

d.

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

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

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

Want to maximize $g(S)$

$$g_1(S) = \int_S^N (x - S) f(x) dx + \int_N^{\infty} (x - S) f(x) dx$$

$$\begin{aligned} \frac{\partial g_1(S)}{\partial S} &= -(S - S)f(S) + \int_S^N \frac{\partial}{\partial S} (x - S) f(x) dx + \int_N^{\infty} \frac{\partial}{\partial S} (x - S) f(x) dx \\ &= \int_S^{\infty} -f(x) dx \end{aligned}$$

$$g_2(S) = \int_{-\infty}^{-N} (S - x) f(x) dx + \int_N^S (S - x) f(x) dx$$

$$\begin{aligned} \frac{\partial g_2(S)}{\partial S} &= \int_{-\infty}^{-N} \frac{\partial}{\partial S} (S - x) f(x) dx + (S - S)f(S) + \int_N^S \frac{\partial}{\partial S} (S - x) f(x) dx \\ &= \int_{-\infty}^S f(x) dx \end{aligned}$$

$$\frac{\partial g(S)}{\partial S} = -p \int_S^{\infty} f(x) dx + h \int_{-\infty}^S f(x) dx = 0$$

$$-p (1 - F(S)) + h F(S) = 0 \quad \text{where } F'(S) = f(S)$$

$$F(S) = \frac{p}{h+p}$$

It's the same as holding a call option on an asset and a put option on the same asset. Both options have the same strike price.

Stanford CME 241 (Winter 2021) - Assignment 8

Assignments:

1. Assume you are the owner of a bank where customers come in randomly everyday to make cash deposits and to withdraw cash from their accounts. At the end of each day, you can borrow (from another bank, without transaction costs) any cash amount $y > 0$ at a constant daily interest rate R , meaning you will need to pay back a cash amount of $y(1 + R)$ at the end of the next day. Also, at the end of each day, you can invest a portion of your bank's cash in a risky (high return, high risk) asset. Assume you can change the amount of your investment in the risky asset each day, with no transaction costs (this is your mechanism to turn any amount of cash into risky investment or vice-versa). The key point here is that once you make a decision to invest a portion of your cash in the risky asset at the end of a day, you will not have access to this invested amount as cash that otherwise could have been made available to customers who come in the next day for withdrawals. More importantly, if the cash amount c in your bank at the start of a day is less than C , the banking regulator will make you pay a penalty of $K \cdot \cot(\frac{\pi \cdot c}{2C})$ (for a given constant $C > 0$ and a given constant $K > 0$).

For convenience, we make the following assumptions:

- Assume that the borrowing and investing is constrained so that we end the day (after borrowing and investing) with positive cash ($c > 0$) and that any amount of regulator penalty can be immediately paid (meaning $c \geq K \cdot \cot(\frac{\pi \cdot c}{2C})$ when $c \leq C$).
- Assume that the deposit rate customers earn is so small that it can be ignored.
- Assume for convenience that the first half of the day is reserved for only depositing money and the second half of the day is reserved for only withdrawal requests.
- Assume that if you do not have sufficient cash to fulfill a customer withdrawal request, you ask the customer to make the withdrawal request again the next day.
- Assume all quantities are continuous variables.

Model an MDP so you can run the bank in the most optimal manner, i.e., maximizing the Expected Utility of assets less liabilities at the end of a T -day horizon, conditional on any current situation of assets and liabilities. Specify the states, actions, transitions, rewards with precise mathematical notation (make sure you do the financial accounting from one day to the next precisely).

Based on the methods you have learnt so far (DP and ADP), how would you go about solving this MDP Control problem?

2. You are a milk vendor 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.

3. [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](#).