CMPSCI 687 Homework 2

Due October 8, 2018, 11:55pm Eastern Time

Instructions: This homework assignment consists of a written portion and a programming portion. Collaboration is not allowed on any part of this assignment. Submissions must be typed (hand written and scanned submissions will not be accepted). You must use LaTeX. The assignment should be submitted on Moodle as a .zip (.gz, .tar.gz, etc.) file containing your answers in a .pdf file and a folder with your source code. Include with your source code instructions for how to run your code. You may not use any reinforcement learning or machine learning specific libraries in your code (you may use libraries like C++ Eigen and numpy though). If you are unsure whether you can use a library, ask on Piazza. If you submit by October 14, you will not lose any credit. The automated system will not accept assignments after 11:55pm on October 14.

Part One: Written (25 Points Total)

1. (5 Points) Prove that the following two definitions of the state-value function are equivalent:

$$v^\pi(s) \coloneqq \mathbf{E}\left[G_t | S_t = s, \pi\right] \tag{1}$$
$$v^\pi(s) \coloneqq \mathbf{E}\left[G | S_0 = s, \pi\right]. \tag{2}$$

$$\mathbf{E}[G \mid S_t = s, \pi]$$
$$= \mathbf{E}[\sum_{k=0}^{\infty} \gamma^k R_{t+k} \mid S_t = s, \pi]$$
$$= \mathbf{E}[\gamma^0 R_0 + \sum_{t=1}^{\infty} \gamma^t R_t \mid S_t = s, \pi]$$
$$= \sum_{a \in A} \pi(s,a) R(S_t = s, A_t = a) + \mathbf{E}[\sum_{k=1}^{\infty} \gamma^k R_{t+k} \mid S_t = s, \pi]$$
$$= \sum_{a \in A} \pi(s,a) R(S_t = s, A_t = a) + E[\gamma \sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k} \mid S_t = s, \pi]$$
$$= \sum_{a \in A} \pi(S_t = s, A_t = a) R(S_t = s, A_t = a) + E[\gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, \pi]$$
$$= \sum_{a \in A} \pi(S_t = s, A_t = a) R(S_t = s, A_t = a) + \sum_{a \in A} \pi(S_t = s, A_t = a)$$
$$\sum_{s' \in S} P(S_t = s, A_t = a, S_{t+1} = s') E[\gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a, S_{t+1} = s', \pi]$$

Following markov's property

$$= \sum_{a \in A} \pi(S_t = s, A_t = a) R(S_t = s, A_t = a) + \sum_{a \in A}$$

$$\pi(S_t = s, A_t = a) \sum_{s' \in S} P(S_t = s, A_t = a, S_{t+1} = s')$$

$$\gamma E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_{t+1} = s', \pi]$$

This can be expanded further as

$$= \sum_{a \in A} \pi(S_t = s, A_t = a) R(S_t = s, A_t = a) + \sum_{a \in A} \pi(S_t = s, A_t = a) \sum_{s' \in S} P(S_t = s, A_t = a, S_{t+1} = s')$$

$$\gamma[\sum_{a \in A} \pi(S_{t+1} = s, A_{t+1} = a) R(S_{t+1} = s, A_{t+1} = a) + \sum_{a \in A}$$

$$\pi(S_{t+1} = s, A_{t+1} = a) \sum_{s' \in S} P(S_{t+1} = s, A_{t+1} = a, S_{t+2} = s')$$

$$\gamma[E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+2} \mid S_{t+2} = s', \pi]]$$

this can be further expanded as

$$= \sum_{a \in A} \pi(S_t = s, A_t = a) R(S_t = s, A_t = a) + \sum_{a \in A} \pi(S_t = s, A_t = a) \sum_{s' \in S} P(S_t = s, A_t = a, S_{t+1} = s')$$

$$\gamma[\sum_{a \in A} \pi(S_{t+1} = s, A_{t+1} = a) R(S_{t+1} = s, A_{t+1} = a) +$$

$$\sum_{a \in A} \pi(S_{t+1} = s, A_{t+1} = a) \sum_{s' \in S} P(S_{t+1} = s, A_{t+1} = a, S_{t+2} = s')$$

$$\gamma[\sum_{a \in A} \pi(S_{t+2} = s, A_{t+2} = a) R(S_{t+2} = s, A_{t+2} = a) + \sum_{a \in A}$$

$$\pi(S_{t+2} = s, A_{t+2} = a) \sum_{s' \in S} P(S_{t+2} = s, A_{t+2} = a, S_{t+3} = s')$$

$$\gamma[E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+3} \mid S_{t+3} = s', \pi]]]$$

As seen in the previous step V(s) depends only on all actions taken
from state s and all states reached and actions taken thereafter and is independent of time.

$$v^{\pi}(s) \coloneqq \mathbf{E}\left[G_t | S_t = s, \pi\right]$$

$$= v^{\pi}(s) \coloneqq \mathbf{E}\left[G_0 | S_0 = s, \pi\right]$$

$$= \mathbf{E}[\sum_{t=0}^{\infty} \gamma^t R_t \mid S_0 = s, \pi] = \mathbf{E}[\mathrm{G} \mid S_0 = s, \pi]$$

(3)

2. (5 Points) Prove that the following two definitions of the action-value function are equivalent:

$$q^{\pi}(s,a) \coloneqq \mathbf{E}\left[G_t | S_t = s, A_t = a, \pi\right] \tag{4}$$

$$q^{\pi}(s,a) \coloneqq \mathbf{E}\left[G | S_0 = s, A_0 = a, \pi\right]. \tag{5}$$

$$\mathbf{E}[G \mid S_t = s, \pi, A_t = a]$$

$$= \mathbf{E}[\sum_{k=0}^{\infty} \gamma^k R_{t+k} \mid S_t = s, \pi, A_t = a]$$

$$= \mathbf{E}[\gamma^0 R_0 + \sum_{t=1}^{\infty} \gamma^t R_t \mid S_t = s, \pi, A_t = a]$$

$$= R(S_t = s, A_t = a) + \mathbf{E}[\sum_{k=1}^{\infty} \gamma^k R_{t+k} \mid S_t = s, \pi, A_t = a]$$

$$= R(S_t = s, A_t = a) + E[\gamma \sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k} \mid S_t = s, \pi, A_t = a]$$

$$= R(S_t = s, A_t = a) + E[\gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, \pi, A_t = a]$$

$$= R(S_t = s, A_t = a) + \sum_{a' \in A} \pi(S_{t+2} = s', A_{t+1} = a')$$

$$\sum_{s' \in S} P(S_t = s, A_t = a, S_{t+1} = s') \pi(S_{t+1} = s', A_{t+1} = a')$$

$$E[\gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a, S_{t+1} = s', \pi, A_{t+1} = a']$$

Following markov's property

$$= R(S_t = s, A_t = a) + \sum_{s' \in S} \sum_{a' \in A} \pi(S_{t+1} = s', A_{t+1} = a')$$

$$P(S_t = s, A_t = a, S_{t+1} = s')$$

$$\gamma E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_{t+1} = s', \pi, A_{t+1} = a']$$

This can be expanded further as

$$= R(S_t = s, A_t = a) + \sum_{s' \in S} \sum_{a' \in A} \pi(S_{t+1} = s', A_{t+1} = a') P(S_t = s, A_t = a, S_{t+1} = s')$$

$$\gamma[R(S_{t+1} = s, A_{t+1} = a) + \sum_{s' \in S} \sum_{a' \in A}$$

$$\pi(S_{t+2} = s', A_{t+2} = a') P(S_{t+1} = s, A_{t+1} = a, S_{t+2} = s')$$

$$\gamma[E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+2} \mid S_{t+2} = s', \pi, A_{t+2} = a']]$$

this can be further expanded as

$$= R(S_t = s, A_t = a) + \sum_{s' \in S} \sum_{a' \in A} \pi(S_{t+1} = s', A_{t+1} = a') P(S_t = s, A_t = a, S_t = s')$$

$$\gamma[R(S_{t+1} = s, A_{t+1} = a) +$$

$$\sum_{s' \in S} \sum_{a' \in A} \pi(S_{t+2} = s', A_{t+2} = a') P(S_{t+1} = s, A_{t+1} = a, S_{t+2} = s')$$

$$\gamma[R(S_{t+2} = s, A_{t+2} = a) + \sum_{s' \in S} \sum_{a' \in A}$$

$$\pi(S_{t+3} = s', A_{t+3} = a') P(S_{t+2} = s, A_{t+2} = a, S_{t+3} = s')$$

$$\gamma[E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+3} \mid S_{t+3} = s', \pi, A_{t+3} = a']]]$$

As seen in the previous step Q(s,a) depends only on all states reached from state s and all states reached and actions taken thereafter and is independent of time.

$$Q^{\pi}(s, a) \coloneqq \mathbf{E}\left[G_t | S_t = s, \pi, A_t = a\right]$$

$$= Q^{\pi}(s, a) \coloneqq \mathbf{E}\left[G_0 | S_0 = s, \pi, A_t = a\right]$$

$$= \mathbf{E}[\sum_{t=0}^{\infty} \gamma^t R_t \mid S_0 = s, \pi, A_0 = a] = \mathbf{E}[\mathrm{G} \mid S_0 = s, \pi, A_0 = a] \tag{6}$$

3. (5 Points) Let M and M' be two MDPs that are identical except for their initial state distributions. Prove that v^{π} is the same for both MDPs. You may write v^{π}_{M} and $v^{\pi}_{M'}$ to denote the state-value functions for π on M and M' respectively.

Let initial distribution of M and M' be d_0 and $d_{0'}$

and let the intial state for $M1$ selected stochastically

be $s1$ and that for M' be $s2$

$$\forall s \in S v^{\pi}(s) = \mathbf{E}[G \mid S_0 = s, \pi]$$

$$= \mathbf{E}[\sum_{k=0}^{\infty} \gamma^k R_k \mid S_0 = s, \pi]$$

$$= \mathbf{E}[\gamma^0 R_0 + \sum_{t=1}^{\infty} \gamma^t R_t \mid S_0 = s, \pi]$$

$$= \sum_{a \in A} \pi(s, a) R(S_0 = s, A_0 = a) + \mathbf{E}[\sum_{k=1}^{\infty} \gamma^k R_k \mid S_0 = s, \pi]$$

$$= \sum_{a \in A} \pi(s, a) R(S_0 = s, A_0 = a) + E[\gamma \sum_{k=1}^{\infty} \gamma^{k-1} R_k \mid S_0 = s, \pi]$$

$$= \sum_{a \in A} \pi(S_0 = s, A_0 = a) R(S_0 = s, A_0 = a) + E[\gamma \sum_{k=0}^{\infty} \gamma^k R_{k+1} \mid S_0 = s, \pi]$$

$$= \sum_{a \in A} \pi(S_0 = s, A_0 = a) R(S_0 = s, A_0 = a) + \sum_{a \in A} \pi(S_0 = s, A_0 = a)$$

$$\sum_{s' \in S} P(S_0 = s, A_0 = a, S_1 = s') E[\gamma \sum_{k=0}^{\infty} \gamma^k R_{k+1} \mid S_0 = s, A_0 = a, S_1 = s', \pi]$$

Following markov's property

$$= \sum_{a \in A} \pi(S_0 = s, A_0 = a) R(S_0 = s, A_0 = a) + \sum_{a \in A}$$

$$\pi(S_0 = s, A_0 = a) \sum_{s' \in S} P(S_0 = s, A_0 = a, S_1 = s')$$

$$\gamma E[\sum_{k=0}^{\infty} \gamma^k R_{k+1} \mid S_1 = s', \pi]$$

$$= \sum_{a \in A} \pi(S_0 = s, A_0 = a) R(S_0 = s, A_0 = a) + \gamma \sum_{a \in A}$$

$$\pi(S_0 = s, A_0 = a) \sum_{s' \in S} P(S_0 = s, A_0 = a, S_1 = s') v^(S_1 = s')$$

$$\forall s \in S$$

$$v_M^{\pi}(s) = \sum_{a \in A} \pi(S_0 = s, A_0 = a) R(S_0 = s, A_0 = a) + \gamma \sum_{a \in A}$$

$$\pi(S_0 = s, A_0 = a) \sum_{s' \in S} P(S_0 = s, A_0 = a, S_1 = s') v^(S_1 = s')$$

$$v_{M'}^{\pi}(s) = \sum_{a \in A} \pi(S_0 = s, A_0 = a) R(S_0 = s, A_0 = a) + \gamma \sum_{a \in A}$$

$$\pi(S_0 = s, A_0 = a) \sum_{s' \in S} P(S_0 = s, A_0 = a, S_1 = s') v^(S_1 = s')$$

As seen in the previous step V(s) depends only on all actions taken
from state s and corresponding transition probability
as well as all states reached, actions taken thereafter
and corresponding transition probabilities.
This expression is indepedent of the probability of initial state
and is defined by expected returns from state s
Since the two MDPs differ only in the initial distribution, they both will have the same function. (7)

4. (2 Points) Prove that multiplying all rewards (of a finite MDP with bounded rewards and $\gamma < 1$) by a positive scalar does not change which policies are optimal.

5. (2 Points) Prove that adding a positive constant to all rewards (of a finite MDP with bounded rewards and $\gamma < 1$) can change which policies are optimal.

Consider the 687 Gridworld MDP. If a positive constant is added S_{21} which is large than -10, the previous optimal policybefore the reward modification will no longer be an optimal policy as the robot will pass through the water state to get more rewards when in state 20. (8)

6. (1 Point) Your boss asked you to estimate the state-value function associated with a known policy, π, for a specific MDP. You misheard and instead estimated the action-value function. This estimation was very expensive, and so you do not want to do it again. Explain how you could easily retrieve the value of any state given what you have already computed.

$$V^\pi(s) \qquad = \qquad \sum \pi(s,a)q(s,a) \quad (9)$$

7. (5) Consider a finite MDP with bounded rewards, where all rewards are negative. That is, $R_t < 0$ always. Let $\gamma = 1$. The MDP is finite horizon, with horizon L, and also has a deterministic transition function and initial state distribution (rewards may be stochastic). Let $H_\infty = (S_0, A_0, R_0, S_1, A_1, R_1, \ldots, S_{L-1}, A_{L-1}, R_{L-1})$ be any history that can be generated by a deterministic policy, π. Prove that the sequence $v^\pi(S_0), v^\pi(S_1), \ldots, v^\pi(S_{L-1})$ is strictly increasing.

$\forall s \in S$

$$V(s) = \mathbf{E}[G \mid S_t = s, \pi]$$
$$= \mathbf{E}[\sum_{k=0}^{\infty} \gamma^k R_{t+k} \mid S_t = s, \pi]$$
$$= \mathbf{E}[\gamma^0 R_0 + \sum_{t=1}^{\infty} \gamma^t R_t \mid S_t = s, \pi]$$
$$= \sum_{a \in A} \pi(s,a) R(S_t = s, A_t = a) + \mathbf{E}[\sum_{k=1}^{\infty} \gamma^k R_{t+k} \mid S_t = s, \pi]$$
$$= \sum_{a \in A} \pi(s,a) R(S_t = s, A_t = a) + E[\gamma \sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k} \mid S_t = s, \pi]$$
$$= \sum_{a \in A} \pi(S_t = s, A_t = a) R(S_t = s, A_t = a) + E[\gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, \pi]$$
$$= \sum_{a \in A} \pi(S_t = s, A_t = a) R(S_t = s, A_t = a) + \sum_{a \in A} \pi(S_t = s, A_t = a)$$

$$\sum_{s' \in S} P(S_t = s, A_t = a, S_{t+1} = s') E[\gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a, S_{t+1} = s', \pi]$$

Following markov's property

$$= \sum_{a \in A} \pi(S_t = s, A_t = a) R(S_t = s, A_t = a) + \sum_{a \in A}$$

$$\pi(S_t = s, A_t = a) \sum_{s' \in S} P(S_t = s, A_t = a, S_{t+1} = s')$$

$$\gamma E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_{t+1} = s', \pi]$$

$$V(s) = \sum_{s \in S} \sum_{a \in A} \pi(S_t = s, A_t = a) P(S_t = s, A_t = a, S_{t+1} = s')[R(S_t = s, A_t = a) + \gamma V(s')]$$

This is the bellman equation for V(s)

For the given MDP, horizon is of length L and the policy is deterministic as well as the transition probability. So the next state from any state s is fixed.

For this MDP Value function can be written as:

$$V^{\pi}(s) = R(S_t = s, \pi(s)) + V(S_{t+1} = s')$$

where s' is the state reached from state s on following policy π

$$V^{\pi}(s_{\text{inf}}) = 0$$

In terminal state reward is 0.

The MDP transitions to terminal state from state S_{L-1}

$$V^{\pi}(s_{L-1}) = R(S_{L-1}, \pi(S_{L-1})) + V(S_{\text{inf}})$$

$$V^{\pi}(s_{L-2}) = R(S_{L-2}, \pi(S_{L-2})) + V(S_{L-1}) =$$

$$= R(S_{L-2}, \pi(S_{L-2})) + R(S_{L-1}, \pi(S_{L-1})) \leq V^{\pi}(s_{L-1})$$

since rewards are always negative in this MDP

Similarly $V^{\pi}(s_{L-3})$ is less than $V^{\pi}(s_{L-2})$

$$V^{\pi}(s_{L-3}) = R(S_{L-3}, \pi(S_{L-3})) + V(S_{L-2})$$

$$= R(S_{L-3}, \pi(S_{L-3})) + R(S_{L-2}, \pi(S_{L-2})) + R(S_{L-1}, \pi(S_{L-1}))$$

$$\leq V^{\pi}(s_{L-2})$$

$$V^{\pi}(S_t) < V^{\pi}(S_{t+1}) \forall t \in [0, ..L-1]$$

Hence $v^{\pi}(S_0), v^{\pi}(S_1), \ldots, v^{\pi}(S_{L-1})$ is strictly increasing

(10)

Part Two: Programming (55 Points Total)

Implement the 687-Gridworld domain described in class and in the class notes, and the cart-pole domain using the (frictionless) dynamics described by **?**, Equations 23 and 24. When implementing Cart-Pole, the state should include

the position of the cart, velocity of the cart, angle of the pole, and angular velocity of the pole. You **must** use a forward Euler approximation of the dynamics. If the cart hits the boundary of the track, terminate the episode. **You may not use existing RL code for this problem—you must implement the agent and environment entirely on your own and from scratch.** Four questions ask for a plot. You may report two plots: one for cart-pole and one for 687-Gridworld, where each of these two plots has a curve corresponding to the cross-entropy method and a curve corresponding to first-choice hill-climbing. Use the following values for the Cart-Pole constants:

- Fail angle = $\pi/2$. (If it exceeds this value or its negative, the episode ends in failure.)
- Motor Force F = 10.0 (force on cart in Newtons).
- Gravitational constant g is 9.8.
- Cart mass = 1.0.
- Pole mass = 0.1.
- Pole half-length $l = 0.5$.
- $\Delta t = 0.02$seconds (time step).
- Max time before end of episode = 20seconds + $10\Delta t$ = 20.2seconds.

Problems:

- (10 Points) Implement the cross-entropy method as described in the class notes and apply it to the 687-Gridworld. Use a tabular softmax policy. Search the space of hyperparameters for hyperparameters that work well. Report how you searched the hyperparameters, what hyperparameters you found worked best, and present a learning curve plot using these hyperparameters, as described in Figure **??**. This plot may be over any number of episodes, but should show convergence to a nearly optimal policy. The plot should average over at least 500 trials and should include standard error (or standard deviation) error bars (say which error bar variant you used).
  - Hyperparameters used gridworld Cross Entropy:
  - $sigma = 1, exploration\_parameter = 10.5, eta = 0.6,$
  - $No\_of\_episodes\_per\_policy = 80,$
  - $No\_of\_iterations = 400,$
  - $size\_of\_elite\_population = 5, Size\_of\_population40$
  - Used standard deviation to plot error bars
  - mean was randomly initialised with random numbers in the range $[0, 1]$

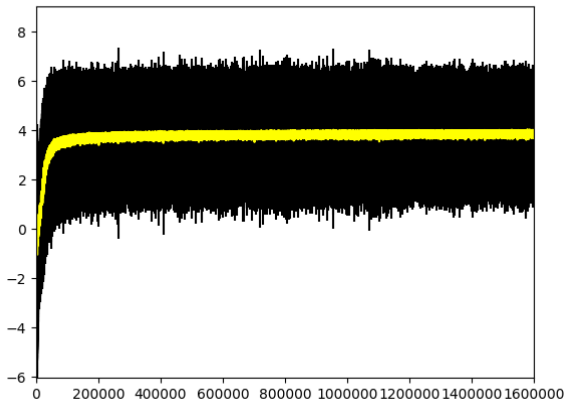

Figure 1: gridworld ce

- (10 Points) Repeat the previous question, but using the cross-entropy method on the cart-pole domain. Notice that the state is not discrete, and so you cannot directly apply a tabular softmax policy. It is up to you to create a representation for the policy for this problem. Report the same quantities, as well as how you parameterized the policy.
  - Policy was parameterized using the formula $X = a * (Distance) + b * (Velocity) + c * (Angle) + d * (Angular\_velocity)$ since the states are continuous.
  - a,b,c,d are policy parameters. when $X \geq 0$, force would be +10 and when $x \leq 0$ force would be -10.
  - Hyperparameters used gridworld Cross Entropy:
  - $sigma = 1, exploration\_parameter = 15.0, eta = 0.4,$
  - $No\_of\_episodes\_per\_policy = 30,$
  - $No\_of\_iterations = 30,$
  - $size\_of\_elite\_population = 20, Size\_of\_population = 10$
  - mean was randomly initialised with random numbers in the range $[1, 20]$
  - Used standard deviation to plot error bars
- (10 Points) Repeat the previous question, but using first-choice hill-climbing on the 687-Gridworld domain. Report the same quantities.
  - Hyperparameters used gridworld Cross Entropy:
  - $sigma = 5.0, exploration\_parameter = 0.2,$

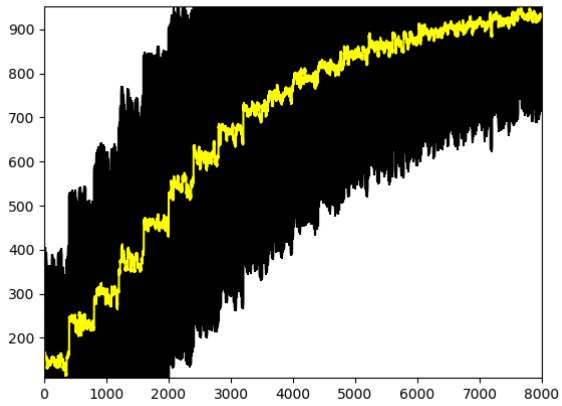

Figure 2: Cartpole ce

  - $No\_of\_episodes\_per\_policy = 80$,
  - $No\_of\_iterations = 12000$,
  - mean was randomly initialised with random numbers in the range $[0, 1]$
  - Used standard deviation to plot error bars

- (10 Points) Repeat the previous question, but using first-choice hill-climbing (as described earlier in these notes) on the cart-pole domain. Report the same quantities and how the policy was parameterized.

  - Hyperparameters used gridworld Cross Entropy:
  - $exploration\_parameter = 0.6$
  - $No\_of\_episodes\_per\_policy = 60$,
  - $No\_of\_iterations = 200$,
  - mean was randomly initialised with random numbers in the range $[1, 20]$
  - Used standard deviation to plot error bars

- (5 Points) Reflect on this problem. Was it easier or harder than you expected to get these methods working? In the previous assignment you hypothesized how long it would take an agent to solve the 687-Gridworld problem. Did it take more or fewer episodes than you expected? Why do you think this happened?

  - It was harder than I expected since finding the right hyperparameters required us to run the program several time and try all possible ranges

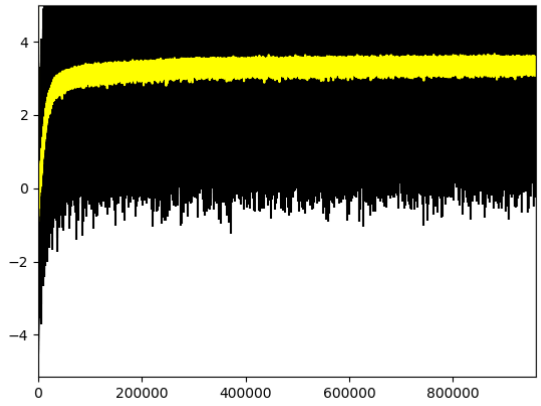

Figure 3: gridworld fchc

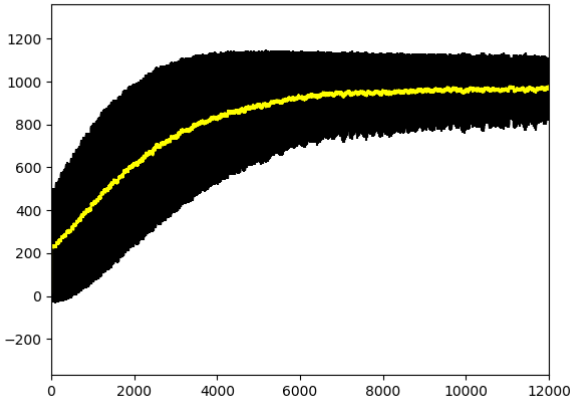

Figure 4: Cartpole fchc

of real values for above mentioned hyperparameters and each trial took 2-5 minutes to run. In the previous assignment I assumed it would take lakhs of episodes but it took

- around 1.5 lakh episodes to reach the optimal policy. This was expected since the
- probability of transition as well as action selection is stochastic and hence robot
-
- would end up exploring a large number of possible paths.

- To run cross entropy for gridworld run $->$ python $gridworld.py$ no_of_trials
- To run cross entropy for cartpole run $->$ python $final\_cartpole.py$ no_of_trials
- To run fchc for cartpole run $->$ python $cartpole\_fc.py$ no_of_trials
- To run fchc for gridworld run $->$ python $gridworld\_fc\_revised.py$ no_of_trials
- To generate plot run $->$ python $get\_mean\_std.py$ $output\_file$