CS 4180/5180: Reinforcement Learning and Sequential Decision Making (Spring 2024)

1 General Information

Time: Monday, Wednesday 2:50–4:30 Location: 335 Shillman Hall

2 Teaching Staff

- **The preferred platform for asking questions and contacting staff is Piazza.**
- Only e-mail individual staff if absolutely necessary, and note that response will typically be slower than contacting all staff via Piazza.

Role	Name and E-mail	Office Hours	Location
Instructor	Lawson L.S. Wong	Fri 3:30–5	513 ISEC
	Contact via Piazza	and by appointment	and Zoom 617 373 7459
TA	Yu Qi	Thu 4–6	7/F EXP common area
	`qi.yu2@northeastern.edu`		and Zoom 757 826 7655
TA	Adithya Ramesh	Tue 4–6	7/F EXP common area
	`ramesh.adi@northeastern.edu`		and Zoom 987 3781 2634
TA	Linfeng Zhao	Wed 5–7	7/F EXP common area
	`zhao.linf@northeastern.edu`		and Zoom:
			`https://northeastern.zoom.us/my/linfeng`

3 Course Overview

This course will introduce students to the fundamentals of reinforcement learning (RL) and sequential decision making, as well as a selection of recent advances in the field. The course is centered around the framework of Markov decision processes (MDPs), including aspects of modeling, planning, and learning in MDPs.

The content of the course emphasizes the foundational ideas and algorithms of reinforcement learning, and only lightly covers a small portion of the recent advances that have generated much recent excitement in the field. Due to the rapidly developing nature of the field, and partly because we believe it is premature to consider the latest developments, we will omit most algorithms that were introduced in the past five years. Nevertheless, by the end of the course, students will have developed a sufficiently broad set of technical tools, and have gained sufficent depth of understanding in the core concepts, that will enable them to solve real-world problems, self-learn additional techniques, and pursue further specialized content in reinforcement learning. Students are also encouraged to investigate recent methods in their course projects.

The course material will focus on problem formulation, models, and algorithms. Applications will be discussed when relevant, but will not be the focus of the content. However, in the spirit of experiential learning, there will be significant opportunities for implementation and application, through the programming assignments and the project.

4 Textbook and Reference Materials

The main textbook is *Reinforcement Learning: An Introduction* (2nd edition), by Richard S. Sutton and Andrew G. Barto. This textbook gives a broad introductory overview of most topics that are covered in this class. We will follow the book closely for the first two months of the course, covering Part I and sections of Part II in detail.

- *Reinforcement Learning* (2nd edition), by Richard S. Sutton and Andrew G. Barto.
  Website (with free PDF): `http://www.incompleteideas.net/book/the-book-2nd.html`

In the second half of our course (after Spring Break), we will be covering more recent material drawn from recent papers in reinforcement learning. Links to relevant papers will be provided then.

When learning, sometimes it helps to be exposed to multiple perspectives of the same content. The following material from courses at other institutions is optional, but may be helpful to view when the textbook / lectures are insufficient:

- DeepMind × UCL: RL Lecture Series (2021)
  Lecture videos:
  `https://www.youtube.com/playlist?list=PLqYmG7hTraZDVH599EItlEWsUOsJbAodm`
- UC Berkeley: CS 285 – Deep Reinforcement Learning (2023)
  Course website: `https://rail.eecs.berkeley.edu/deeprlcourse`
  Lecture videos:
  `https://www.youtube.com/playlist?list=PL_iWQOsE6TfVYGEGiAOMaOzzv41Jfm_Ps`

5 Prerequisites

- All programming assignments must be completed in Python 3. In the second half of the course, you will need to install and use OpenAI Gym and PyTorch. A Linux-based OS such as Ubuntu is recommended, but other common operating systems should work as well.
- Familiarity with mathematical concepts, including probability, optimization, linear algebra, and calculus. Familiarity with reading pseudocode and implementing algorithms.
- For implementing deep RL algorithms such as those covered in the final part of the course, it may help to have access to a CUDA-enabled GPU, but it is not strictly necessary.

6 What is Where

- Piazza: `https://piazza.com/northeastern/spring2024/cs41805180`
  Content: Announcements, lecture slides, assignments, materials, Q&A, discussion.
  Preferred platform for contacting course staff.
  Piazza is our hub; if in doubt, try to look/ask on Piazza first.
  The site also offers an excellent discussion forum, where both instructors and fellow students can answer questions. Everyone is encouraged to participate. Questions/notes can be posted anonymously or with identity, and may also be posted privately only to instructors. Note that posting questions/notes via Piazza will most likely result in faster responses compared to e-mailing individual instructors.

- Canvas: `https://northeastern.instructure.com/courses/170955`
  Content: Assignments, grades.
  The main exception to the "everything is on Piazza" rule is that we will use Canvas for assignments and grades. All assignments should be submitted on Canvas, and all grades/feedback will be posted on Canvas. Assignment files may be duplicated on Piazza for convenience, but the version on Canvas supersedes the rest.

7 Coursework

Type	Total weight	Frequency	Due dates
Exercises	60%	$\sim$ Weekly ($\sim$ 8–10 total)	Friday (11:59 PM)
Project	40%	1 total	See schedule below

- Exercises are based on the previous week of material; we roughly cover one topic per week. Exercises may include both written and programming components. Students may discuss the problems with other students, but must write up their own solutions / code up their own implementations. On each assignment, please also indicate who you discussed with (if any). Also see Collaboration Policy below.
  Lateness: Up to two days late (24-hour period), penalized by 5% per day.
- The project offers an opportunity to apply learned techniques on a substantial problem that interests the student. The project should be completed in teams of 1–3 (ideally 2). Further details and (non-exhaustive) topic suggestions will be provided in February. Here is a rough timeline for the project, but is subject to change:
  - March 8: Project proposal due
  - April 12: Milestone
  - April 17/22/24: Presentation
  - April 19: Draft report
  - April 25: Final report
- Students enrolled in CS 4180 will complete shorter versions of selected exercises, and will be graded more leniently overall.

8 Academic Integrity

We encourage collaboration and discussion in the class, as long as help is fully acknowledged. Discussion of high-level ideas is generally permitted (except during exams), but submissions should always be your own work (except project, which should be your team's work). For specific collaboration constraints on different parts of the course, see the Coursework section above.

This collaboration policy applies equally to recent AI tools such as ChatGPT and other large language models (LLMs). You are welcome to explore these tools and engage in discussion with them, but you should not be copying answers/code from them. Additionally, you should acknowledge usage of such tools and include your work (interaction) with them.

Cheating and other acts of academic dishonesty will be referred to OSCCR (office of student conduct and conflict resolution) and the Khoury College of Computer Sciences.

9 Schedule (subject to change; version 20240404)

Date	Lec #	Topic	Reference	Assignments due (Fri 11:59 PM)
1/8	1	Course overview	Ch. 1	
1/10	2	Multi-armed bandits	Ch. 2	
1/15		MLK Day (no class)		
1/17	3	Multi-armed bandits	Ch. 2	
1/19				Ex. 0: Introduction
1/22	4	Markov decision processes	Ch. 3	
1/24	5	Markov decision processes	Ch. 3	
1/26				Ex. 1: Bandits
1/29	6	Dynamic programming	Ch. 4	
1/31	7	Dynamic programming	Ch. 4	
2/2				Ex. 2: MDPs
2/5	8	Monte-Carlo methods	Ch. 5	
2/7	9	Monte-Carlo methods	Ch. 5	
2/9				Ex. 3: Dynamic programming
2/12	10	Monte-Carlo methods	Ch. 5	
2/14	11	Temporal-difference learning	Ch. 6	
2/16				Ex. 4: Monte-Carlo
2/19		Presidents' Day (no class)		
2/21	12	Temporal-difference learning	Ch. 6	
2/23				Ex. 5: Monte-Carlo + TD
2/26	13	Function approximation	Ch. 9	
2/28	14	Function approximation	Ch. 9	
3/1				Ex. 6: Temporal-difference
3/4		Spring break (no class)		
3/6		Spring break (no class)		
3/8				*Project proposal*
3/11	15	Deep Q-Networks	Ch. 16.1, 16.5	
3/13	16	Deep Q-Networks	Ch. 16.1, 16.5	
3/15				Ex. 7: Function approximation
3/18	17	AlphaGo	Ch. 8.8–8.11, 16.6	
3/20	18	AlphaGo	Ch. 8.8–8.11, 16.6	
3/25	19	Policy-gradient methods	Ch. 13	
3/27	20	Policy-gradient methods	Ch. 13	
4/1	21	Guest lecture: Equivariant RL (Dian Wang)		
4/3	22	Advanced policy-gradient methods		Ex. 8: Deep RL
4/8	23	Guest lecture: Partially observable RL (Andrea Baisero)		
4/10	24	Guest lecture: Learning for long-horizon planning (Linfeng Zhao)		
4/12				*Project milestone*
4/15		Patriots' Day (no class)		
4/17	25	High-level guidance for RL (Lawson Wong); *Project presentations*		
4/19				*Project draft report*
4/22		*Project presentations*		
4/24		*Project presentations*		
4/25				*Project final report*