Machine Learning for Economic Analysis
Problem Set 10

Jonas Lieber*

Due: 11:59pm Fri, April 19, 2024

Problem 1. *Dynamic Programming*

Imagine you find yourself in the desert. The desert is large. To get out of the it, you need water. You find a water merchant. The water salesman has a monopoly, so prices are high. To buy enough water to get out of the desert, you would have to pay $100. $99 does not get you out of the desert. You only have $ 50 in your pocket in single dollar bills.

There is a casino next to the water merchant. Because it is in the desert, the casino is very frugal. The only game offered is a sequence of coin flips. The success (say 'heads') probability of the coin is $p \in [0, 1]$. *You have $ 50 in single dollar bills in your pocket, so does the casino so that you can win a maximum of $50. The game proceeds in rounds. At every round, you can bet any number of dollar bills that you have. If the coin flip is successful, you get twice your bet - up to a maximum of $ 100 total. If the coin flip is unsuccessful, you loose your bet. In the next round, you can only bet if you have at least 1 dollar bill left. There is only one coin flip per hour, so you use a discount factor of* $\gamma = 0.99$. *You only care about leaving the desert (utility of 10). You do not care about how much money you have unless it's $100. You now wonder how you should bet. Should you put down all your money at once or place small bets etc?*

1. *Write down every component of the MDP.*
2. *Consider the policy of betting $1 every round when it is possible. We are interested to compute the value of this policy when* $p = 0.4, 0.5$ *and* $p = 0.6$.
   (a) *Use linear algebra to evaluate the policy for each* p.
   (b) *Implement value function iteration to evaluate the policy for each* p.

   Do the two approaches yields the same results?
3. *Now implement policy iteration and value function iteration to compute the optimal policy. For each* p, *illustrate the optimal policy in a plot. What is the intuition behind the optimal policy for each* p?

*Department of Economics, Yale University. jonas.lieber@yale.edu

Problem 2. *Multi-Armed Bandits*

Consider a multi-armed bandit with 10 arms. For arm $a \in \{1, \ldots, 10\}$, the reward is distributed as $\mathcal{N}(\mu_a, 1)$, where $\mu_a = a/10$. The agent does not discount rewards and plays the game for $T = 200$ rounds.

1. *Which is the optimal arm? What is the expected sum of rewards after T rounds if you know the optimal arm?*

2. *Implement the explore-then-commit algorithm where you explore each arm for m times. What trade-off do you face when choosing m? Evaluate this strategy using 1000 Monte Carlo simulation for $m = 1, 2, \ldots, 20$ and plot the estimated regret for each m. What is the optimal m according to your simulation?*

3. *Implement the ε-greedy algorithm for some $\varepsilon > 0$. Evaluate this strategy using 1000 Monte Carlo simulation for $\varepsilon = 0.1, 0.2, 0.5, 0.9, 0.95, 0.99$ and plot the estimated regret for each ε. What is the optimal ε according to your simulation?*

4. *Suppose you know that the rewards are distributed as Gaussians with unit variance. Fix some $\delta \in (0, 1)$. After seeing n_a observations of the rewards associated with arm a, what is the smallest $C = C(\delta)$ such that*

$$\mathbb{P}[\mu_a \geq \hat{\mu}_a + C(\delta)] \leq \delta?$$

Implement the UCB-algorithm for $\delta = (10^{-5}, 2 * 10^{5}, 3 * 10^{-5}, 4 * 10^{-5}, 5 * 10^{-5})$ and $\delta = \delta_t = 1/(1 + t * log(t))$ where t a the time index that starts after exploring each arm once with $t = 1$.

Problem 3. *Bayesian Bandit*

Consider a coin with success probability p. If the coin shows heads, you win \$1, if it shows tail, you loose \$1. You are risk-neutral, value money linearly (each \$ gives you a utility of 1) and are extremely patient as the coin flips are very fast so that $\gamma = 1$.

You do not know p but you have a prior that

$$p = \begin{cases} 0.2 \text{ with probability } 0.51, \\ 0.8 \text{ with probability } 0.49. \end{cases}$$

1. *If you could only flip the coin once, would you do it? Why? Compute the expected reward from flipping the coin.*

2. *Now imagine you have the choice to flip the coin or not. After observing the result. You may flip it a second time. Would you flip it? Why? Use the Bayesian prior and backwards induction.*