Stanford CME 241 (Winter 2023) - Assignment 5

Due: 2/13 @ 11:59pm. Solve any 3 of the 4 questions.

1. You are a milkvendor and your task is to bring to your store a supply (denoted $S \in \mathbb{R}$) of milk volume in the morning that will give you the best profits. You know that the demand for milk through the course of the day is a probability distribution function f (for mathematical convenience, assume people can buy milk in volumes that are real numbers, hence milk demand $x \in \mathbb{R}$ is a continuous variable with a probability density function). For every extra gallon of milk you carry at the end of the day (supply S exceeds random demand x), you incur a cost of h (effectively the wasteful purchases amounting to the difference between your purchase price and end-of-day discount disposal price since you are not allowed to sell the same milk the next day). For every gallon of milk that a customer demands that you don't carry (random demand x exceeds supply S), you incur a cost of p (effectively the missed sales revenue amounting to the difference between your sales price and purchase price). So your task is to identify the optimal supply S that minimizes your Expected Cost $g(S)$, given by the following:

$$g_1(S) = E[\max(x - S, 0)] = \int_{-\infty}^{\infty} \max(x - S, 0) \cdot f(x) \cdot dx = \int_{S}^{\infty} (x - S) \cdot f(x) \cdot dx$$

$$g_2(S) = E[\max(S - x, 0)] = \int_{-\infty}^{\infty} \max(S - x, 0) \cdot f(x) \cdot dx = \int_{-\infty}^{S} (S - x) \cdot f(x) \cdot dx$$

$$g(S) = p \cdot g_1(S) + h \cdot g_2(S)$$

   After you solve this problem, see if you can frame this problem in terms of a call/put options portfolio problem.

2. rl/chapter8/optimal_bin_tree.py models the American Payoff pricing problem as a FiniteMarkovDecisionProcess in the form of a binary tree with only two discrete transitions for any given asset price. In the world of mathematical and computational finance, it is common practice to work with continuous-valued asset prices and transitions to a continuous set of asset prices for the next time step. Your task is to model this problem as a MarkovDecisionProcess (not finite) with discrete time, continuous-valued asset prices and a continuous-set of transitions. Assume an arbitrary probability distribution for asset price movements from one time step to another, so you'd be sampling from the arbitrary transition probability distribution. Hence, you will be solving this problem with Approximate Dynamic Programming using the code in rl/approximate_dynamic_programming.py.

3. We'd like to build a simple simulator of Order Book Dynamics as a MarkovProcess using the code in rl/chapter9/order_book.py. An object of type OrderBook constitutes the *State*. Your task is to come up with a simple model for random arrivals of Market Orders and Limit Orders based on the current contents of the OrderBook. This model of random arrivals of Marker Orders and Limit Orders defines the probabilistic transitions from the current state (OrderBook object) to the next state (OrderBook object). Implement the probabilistic transitions as a MarkovProcess and use it's simulate method to complete your implementation of a simple simulator of Order Book Dynamics.

   Experiment with different models for random arrivals of Market Orders and Limit Orders.

4. Derive the expressions for the Optimal Value Function and Optimal Policy for the *Linear-Percentage Temporary* (LPT) Price Impact Model formulated by Bertsimas and Lo. The LPT model is described below for all $t = 0, 1, \ldots T - 1$:

$$P_{t+1} = P_t \cdot e^{Z_t}$$

$$X_{t+1} = \rho \cdot X_t + \eta_t$$

$$Q_t = P_t \cdot (1 - \beta \cdot N_t - \theta \cdot X_t)$$

where Z_t are independent and identically distributed random variables with mean μ_Z and variance σ_Z^2 for all $t = 0, 1, \ldots, T - 1$, η_t are independent and identically distributed random variables with mean 0 for all $t = 0, 1, \ldots, T - 1$, Z_t and η_t are independent of each other for all $t = 0, 1, \ldots, T - 1$, and ρ, β, θ are given constants. The model assumes no risk-aversion (Utility function is the identity function) and so, the objective is to maximize the Expected Total Sales Proceeds over the finite-horizon up to time T (discount factor is 1). In your derivation, use the same methodology as we followed for the *Simple Linear Price Impact Model with no Risk-Aversion*.

Implement this LPT model by customizing the class OptimalOrderExecution in

rl/chapter9/optimal_order_execution.py.

Compare the obtained Optimal Value Function and Optimal Policy against the closed-form solution you derived above.