Simple Enroll: An MDP model for Course Scheduling

Robert Heckerman, Ozioma-Jesus Anyanwu
{robheck, ozijesus}@stanford.edu

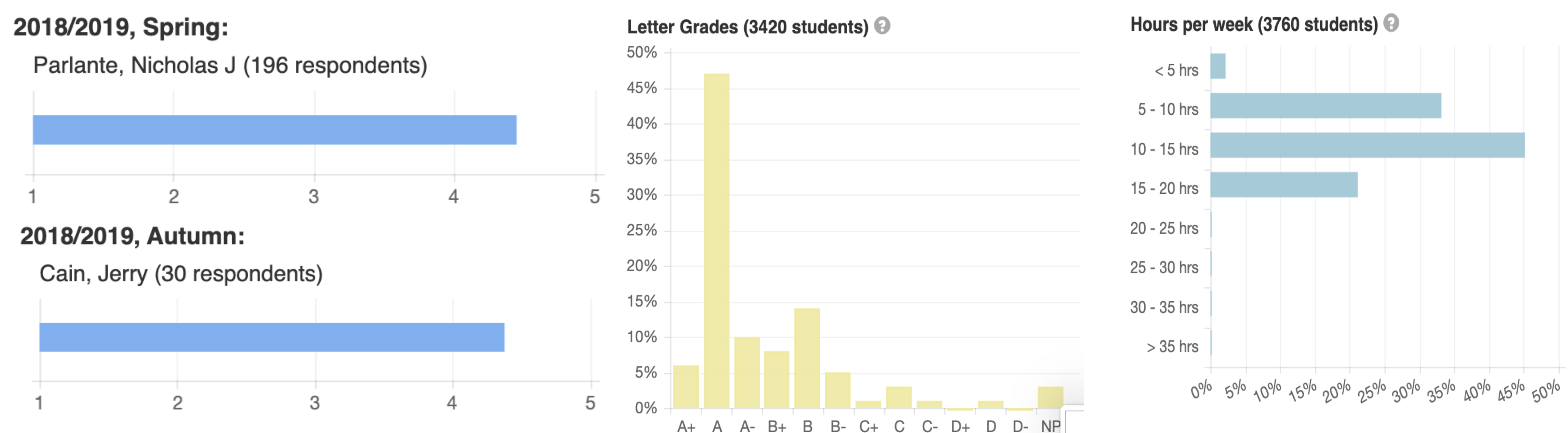
1. Overview

Problem:
- Picking a good course schedule at Stanford is a stressful process. Even experienced students often pick a regrettable quarter.

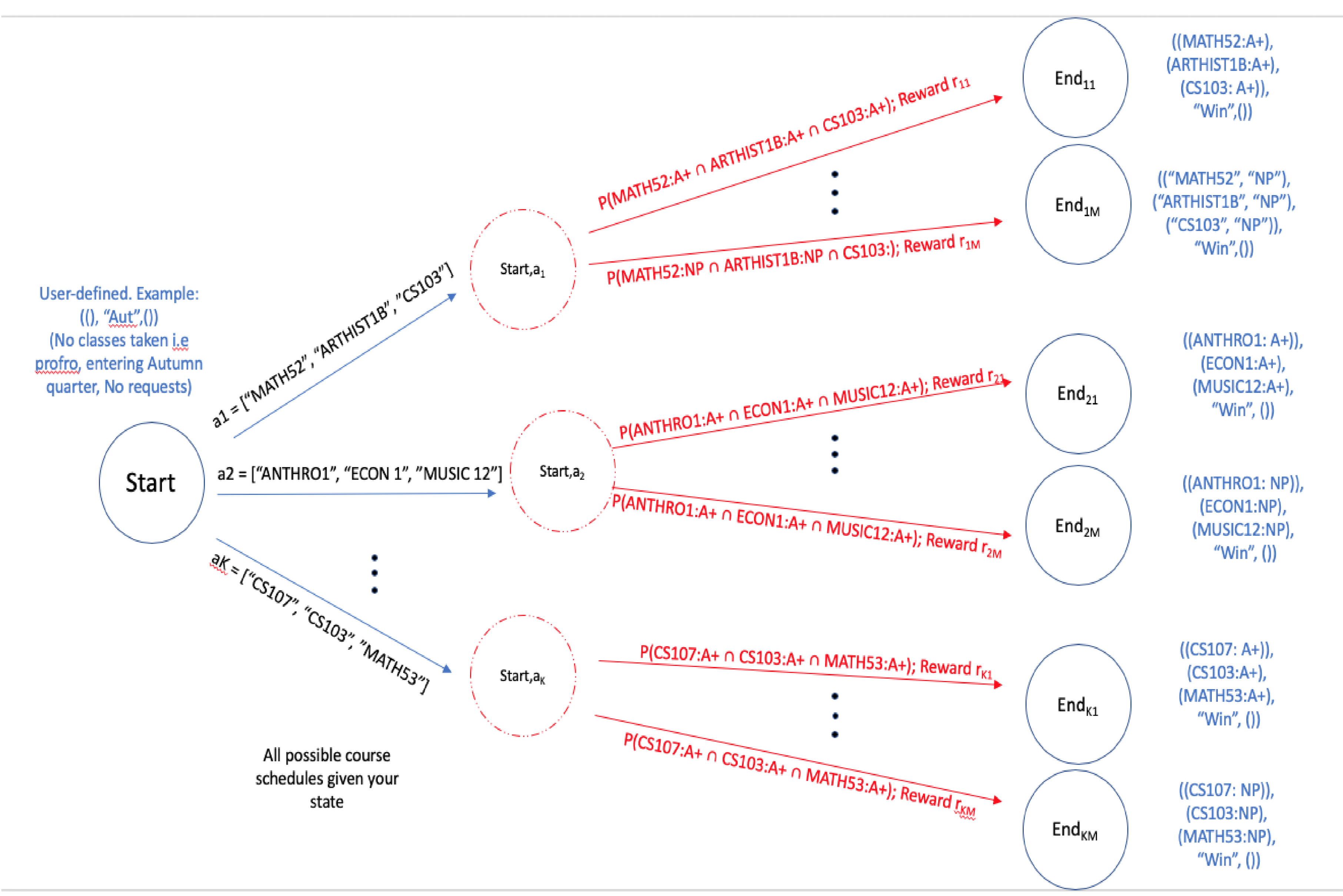
Project Goal:
- Our aim is to make the process of course scheduling stress-free and successful.
- In particular, we will recommend a schedule for the upcoming quarter to a CS student in the AI track given their previous classes and their grades.

2. Approach

1. Data Acquisition
- We used CARTA to manually create a bulletin of ~100 courses. We recorded information like grade distribution, time commitment, prerequisites, quarters offered and course rating.



2. MDP Definition
- The user defines the **start state** with a list of previous classes and grades, quarter the user is entering, and class requests.
- The **actions** from this state are valid 3-class course schedules given the classes the user has taken.
- **Rewards** are a combination of, **course rating**, **grade achieved**, **time-commitment**, and **progress toward graduation**.
- We originally use the grade distributions from CARTA as our **transitions probabilities.** We came up with a better method which we discuss in the "Challenges" section.
- An **end state** is any state with a quarter after the one defined by the user.
- In effect every state but the start state is an end state.



3. Policy Generation
- We used **Value Iteration** to generate the optimal course schedule.

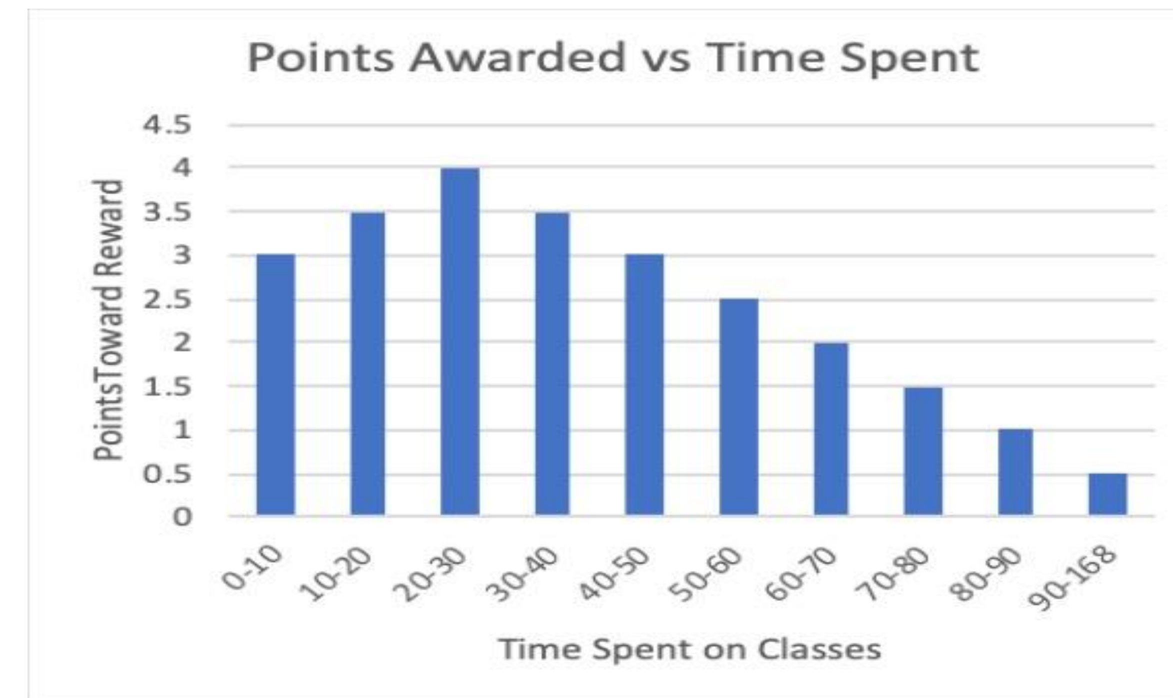
3. Challenges

State Space:
- Massive state space due to large branching factor.
- Handled by limiting MDP to depth of one quarter. This can lead to greedy results though.

Dependence of Grades
- Original approach ignores dependence of grades.
- To fix, assume that in classes with similar focuses students will perform in about the same percentile.

Encouraging Graduation
- Fraction of classes that complete a grad requirement.

Avoiding Overly Difficult Schedules
- Lower reward for more time commitment.



4. Results

Sample of Test Data

	Input	Output
1	MATH 19/20 : A CS 106A: A Frosh Autumn	CS 106B MUSIC 19A ARTSTUDI 170
2	MATH 19/20/21/51: A+ CS 106A: A+ Frosh Winter	MATH 53 CS 106B MUSIC 19A
3	MATH 19/20/21/51: C CS 106A: A Frosh Winter	CS 106B ANTHRO 1 MUSIC 19A
4	All WAYS complete: A MATH 19/20: A MATH 21/51: A- Sophomore Spring	CS 103 E 40M CS 278
5	MATH 19/20/21/51/104: A CS 106A/B/107/103/109: A PSYCH 1: A Sophomore Autumn	CS 230 CS 238 CS 274

Overall Results

Below is a chart comparing the percent of reasonable schedules produced by the **baseline** (picking the top 3 classes by rating), the **oracle** (student with hindsight), and our model.

5. Analysis

Test 1:
- This schedule has a nice balance of exploration, difficulty, enjoyable classes and completion of graduation requirements.

Tests 2 & 3:
- Here, we see that the system understands the difference between a student who has performed well in math, and one hasn't, and only recommends additional math classes to the student who has performed well. This is a direct result of using percentiles to reflect the dependencies between grades in similar classes.
- Also, we noticed that MUSIC 19A was showing up in many of our schedules. This is because the class has very high grades (the mode is A+), great reviews, and a relatively light workload.

Test 4:
- Here, the system avoids WAYs classes, since the student has already completed them. It also recommends a great class that many students (ourselves included) are unfamiliar with – CS 278. This is exactly what we wanted our model to do.

Test 5:
- This schedule is probably too difficult to be reasonable, and shows that we still have to work on shaping our points awarded vs time spent curve (seen above). What likely happened here is that the model knows that you do well in tech classes, but doesn't have much information on WAYs classes, so it takes the safe route and recommends just tech classes. We need harsher punishment for taking multiple difficult classes at the same time.

6. Future Work

1. Fine-tune weights to create a more balanced course schedule.
2. Extend beyond AI track.
3. Increase our MDP depth to a one-year period instead of one quarter. This will help avoid greedy policies.