MAE5000 – Markov Decision Processes and Reinforcement Learning

Prof. L. Felipe Martins
Department of Mathematics and Statistics
Cleveland State University
l.martins@csuohio.edu

1 Course Objectives

To provide students with a solid foundation in sequential decision-making processes, with an emphasis on Markov Decision Processes and Reinforcement Learning. By the end of the course, students are expected to be able to formulate decision-making problems under uncertainty, understand the theoretical foundations of value functions and the Bellman equations, implement classical dynamic programming algorithms and modern reinforcement learning methods, and critically evaluate their performance on practical problems.

2 Rationale

Sequential decision-making is one of the fundamental paradigms of modern Artificial Intelligence, with applications in robotics, autonomous systems, finance, healthcare, control, logistics, and recommender systems. Reinforcement Learning, grounded in the theory of Markov Decision Processes, has become one of the most important areas of data science and artificial intelligence, driven by recent advances in deep learning and computational power. This course provides students with the mathematical, computational, and algorithmic foundations necessary to understand, develop, and apply optimization and learning methods to sequential decision-making problems under uncertainty, thereby supporting their research training and enabling the development of applications across a wide range of disciplines.

3 Course Description

This course studies *sequential decision-making processes*, in which an *agent* interacts with an *environment* and receives *rewards*, which are a function of both the selected actions and the current state of the environment. Application areas include robotics, games, recommender systems, healthcare, and control.

The mathematical framework used for this class of problems is known as a *Markov Decision Process*. In this model, once the agent selects a policy, which determines (possibly in a probabilistic manner) the actions to be taken in each state of the environment, the resulting process follows

the dynamics of a *Markov reward process*, a mathematical structure that has been extensively studied. The objective is to maximize the total long-term reward.

The course covers both classical tabular methods (dynamic programming) and modern techniques (reinforcement learning). It addresses both theoretical and practical aspects of designing optimization algorithms. Computational experiments will be carried out using open-source software, including Python, NumPy, Jupyter, Gymnasium, and PyTorch.

4 Prerequisites

Probability and Linear Algebra.

No prior programming experience is assumed, although familiarity with at least one programming language is recommended.

5 Assessment

Assessment components include problem sets, computational laboratories, and a final project presented in class.

6 References

1. Sutton and Barto, *Reinforcement Learning: An Introduction*
   `http://incompleteideas.net/book/RLbook2020.pdf`
2. Powell, *Reinforcement Learning and Stochastic Optimization*
   `https://castle.princeton.edu/wp-content/uploads/2019/10/Powell-Reinforcement-Learning-a pdf`
3. Bertsekas, *A Course in Reinforcement Learning*
   `https://www.mit.edu/~dimitrib/RLCOURSECOMPLETE%202ndEDITION.pdf`
4. Szepesvári, *Algorithms for Reinforcement Learning*
   `https://sites.ualberta.ca/~szepesva/papers/RLAlgsInMDPs.pdf`
5. Plaat, *Deep Reinforcement Learning*
   `https://deep-reinforcement-learning.net`
6. Cai, *An Introduction to Reinforcement Learning*
   `https://rlbook.adzc.ai`
7. Xiao, *Reinforcement Learning: Theory and Python Implementation*
   `https://zhiqingxiao.github.io/rl-book`

7 Topics

1. Markov Decision Processes (MDPs): definition of environment, state, agent, actions, and rewards. The decision-making cycle of an MDP. Examples of MDPs. Modeling examples. Introduction to computational tools.

2. Value Functions and the Bellman Equation: state-value functions and action-value functions. Bellman equations for a given policy. Exact and iterative numerical methods. Contraction property, existence of solutions, and convergence.

3. Optimal Value Functions, the Optimal Bellman Equation, and the Verification Theorem. Solution methods: value iteration and policy iteration. Practical implementation in simple models.

4. Introduction to model-free learning. Multi-armed bandit problems and the exploration–exploitation trade-off. ε-greedy strategies. Upper Confidence Bound (UCB) methods and Thompson Sampling. Computational experiments with ε-greedy strategies.

5. Monte Carlo Methods: prediction and control. On-policy and off-policy methods. Applications to problems of moderate complexity, such as Blackjack.

6. Temporal-Difference (TD) Learning. TD(0), SARSA, and Q-Learning. Multi-step methods, eligibility traces, and TD(λ).

7. Function Approximation. Limitations of tabular methods for model-based and model-free problems. Feature representations, linear function approximation, semi-gradient methods. Policy gradient methods: REINFORCE and gradient-based estimators.

8. Actor–Critic Methods. Advantage functions, variance reduction, and actor–critic architectures.

9. Deep Reinforcement Learning. Introduction to neural networks, Deep Q-Learning, experience replay, target networks, and modern extensions.