CS6700: Reinforcement Learning

Faculty: Prof. Balaraman Ravindran (ravi@cse.iitm.ac.in)
CS36 ; Jan-May Semester 2024 ; Slot: E;
TA(s): Returaj Burnwal (cs21d406@smail.iitm.ac.in)
Sai Shashank GP (ee20b040@smail.iitm.ac.in)
Vishnu Vinod (cs19b048@smail.iitm.ac.in)
Siddarth C (siddarthc2000@gmail.com)
Argha Boksi (cs21d407@smail.iitm.ac.in)
Kalyan Nadimpalli (nkalyanv@gmail.com)

I. Learning Outcomes

Reinforcement Learning is a technique used to make sequential decisions in stochastic environments. The generality of its definition allows us to approach key problems in domains ranging from robotics to advertising. The vast modeling capability of deep learning gave new life to the field, evident in breakthroughs such as Deepmind's AlphaGo Zero and OpenAI's DoTA 2 Bot. By the end of this course, the student will have an understanding of the core principles and major advances in the subject and will be introduced to the relatively modern area of deep reinforcement learning.

II. Course prerequisite(s)

Pattern Recognition and Machine Learning (CS5691 or equivalent)

III. Classroom Mode

Offline lectures and online tutorials (click to join) in scheduled slots. Unless notified there will be 1 tutorial and 3 lectures per week.

IV. Mode of Communication

We will use discord and moodle for communication.

V. Textbooks and References

Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction, Second Edition

VI. Course Requirements

You are *required* to attend all the lectures. If you miss any of them it is your responsibility to find out what went on during the classes and to collect any materials that may be handed out. You are required to adhere to the consequently enlisted submission deadlines of reports/assignments.
Class participation is strongly encouraged to demonstrate an appropriate level of understanding of the material being discussed in the class.

VII. Planned Syllabus

1) **Bandits:** Explore-exploit dilemma, Value functions, Multi-armed bandits, Contextual bandits.
2) **The Full Reinforcement Learning Problem:** Evaluative feedback, Non-associative learning, RL vs MPC, Rewards and returns, Markov Decision Processes, Value functions, Optimality and approximation
3) **Bellman Equations & Dynamic Programming:** Bellman Equation & Optimality, Value iteration, Policy iteration, Asynchronous DP, Generalized Policy Iteration
4) **Evaluation & Control:** TD learning, SARSA, Q-learning, Monte Carlo RL

5) **Function Approximation & Value-based Methods:** Maximization Bias, Double Q learning, Value function approximation, Gradient descent methods, Linear function approximation, LSTD, LSTDQ, LSPI, Fitted Q, Deep Q learning, Double DQN, Prioritized Experience Replay, Duelling Architectures, Expected SARSA
6) **Policy Gradient & Actor Critic Methods:** REINFORCE, PG Theorem, Actor Critic methods, Baselines, A2C, A3C, Deterministic PG and DDPG, SAC, Constrained Policy Optimization
7) **Hierarchical RL:** Intro to Hierarchies, SMDPs, Options framework and Option discovery, DDO, MAXQ framework, HAMs
8) **POMDPs:** Definitions, Belief States, Solution Methods: Q-MDPs, LSTMs, Direct Solutions, PSR
9) **Model-based RL:** Connections to Planning, Types of MBRL, RL with a Learnt Model, Dyna-style models, Latent variable models, Implicit MBRL

VIII. Lecture Schedule

The lecture schedule is available here. Please check this sheet regularly for updates.

IX. Grading Scheme

- Written Assignments - 24%
- Programming Assignments - 34%
- Written Exam 1 - 14%
- Written Exam 2 - 14%
- Tutorials - 14%

X. Written Assignments

The written assignments are compulsory for all the students. There will be 2 written assignments during the course. These assignments encourage students to read and appreciate the course content beyond the standard textbook chapters by referring to several technical papers and tutorials. Both assignments carry equal weightage and are to be done individually.

XI. Programming Assignments

The programming assignments focus on the implementation of standard RL algorithms. 3 programming assignments will be carried out in Gymnasium or related RL environments. The students, as teams of two at most, are expected to submit well-documented code and a detailed report. Grading Scheme for first programming assignment is 10% and other two is 12%.

XII. Tutorials

There will be 7 graded tutorials where each tutorial has equal weightage. Tutorials will be discussed on Thursdays unless otherwise notified. The students are expected to complete tutorials individually and submit them by 11:59PM on the following day (Friday, unless otherwise notified).

XIII. Penalty

Submissions received one day past the deadline will incur a 5% penalty, while those received two days late will face a 10% penalty. No submissions will be accepted beyond the two-day grace period.

XIV. Academic Honesty

Academic honesty is expected from each student participating in the course. NO sharing (willing, unwilling, knowing, unknowing) of reports/assignments between students, submission of downloaded material (from the Internet, Campus LAN, or anywhere else) is allowed.

The project work done as a part of this course cannot be used as-is, to meet any other degree requirements. The project must NOT be copied/downloaded material from the Internet or elsewhere.

Academic violations will be handled by IITM Senate Discipline and Welfare (DISCO) Committee. Typically, the first violation instance will result in ZERO marks for the corresponding component of the Course Grade and a drop of one- penalty in overall course grade. The second instance of code copying will result in a 'U' Course Grade and/or other penalties. The DISCO Committee can also impose additional penalties.

Please protect your Moodle account password. Do not share it with ANYONE. Do not share your academic disk drive space on the Campus LAN.