Welcome to Winter 2020 edition of CME 241:

Reinforcement Learning for Stochastic Control Problems in Finance

Instructor: Ashwin Rao

• Classes: Wed & Fri 4:30-5:50pm. Bldg 380 (Sloan Mathematics Center - Math Corner), Room 380w

• Office Hours: Fri 2-4pm (or by appointment) in ICME M05 (Huang Engg Bldg)

Overview of the Course

- Theory of Markov Decision Processes (MDPs)
- Dynamic Programming (DP) Algorithms
- Reinforcement Learning (RL) Algorithms
- Plenty of Python implementations of models and algorithms
- We apply these algorithms to 5 Financial/Trading problems:
  - (Dynamic) Asset-Allocation to maximize Utility of Consumption
  - Pricing and Hedging of Derivatives in an Incomplete Market
  - Optimal Exercise/Stopping of Path-dependent American Options
  - Optimal Trade Order Execution (managing Price Impact)
  - Optimal Market-Making (Bid/Ask managing Inventory Risk)
- By treating each of the problems as MDPs (i.e., Stochastic Control)
- We will go over classical/analytical solutions to these problems
- Then we will introduce real-world considerations, and tackle with RL (or DP)
- The course blends Theory/Mathematics, Programming/Algorithms and Real-World Financial Nuances

Learning Material will be a combination of

- Technical Documents/Lecture Slides I have prepared specifically for this course
- Python codebase I have developed for this course to help you "learn through coding"
- Slides and Videos from David Silver's UCL course on RL
- For deeper self-study and reference, augment the above content with The Sutton-Barto RL Book and Sutton's accompanying teaching material

Lecture-by-Lecture (tentative) schedule with corresponding lecture slides, reading/videos, and assignments

Date	Lecture Slides	Reading/Videos	Suggested Assignments
January	Course Overview		

8		• **First (Introduction) chapter of Sutton-Barto (pages 1-12)** • **Optional: Rich Sutton's corresponding slides on Intro to RL** • **Optional: David Silver's slides on Intro to RL** • **Optional: David Silver's corresponding video (youtube) on Intro to RL**	• **Register for the Course on Piazza** • **Install/Setup on your laptop with LaTeX, Python 3 (and optionally Jupyter notebook)** • **Create a git repo for this course where you can upload and organize all the code and technical writing you will do as part of assignments and self-learning** • **Send me a message on Piazza with your git repo URL, so I can periodically review your assignments and other self-learning work**
January 10	**Markov Processes (MP) and Markov Reward Processes (MRP)**	• **Optional: David Silver's corresponding video (on youtube) on MPs/MRPs/MDPs**	• **Write out the MP/MRP definitions and MRP Value Function definition (in LaTeX) in your own style/notation (so you really internalize these concepts)** • **Think about the data structures/class design (in Python 3) to represent MP/MRP and implement them with clear type declarations** • **Remember - your data structure/code design must resemble the Mathematical/notational formalism as much as possible** • **Specifically the data structure/code design of MRP should be incremental (and not independent) to that of MP** • **Separately implement the r(s,s') and the R(s) = \sum_{s'} p(s,s') * r(s,s') definitions of MRP** • **Write code to convert/cast the r(s,s') definition of MRP to the R(s) definition of MRP**

			(put some thought into code design here) • **Write code to generate the stationary distribution for an MP**
January 15	**Markov Decision Processes (MDP), Value Function, and Bellman Equations**	• **Third (MDP) chapter of Sutton-Barto book (pages 47-67)** • **Optional: Rich Sutton's corresponding slides on MDPs** • **Two ways of arriving at the identical MRP from an MDPRefined (r(s,s',a) definition)**	• **Write the Bellman equation for MRP Value Function and code to calculate MRP Value Function (based on Matrix inversion method you learnt in this lecture)** • **Write out the MDP definition, Policy definition and MDP Value Function definition (in LaTeX) in your own style/notation (so you really internalize these concepts)** • **Think about the data structure/class design (in Python 3) to represent MDP, Policy, Value Function, and implement them with clear type definitions** • **The data structure/code design of MDP should be incremental (and not independent) to that of MRP** • **Separately implement the r(s,s',a) and R(s,a) = \sum_{s'} p(s,s',a) * r(s,s',a) definitions of MDP** • **Write code to convert/cast the r(s,s',a) definition of MDP to the R(s,a) definition of MDP (put some thought into code design here)** • **Write code to create a MRP given a MDP and a Policy** • **Write out all 8 MDP Bellman Equations and also the transformation from Optimal Action-Value function to**

			Optimal Policy (in LaTeX)
January 17	**Dynamic Programming Algorithms**	• **Fourth (Dynamic Programming) chapter of Sutton-Barto book (pages 73-88)** • **Understanding Dynamic Programming through Bellman Operators** • **Optional: Rich Sutton's corresponding slides on Dynamic Programming** • **Optional: David Silver's video (on youtube) on Dynamic Programming**	**Further assignment suggestions will show up here as the course progresses ...**
January 22	**Understanding Risk-Aversion through Utility Theory (as a pre-req for Finance Applications)**	• **Optional (Related) Reading: A Terse Introduction to Efficient Frontier Mathematics**	
January 24	**Application Problem 1 - Optimal Asset Allocation/Consumption (Merton's 1969 Portfolio Problem)**	• **Optional Review: Stochastic Calculus Foundations (used in setting up HJB)** • **Reference: A paper on *A Deep RL Framework for Optimal Asset Allocation*** • **Some (rough) pointers on Discrete versus Continuous MDPs, and solution techniques**	
January 29	**Application Problem 2 - Pricing and Hedging of Derivatives in Incomplete Markets and Application Problem 3 - Optimal Exercise of American Options**	• **Optional Review: Foundations of Arbitrage-Free and Complete Markets** • **Reference: JP Morgan Research paper on Deep Hedging** • **Reference: Longstaff-Schwartz paper on Pricing American Options (industry-standard approach)** • **Reference: A paper on *RL for Optimal Exercise of American Options***	
Januuary 31	**Application Problem 4 - Optimal Trade Order**	• **Reference: Bertsimas-Lo**	

	Execution and Application Problem 5 - Optimal Market-Making	**paper on Optimal Trade Order Execution** • **Reference: Almgren-Chriss paper on Risk-Adjusted Optimal Trade Order Execution** • **Reference: Avellaneda-Stoikov paper on Optimal Market-Making**	
February 5	**Industry Guest Lecture**		
February 7	**Model-free (RL) Prediction With Monte Carlo and Temporal Difference**	• **Optional: David Silver's corresponding video (youtube) on Model-free Prediction** • **Monte-Carlo and TD (Model-Free) Prediction sections from Sutton-Barto textbook (pages 91-95, 119-128)**	
February 10-11	**Take-home Midterm Exam**		
February 12	**Model-free (RL) Prediction with Eligibility Traces (TD(Lambda))**	• **Optional: David Silver's corresponding video (youtube) on Model-free Prediction** • **n-Step TD section of Sutton-Barto textbook (pages 141-145)** • **Optional: TD(Lambda) and Eligibility Traces-based Prediction is covered on pages 287-297, but this treatment is for the more general case of function approximation of Value Function (we've only covered tabular RL algorithms so far).**	
February 14	**Model-free Control (RL for Optimal Value Function/Policy)**	• **Optional: David Silver's corresponding video (youtube) on Model-free Control** • **MC and TD-based Control sections of Sutton-Barto textbook (pages 96-111, 129-134, 146-149)**	

		• **Optional: SARSA(Lambda) is covered on pages 303-307, but this treatment is for the more general case of function approximation of Value Function (we've only covered tabular RL algorithms so far)**	
February 19 and 21	**RL with Function Approximation (including Deep RL and Batch Methods)**	• **Optional: David Silver's corresponding video (youtube) on RL with Function Approximation** • **Function Approximation sections of Sutton-Barto textbook (pages 197-210, 222-230, 243-248)** • **Optional: Original DQN paper and Nature DQN paper** • **Optional: Lagoudakis-Parr paper on Least Squares Policy Iteration (LSPI)**	
February 26 and 28	**Value Function Geometry and Gradient TD**		
March 4	**Policy Gradient Algorithms**	• **Optional: David Silver's corresponding video (youtube) on Policy Gradient Algorithms** • **Policy Gradient chapter of Sutton-Barto textbook (pages 321-332, 335-336)** • **Optional: Original Paper on Policy Gradient**	
March 6	**Integrating Learning and Planning**	• **Optional: David Silver's corresponding video (youtube) on Integrating Learning and Planning** • **Chapter of Sutton-Barto textbook on Integrating Learning and Planning (pages 159-188)**	
March 11	**Exploration versus Exploitation**	• **Optional: David Silver's corresponding video (youtube) on Exploration versus Exploitation**	

		• **Chapter of Sutton-Barto textbook on Multi-Armed Bandits (pages 25-41)**	
March 13	**Special Topics: Evolutionary Strategies and Adaptive Multistage Sampling/Monte-Carlo Tree Search Algorithms**	• **Reference: Chang, Fu, Hu, Marcus paper on Adapative Multistage Sampling** • **Reference: OpenAI Research paper on Evolutionary Strategies as an Alternative to RL**	• **Upload all of your assignment work on the github account you had created at the start of the course** • **Send me a message on Piazza that you have uploaded all of your work** • **Ensure that all of your assignment work can be accessed by me at github.com (make sure you have pushed and not just commited)**
March 17-18	**Take-home Final Exam**		

Grade will be based on

- 25% Mid-Term Exam (on Theory, Modeling and Algorithms)
- 40% Final Exam (on Theory, Modeling and Algorithms)
- 35% Assignments: Programming, Technical Writing and Theory Problem-Solving (to be done throughout the course)

Purpose and Grading of Assignments

- Assignments are not to be treated as "tests/exams" with a right/wrong answer
- Rather, they should be treated as part of your learning experience
- You will TRULY understand ideas/models/algorithms only when you WRITE down the Mathematics and the Code precisely
- In other words, simply reading the Mathematics or the Code gives you a false sense of understanding things
- Take the initiative to make up your own assignments, especially on topics you feel you don't quite understand
- Individual assignments won't get a grade and there are no due dates for the assignments
- Rather, the entire body of assignments work throughout the course will be graded (upload regularly on your course git repo)
- It will be graded less on correctness and completeness, and more on:
    - Coding and Technical Writing style that is clear and modular
    - Demonstration of curiosity and commitment to learning through the overall body of assignments work
    - Extent of engagement in asking questions and seeking feedback for improvements