IEOR 265 - Course Syllabus
Spring 2020

1 Course Staff

Instructor: Pedro Hespanhol
Office Hours: Tu-Th 2:00-3:30PM
Location: 4175 Etcheverry Hall
e-mail: pedrohespanhol@berkeley.edu

GSI: Caleb Bugg
Office Hours: Mo 1:30-2:30PM
Location: 4176 Etcheverry Hall
e-mail: caleb_bugg@berkeley.edu

GSI: Salar Fattahi
Office Hours: Fr 10:00-11:00AM
Location: 4176 Etcheverry Hall
e-mail: fattahi@berkeley.edu

Lectures: Tu-Th 3:30-5:00PM
Location: 3107 Etcheverry Hall

2 Course Description

This course will cover topics related to the interplay between optimization and statistical learning. The first part of the course will cover the fundamentals, methods, and algorithms for dynamic programming, and optimal control. We will study approximate dynamic programming and the formulation and numerical implementation of several different algorithms, and reinforcement learning methods. In addition we will study learning-based model predictive control (LBMPC), which is a method for robust adaptive optimization that can use machine learning to provide the adaptation online. The second part of the course will deal with inverse decision-making problems, which are problems where an agent's decisions are observed and used to infer properties about the agent, such as preferences, utility functions, etc.

2.1 Prerequisites:

Course on Linear Algebra/Real Analysis; Course on Statistics/Stochastic Processes; Some previous course on optimization is good, but not required.

2.2 Reading Material:

Slides and lecture notes will be posted on Bcourses. Many topics covered in the course will be based on papers. But for the most part, we will follow the material on the following textbooks:

1. Reinforcement Learning and Optimal Control - Dimitri P. Bertsekas. Athena Scientific, 2019.

2. Dynamic Programming and Optimal Control, Vol I and II, 4th Edition - Dimitri P. Bertsekas. Athena Scientific, 2017.

3 Grading

The course grade consists of a number of homework problems and a final project. The homeworks will contain a mix of theoretical problems and coding.(**Note:** it is encouraged to use Python in the coding assignments). The weighting is as follows:

Homework: 50%

Project: 50%
(5% Project Proposal; 10% Project Milestone Report; 35% Final Report)

4 Final Project

For the final project, you may work in groups of up to three people. You may work alone, but from past experience it is encouraged to form groups of two or three people (groups of more than three people are **not** allowed).

The projects can be of the following form:

1. (in-depth) literature review, or;

2. a comprehensive application of data analysis methods, or;

3. development and application of the algorithms/methods presented in the course, or;

4. the exploration of original research ideas

The final project contains **three deliverables:**

1. **Project Proposal:** A one-page document, containing an abstract of the project, describing the idea, the application and the planned work for the group. **Deadline: February 20**.

2. **Project Milestone Report:** A 2-3 pages document, describing the main idea and literature references for the project and the progress done so far. (Note: If the project contains coding, please describe here what are the main parts of the code and what is already implemented). **Deadline: March 31**.

3. **Final Report:** The report should be a **6-8 pages** Conference-type paper (IEEE format, see: Template). In addition, the report should contain, in the introduction section, a short paragraph to explain the role and contribution of each group member. **Deadline: May 05**.

5 Course Outline:

Specific topics that will be covered include (tentative schedule):

1. **Approximate Dynamic Programming and Reinforcement Learning**
   (a) Introduction to Dynamic Programming
   (b) Approximation in Value Space (e.g.: Lookahead methods, Rollout, MCTS)
   (c) Approximation in Policy Space (e.g.: Policy Gradient methods)
   (d) Reinforcement Learning (e.g.:Value Iteration, Policy Iteration, Q-learning, etc)
   (e) Model-based RL and MPC
   (f) LBMPC (formulation, oracle, training, optimization)
   (g) Aggregation methods
2. **Inverse Decision-Making and Game Theory**
   (a) Nash Equilibrium and Variational Inequalities (definition, variational representation, properties, etc)
   (b) Learning and Inverse Variational Inequalities
   (c) Inverse RL
   (d) Inverse Optimization