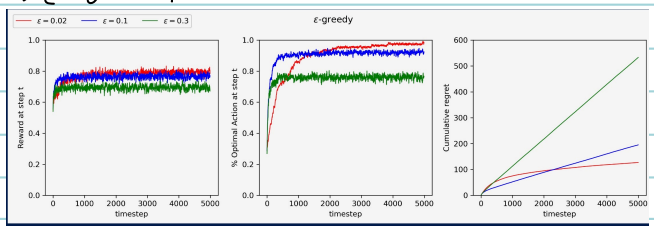
(Q1)

b.

i) ε-greedy:

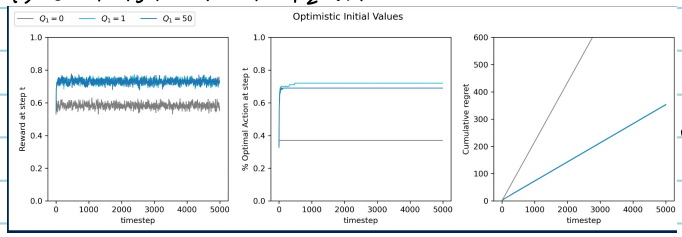


For ε = 0.02, for the initial speed, it is the slowest for reward and %-optimal due to only 2% of the time being pure exploration. For convergence speed, it takes about 1,000 steps to plateau, and for asymptotic performance, the highest eventual reward was around 0.95, and the lowest regret was about 125 at T = 5000.

For ε = 0.1, the initial speed has the fastest jump due to the 10% exploration. For convergence speed, it reaches its plateau around 500 steps. For asymptotic performance, it had a good final reward around 0.92, with a moderate regret around 200.

For ε = 0.3, it has moderate rise, but it is very noisy. The 30% exploration random pulls slow down consistent improvement. For convergence speed, it plateaus early and at a lower level, around 0.75(reward). For Asymptotic performance, it has the worst of the 3, with a regret of about 550.

ii) optimistic initialization:

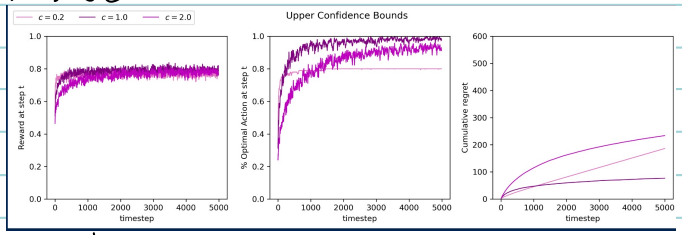


For $Q_0 = 0$, there is no built-in exploration, so the first arm looks the best, the tier are broken randomly, and it doesn't look at the other arms. The reward is about 55, the %-optimal is about 0.37, and the linear regret is about 600.

For $Q_0 = 1$, the mild optimism forces each arm to be tried once, and then it settles on the best. The reward is about 0.75, the %-optimal is about 0.71, and the regret is about 350.

For $Q_0 = 50$, the strong optimism exxaperates the initial exploration, but the estimates, and it behaves similar to $Q_0 = 1$. The reward is about 0.75, the %-optimal is about 0.69, and the regret is about 355.

iii) UCB:



For c = 0.2, the exploration bonus is too small, which leads to it not exploring enough, so it stays with sub-optimal arms. The reward is about 0.79, the %-optimal is about 0.8, and the regret is about 185.

For c = 1.0, it has a well-calibrated bonus, so it quickly differentiates arms and exploits. The reward is about 0.99, the %-optimal is about 0.98, and the regret is about 70.

For c = 2.0, it over-explores early on, which leads to a delaying convergence, but it eventually learns

C.

i. The best overall algorithm would be the UCB algorithm with c=1.0 because it achieves the highest average reward, fastest rise in %-optimal action, and the lowest

cumulative regret. The worst overall would be the naive optimistic-initialization with $Q_0 = 0$ b/c it never really explores beyond the first draw, so it settles on a suboptimal arm. I has a low reward (0.55) and high regret (200). I would use UCB with $c = 1.0$ b/c it needs no hand-tuned ϵ or initial bias, and it has a good confidence-bonus formula

ii. It would be UCB with $c = 1.0$

Q2.

a) $(y_{s,3} - \hat{y}_{s,3})^2 = (2-4)^2 = \boxed{4}$

b)

i. $\min_{U^{(s)}} \frac{1}{2} \sum_{i \in D_s} (y_{s,i} - U^{(s)} \cdot V^{(i)})^2 + \frac{\lambda}{2} \|U^{(s)}\|^2$

ii. $U^{(s)} = \frac{\sum_{i \in D_s} y_{s,i} V^{(i)}}{\sum_{i \in D_s} (V^{(i)})^2 + \lambda} = \frac{5.2 + 2.1}{2^2 + 1^2 + 1} = \frac{10+2}{4+1+1} = \frac{12}{6} = \boxed{2}$

c) $\hat{y}_{s,3} = U^{(s)} \cdot V^{(3)} = 2.1 = \boxed{2}$

Q3.

a) You should make sure to break ties b/c if you do it randomly, you may get different final clusters on every run. Also, it can prevent clean convergence b/c k-means relies on each assignment + update step never increasing the sum of squared distances. Finally, with a fixed rule for ties, you know exactly when no point changes cluster and the centroids stabalize.

b)

i. k-means converges in 2 iterations

ii.

cluster	final centroid	points	
C_1:	(3.50, 2.75)	(2,4), (2,3), (3,3), (4,3), (5,3), (3,2), (4,2), (5,2)	left-mouth
C_2:	(7.00, 2.83)	(8,4), (6,3), (7,3), (8,3), (6,2), (7,2)	right-mouth
C_3:	(5.00, 6.50)	(3,7), (4,7), (3,6), (4,6), (6,7), (7,7), (6,6), (7,6)	eyes

c) $z^{(1)} = (5,3)$, $z^{(2)} = (3,6)$, $z^{(3)} = (7,6)$

- $z^{(1)}$ would attract all the mouth points (left and right)
- $z^{(2)}$ would attract the left eye point
- $z^{(3)}$ would attract the right eye point