CSE257 Winter 2023: Assignment 4

Deadline: Mar-19 23:59 PT.

- Submit answer pdf file on Gradescope. **When a question asks you to "Show/Write/List/Plot/Explain" something, you need to show them in the answer pdf. Otherwise points will be taken off.**
- Submit all code as a zip file to `https://www.dropbox.com/request/IPa2QEVyclIld5ostZG3`.

All other requirements are the same as previous assignments.

Question 1 (Implementation Required)**.** Let X be a random variable taking values of either 0 or 1, with probability $P(X = 0) = 0.3$ and $P(X = 1) = 0.7$.

1. (1 Point) Draw N samples of X as $X_1, ..., X_N$ and plot the histogram of the average $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$ with 500 trials (each trial for $\bar{X}_N$ needs N samples of X). Do this for three different values of $N = 1, 10, 500$.

   For the histogram you can divide $[0, 1]$ into 10 bins (i.e., $[0, 0.1)$, $[0.1, 0.2)$, etc., if you want to use more bins to be more accurate, feel free to do so). The x-axis should be the bins for the values of $\bar{X}_N$, and the y-axis should be the frequency of occurrence of $\bar{X}_N$ in each bin. So you should show three plots, one for each different N, and in each one you need do 500 trials (each trial uses N samples), and show how the $\bar{X}_N$ is distributed as a histogram.

2. (2 Points) Let μ be the true expectation of the random variable X specified above. Let $\varepsilon = 0.1$. Plot $P(|\bar{X}_N - \mu| \geq \varepsilon)$ as a function as N grows and compare it with the Hoeffding bound, as follows. Let the x-axis be N going from 1 to 100. For each N, do 10 trials of using N samples to compute the average $\bar{X}_N$, and show the frequency of $|\bar{X}_N - \mu| \geq \varepsilon$ happening on the y-axis (for instance, if you see 3 times of that happening out of 10 trials when $N = 5$ then the y-axis should show 0.3 when $N = 5$).

   On the same plot, show how the Hoeffding bound changes over time as well. That is, plot the righthand side of the Hoeffding inequality (check slides), as a function of N.

Question 2 (Implementation Required)**.** Consider two coins, and write the random variable for the payoff from each coin as $X^{(1)}$ and $X^{(2)}$. The ground truth distribution for each coin is $P(X^{(1)} = 0) = 0.2$, $P(X^{(1)} = 1) = 0.8$, $P(X^{(2)} = 0) = 0.6$ with $P(X^{(2)} = 1) = 0.4$. Plot the regret $R(T)$ as the number of plays T goes from 10 to 500 for each of the following strategies. Recall that the regret is defined as:

$$R(T) = \mathbb{E}[\sum_{i=1}^{T} X_{c^*} - \sum_{i=1}^{T} X_{c_i}], \text{ which you just need to sample as } T\mu^* - \sum_{i=1}^{T} \mu_{c_i},$$

where c^* is the coin with the optimal expected reward, and c_i is the particular choice of coin in each round i. (We say "sample as" because the choice c_i in each round is stochastic in general, but here you only need to consider the particular choice you chose according to each strategy in each round.) In your plot, the x-axis should be T and y-axis should be $R(T)$, and you should plot everything on the same graph so that their regret curves can be compared.

1. (1 Point) Explore-then-commit with $N = \lceil 0.3T \rceil$ ($\lceil \cdot \rceil$ is the ceiling function that gives you an integer).
2. (1 Point) Explore-then-commit with $N = \lceil \frac{1}{2} T^{\frac{2}{3}} (\log T)^{\frac{1}{3}} \rceil$, where $\log$ is the natural logarithm.

3. (1 Point) The ε-Greedy strategy with $\varepsilon = 0.3$. (With $1-\varepsilon$ probability play the currently-best coin, and with ε probability the other).

4. (1 Point) The Upper Confidence Bound strategy, which plays the coin $i \in \{1, 2\}$ that maximizes $\bar{X}^{(i)} + \sqrt{\frac{2 \log T}{N^{(i)}}}$ in each round T.

Question 3 (Implementation Required Only for the Last Sub-Question)**.** Consider the tic-tac-toe game from the previous assignment, and you are still playing for Player O. Answer the following questions conceptually first:

1. (1 Point) Consider an arbitrary initial state, where Player X has put down the first piece, and now you need to do MCTS for Player O. After the first 8 iterations of MCTS from this initial state, what does the search tree look like? Draw it by hand: just the nodes and edges that current exist in the tree. No need to annotate it with any values. (Each iteration refers to the full sequence of using the tree policy, rollout, and backup functions for one round.)

2. (2 Points) Now consider the 9-th iteration of the MCTS algorithm that continues to build the search tree that you wrote down in the previous question. Explain the following:

   (a) Explain how you will choose the next node to expand in the tree.

   (b) Write down one game trajectory that you may see as a sample in the rollout/simulation phase of MCTS at this 9th-iteration, and explain how you choose the actions for each player in this trajectory.

   (c) What nodes are you going to annotate from the outcome of this sample trajectory that you just wrote down? (i.e., changing the win rate and visitation count as part of the backup procedure)

   Again you do not need to implement anything here, just think about what the algorithm will do.

3. (2 Points) Now, consider the game state in Figure 1 again. After 8 iterations of MCTS from this state (i.e., this state is now the root of the MCTS tree), *at most* how many non-root nodes are there in the MCTS tree? Explain your reasoning, which includes drawing the search tree that has this maximum number of nodes.

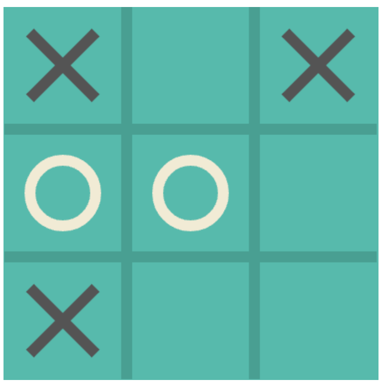

Figure 1: Example of a feasible game state. Your play circle, and it is your turn now.

Now if you have time, implement the MCTS algorithm for tic-tac-toe and answer the following:

4. (Extra 2 Points) Show how the search tree is expanded in the first 10 iterations of MCTS, at the game state in Figure 1. That is, show the search tree obtained in MCTS (you can either hand-draw or print it) at the end of each iteration, for the first 10 iterations.

Question 4 (Implementation Required)**.** Consider the following constraint system. The variables are $X = \{x_1, x_2\}$ with initial domains $x_1 \in D_1 = [0, 2]$ and $x_2 \in D_2 = [0, 2]$. The constraints are

$$\begin{aligned} C_1 &: \; x_1 = x_2^3 \\ C_2 &: \; x_1 = \log(x_2 + 1) \\ C_3 &: \; x_1 \leq \exp(x_2) - 1 \end{aligned}$$

log means natural logrithm. Answer the following questions.

1. (1 Point) Consider C_2 and $x_2 \in D_2$. What is the largest subset of D_1 for x_1 that is arc-consistent with x_2?

2. (2 Points) Implement the backtracking search algorithm based on arc-consistency, to find one satisfying assignment for the constraint system $\{C_1, C_2, C_3\}$ within $D_1 \times D_2$, or report no solution if there isn't any. In the answer pdf, show the solutions you found, if any, and explain where the branching operations happen during solving, if any. Small numerical errors are always allowed.

3. (1 Point) Continuing on the previous question, can you tweak the algorithm to find all satisfying assignments to the system? How do you know that you have exhausted all the possible satisfying assignments? Show all the satisfying assignments you can find for this system.

4. (1 Point) Change the third constraint into

$$C_3' : x_1 \geq \exp(x_2) - 0.9$$

   Use your constraint solving algorithm to find all solutions of the new system under $\{C_1, C_2, C_3'\}$ with the same initial domains $D_1 \times D_2$. Report no solution if there isn't any.

Question 5 (2 Points, No Implementation)**.** Put the following formula into Conjunctive Normal Form (no need to use the general Tseitin expansion as mentioned in the slides, just by expanding the operators you can do this easily, since it is a small formula):

$$\varphi : \neg\Big((p_2 \to (p_3 \to (\neg p_1 \wedge p_2))) \to p_1\Big)$$

then write down in the Dimacs format encoding for SAT solving for this formula, as specified in the slides.