EECS 496: Sequential Decision Making

Soumya Ray

sray@case.edu

Office: Olin 516

Office hours: T 4-5:30 or by appointment

Today

- Intro to course mechanics

What is this course about?

- Sequential Decision Making is a subfield of AI
- How to make a sequence of decisions to reach some goal or maximize some utility function
- Integrate reasoning, learning and acting: a "full" AI agent

What is this class about?

- Languages for representation and reasoning (overlaps with 391 and 491)
- Understand different tradeoffs in designing methods for SDM; theory and empirical evaluation
- Goal: Be able to think about, design, implement and analyze SDM methods of your own

Syllabus

- Representation Languages and Inference (4 weeks)
- Automated Planning (5 weeks)
- Reinforcement Learning (5 weeks)
- Advanced topics / review (1 week)??

Canvas

- “Assignments:” submit written assignments (pdf) here

- “Files:” Lecture Notes, Book Chapters and assignment descriptions, data and solutions

https://csevcs.case.edu

- Git server that we will use to submit/track programming exercises
- Please sign in with your Case ID by Friday
  - Needed to create a repository for you
  - (Go to website and sign in; don't use ssh)
- Learn to use git

Textbooks

- Artificial Intelligence: A modern approach 3ed (Russell and Norvig) for part 1
- Automated Planning and Acting by Ghallab, Nau and Traverso (2016 ed) for part 2
- Reinforcement Learning: An Introduction by Sutton and Barto 2ed (2016) for part 3
- Other material will be linked from course website as needed

Office hours and TA

- T 4-5:30pm
  - Or by appointment

- TAs: ?????

Email

- You can send me email at sray@case.edu
- It will help if you prefix the subject line with "EECS 496"
  - E.g. "EECS 496: Question about homework 1"
- Occasionally I will send email to the class list
  - This goes to your official address in SIS, make sure this can receive email

Assignments

- Written and Programming assignments generally due Friday 11:59pm
- Written
  - Around 2-3 questions per week
  - Submit pdfs for each question individually through Canvas
  - All questions can be answered from lecture slides, or additional reading as indicated

Programming Exercises

- Programming in Python or Java
  - 3 programming exercises (1 per part)
  - First part: can use python or java
  - 2 and 3 must be in Java

Programming: SEPIA

- We will use a strategy game as a framework for the programming exercises in the second and third parts
  - **S**trategy **E**ngine for **P**rogramming **I**ntelligent **A**gents
  - Written in Java
  - Similar to Age of Empires, Warcraft, Starcraft etc, but modified for teaching and research
  - Developed by students here
  - Needs Java 1.8
- You will write code for players (“agents”) that will learn to play (parts of) this game

Programming Exercises

- *Weekly commits* in csevcs.case.edu for multi-week exercises
  - Does not necessarily have to be working code, but must show significant progress each week
  - Supplemented by a README that describes what you have done so far

Class Project

- Read papers on recent advances in planning and RL
- Implement and evaluate algorithms, and compare their behavior
- Code submitted through git repository

Grading

- Written Problems: 30%
- Programming Problems: 35%
- Project: 35%
- Quality Class Participation: 5%

Bonuses to your grade

- Any assignment or commit turned in a week or more in advance +10% to the score
  - Need to notify us of early commits by email
- Programming assignments that are very well written, easy to read and efficient may get bonus points (see description)
- Partial credit is available for all written assignments
- Class Participation points

Penalties to your grade

- One late day each week (up to Saturday 11:59 pm)
  - If Canvas marks your assignments as late, you will receive a 20% penalty to the score
- Submissions after Saturday 11:59pm for a week will NOT be graded
  - (If you have a genuine reason, such as illness, please email or talk to me and ask for an extension)

Quality of Work

- Graduate level work is expected
  - Neatly written or typed answers
  - Clear arguments, no steps omitted proofs
  - Well structured, clean and efficient code that is properly documented

Collaboration policy

- All assignments must be done in groups of 3
  - Each group submits one assignment
  - Self signup enabled in Canvas
  - It is expected that everyone will pull their weight
    - Must contribute equally to all work done
    - If not, person doing less work will be penalized
    - For written assignments, names of everyone who contributed must appear on the assignment
- Projects will be done by groups of 6

Academic Integrity Policy

- You are free to talk to fellow students, TAs/mentors and me about assignments to clarify/refine your ideas

- But any submitted work MUST be substantially your own
  - Do not copy from any source including any online sources
  - Do not put your name on an assignment where your partner did the work
  - Any violation will be penalized up to failure in the class

Course Load

- I recommend setting aside about 6-8 hours each week to work on this course (besides class hours)

- Do timely work

Expected background

- Computer science concepts (algorithms, runtime/space, data structures etc)
- Strong programming skills
- Probability and Statistics

- Helpful
  - General exposure to Artificial Intelligence
  - Optimization

Questions?

Groups

- Similar academic Level
- Schedule Alignment

SEPIA tutorial

- http://engr.case.edu/ray_soumya/sepia/html/

Todo list

- Sign in to csevcs.case.edu
- Form groups
- Communicate with partners, set up schedule and allocate block of time for this class
- Set up your development environment; learn/refresh git