 Methodology

Model: We are considering a search space which forms a Directed Acyclic Graph. A student goes from one state to another state between quarters by taking a set of available courses.

State: (Quarter, Number of Quarters elapsed, Total Credits left to Graduate, Number of Electives left, Number of Core classes left, Classes Taken, etc.)
Start State: A student enrolls into Stanford (No classes chosen yet)
End State: A student graduates from Stanford
Actions: Taking courses at a particular quarter and going to the next State (next quarter)
Reward: Initially, we will define reward as the rating of the class. Later, we will incorporate signals of "satisfaction" and "value" that a student gets from the course, including sentiment analysis from course reviews.

Method: We will implement a standard UCS algorithm that will find the least cost path from our Start State to End State. The cost here would just be $-Reward$. Some of the constraints include course requirements of the program, whether a course is offered during a particular quarter, minimum 8 and maximum 10 credits are taken in a quarter, and graduation in no more than 6 academic quarters. Later on, we will modify our model to include waypoint tags (like taking at least 4 AI courses), multiple end states (graduating before 6 quarters) and notions of uncertainty and individuality.

7 Challenge Description

Distilling this inherently subjective decision-making process - which even diligent, well-informed Stanford students find very complex - into a form that can be presented as an MDP is a challenge. For example, we seek a single reward statistic that aggregates optimal value across several different considerations (e.g. quality of teaching, utility of early graduation). Creating robust evaluation criteria also poses a challenge, because similar students could reasonably assess any particular course recommendation quite differently. We believe that challenges such as this may be mitigated by at least two approaches. First, by framing our MDP as a recommender system, which aims to generate a number of feasible study plans, the system can be a useful tool without having to solve the problem of what the single optimal recommendation is. Second, this project is an opportunity to generate recommendations that, in further study, may be an input to create better designed criteria for success, and thereby be used to refine future versions of our model.

References

[1] Koji Nonobe and Toshihide Ibaraki. Formulation and tabu search algorithm for the resource constrained project scheduling problem. In *Essays and surveys in metaheuristics*, pages 557–588. Springer, 2002.

[2] Filip Deblaere, Erik Demeulemeester, and Willy Herroelen. Rescon: Educational project scheduling software. *Computer Applications in Engineering Education*, 19(2):327–336, 2011.

[3] Kun Wu and William S Havens. Modelling an academic curriculum plan as a mixed-initiative constraint satisfaction problem. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 79–90. Springer, 2005.

[4] Alangkarn Srisamutr, Thitiporn Raruaysong, and Vacharapat Mettanant. A course planning application for undergraduate students using genetic algorithm. In *2018 Seventh ICT International Student Project Conference (ICT-ISPC)*, pages 1–5. IEEE, 2018.

[5] M Emin Aydin and Ercan Öztemel. Dynamic job-shop scheduling using reinforcement learning agents. *Robotics and Autonomous Systems*, 33(2-3):169–178, 2000.

[6] Zhengxing Huang, Wil MP van der Aalst, Xudong Lu, and Huilong Duan. Reinforcement learning based resource allocation in business process management. *Data & Knowledge Engineering*, 70(1):127–145, 2011.

[7] Jonathan Bassen, Bharathan Balaji, Michael Schaarschmidt, Candace Thille, Jay Painter, Dawn Zimmaro, Alex Games, Ethan Fast, and John C Mitchell. Reinforcement learning for the adaptive scheduling of educational activities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[8] Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. Offline policy evaluation across representations with applications to educational games. In *AAMAS*, volume 1077, 2014.

[9] Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*, 2011.