Learning Dynamics: Assignment 3
Multi-Armed Bandits

Hakim Boulahya

Université Libre de Bruxelles
hboulahy@ulb.ac.be - 000391737

December 18, 2017

Contents

1 N-Armed Bandit

1.1 Exercice 1

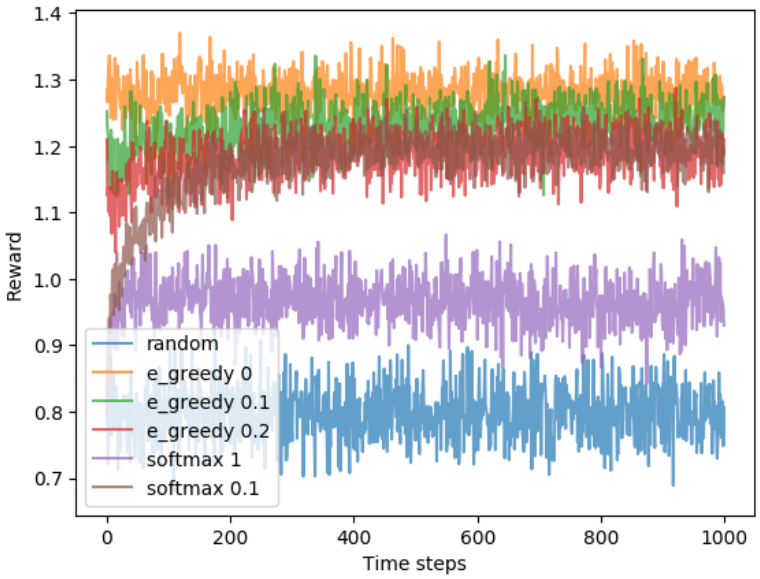
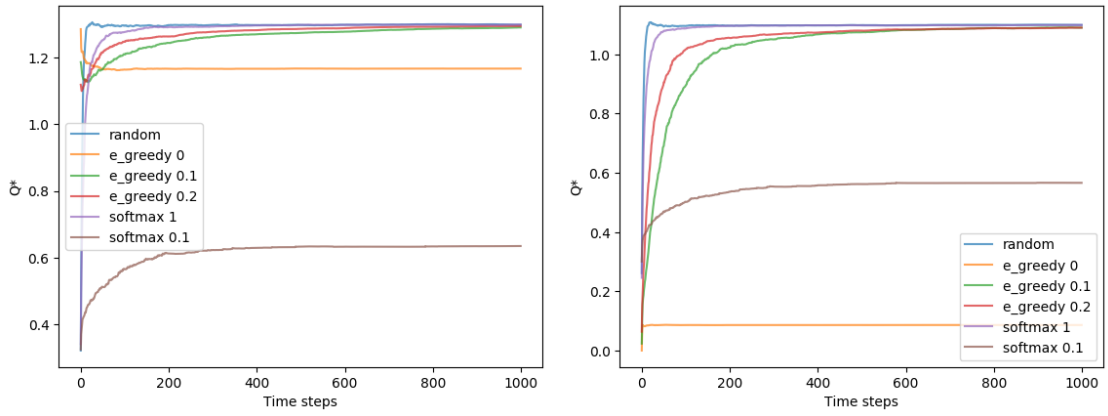
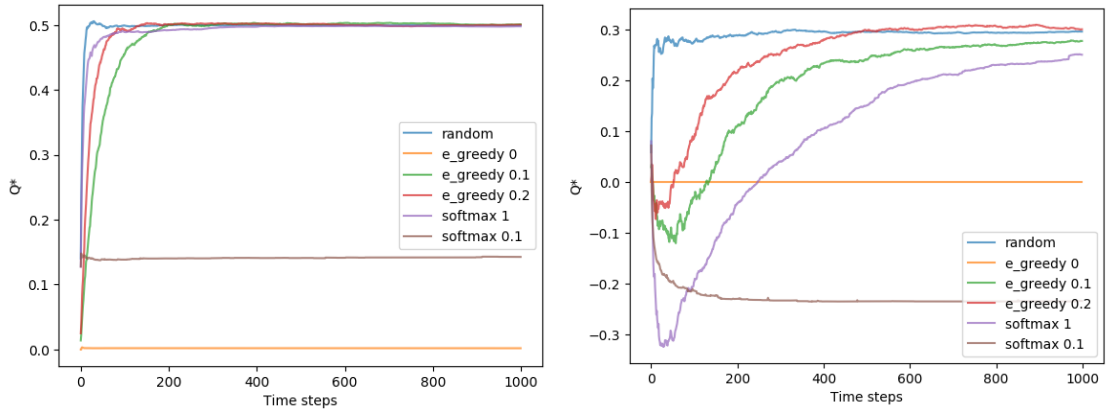


Figure 1: Average rewards for all algorithms

We can see that only exploring *i.e.* doesn't perform well, and the reward stay low. The boltzmann distribution of softmax with a temparature of 1 also has a poor performance compared to the other methods. Using softmax with a temparature of 0.1 gives better results. Using e-greedy with small value, i.e. will likely choose the action with the best optimum, tends to better results. egreedy with 0 seems to arrives *directly* to the best arm. We can see on the graph that when ϵ is bigger it usually takes more time to get to the best arm. Softmax with a temperature of 0.1 does tends to the best arm, but takes more time.

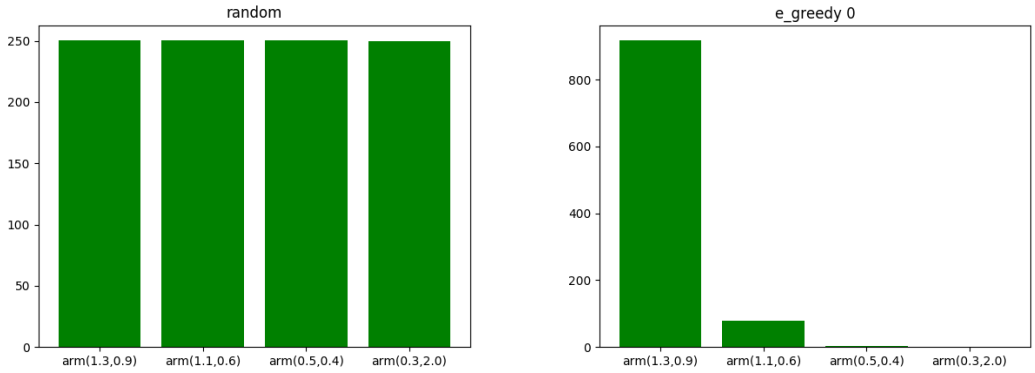


(a) $Q_{a_0}(t)$ (b) $Q_{a_1}(t)$



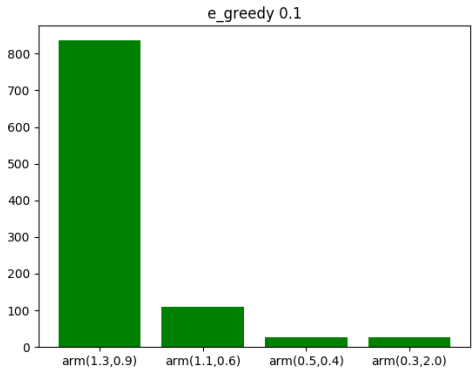
(c) $Q_{a_2}(t)$ (d) $Q_{a_3}(t)$

Figure 2: Plot per arm showing the $Q^*_{a_i}$ of that action along with the actual Q_{a_i} a i estimate over time with μ = (1.3, 1.1, 0.5, 0.3), σ = (0.9, 0.6, 0.4, 2.0)

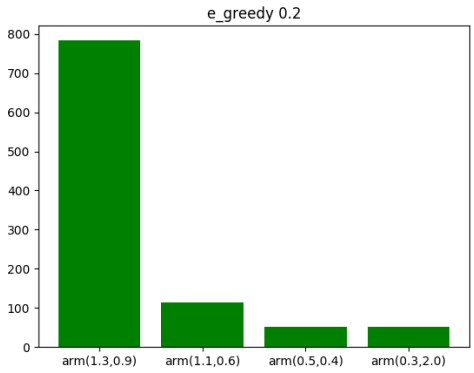


(a)

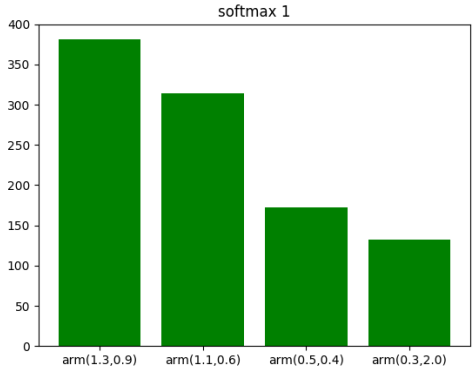
(b)



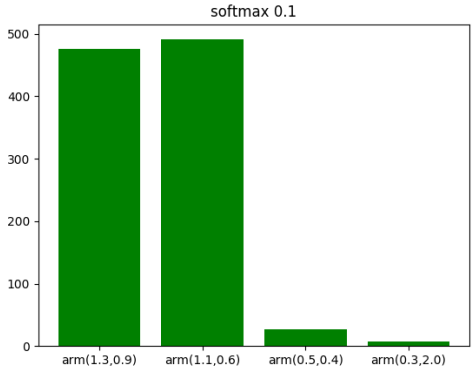
(c)



(d)



(e)



(f)

Figure 3: Histograms showing the number of times each action is selected per selection strategy with μ = (1.3, 1.1, 0.5, 0.3), σ = (0.9, 0.6, 0.4, 2.0)

1.2 Exercice 2

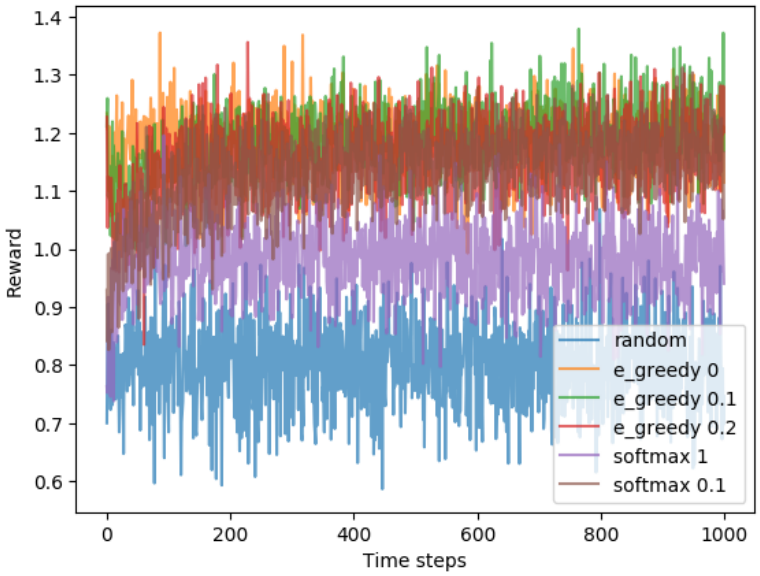
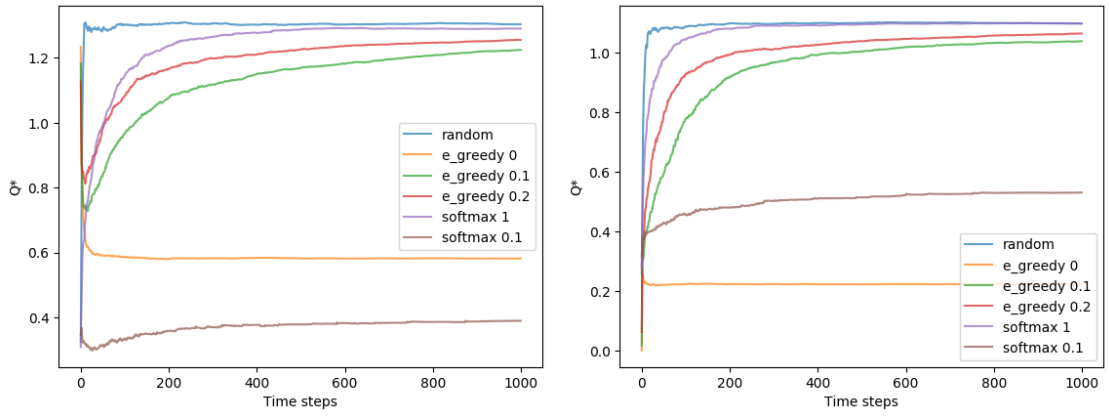


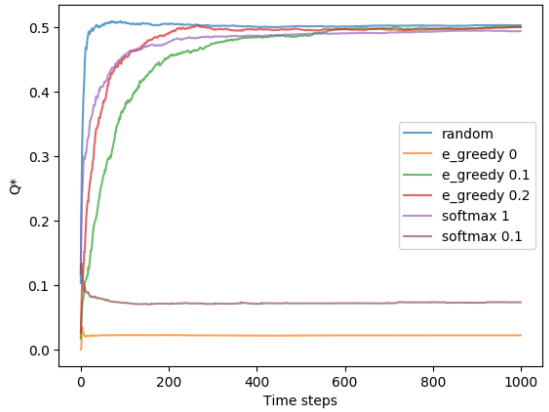
Figure 4: Average rewards for all algorithms

The results order of the algorithm that perform best doesn't really changes. By doubling the standard deviation, we only allowed the algorithms to provide a more random rewards *i.e.* a bigger range. We can see that the dynamics of the algorithms don't change.

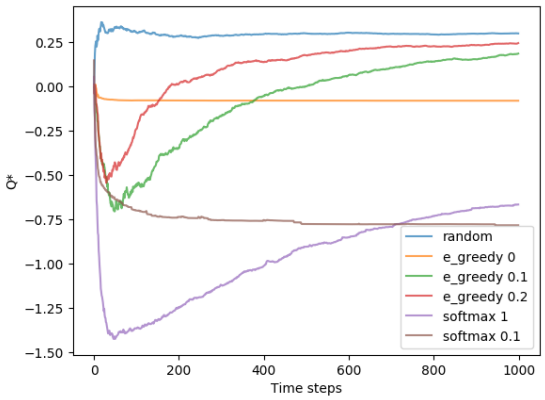


(a) $Q_{a_0}(t)$

(b) $Q_{a_1}(t)$

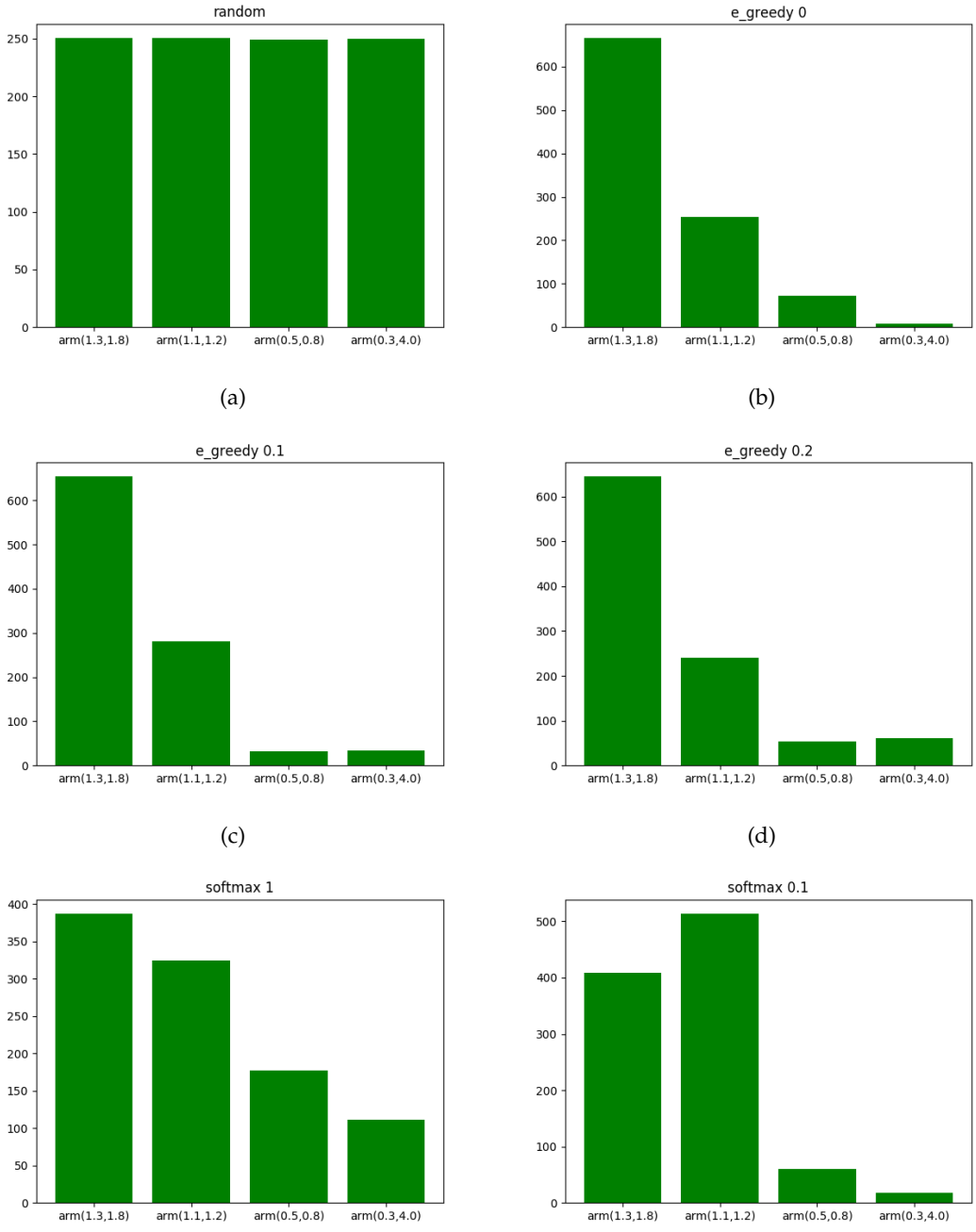


(c) $Q_{a_2}(t)$



(d) $Q_{a_3}(t)$

Figure 5: Plot per arm showing the $Q^*_{a_i}$ of that action along with the actual Q_{a_i} a i estimate over time with μ = (1.3, 1.1, 0.5, 0.3), σ = (1.8, 1.2, 0.8, 4.0)



(a) (b)

(c) (d)

(e) (f)

Figure 6: Histograms showing the number of times each action is selected per selection strategy with μ = (1.3, 1.1, 0.5, 0.3), σ = (1.8, 1.2, 0.8, 4.0)

1.3 Exercice 3

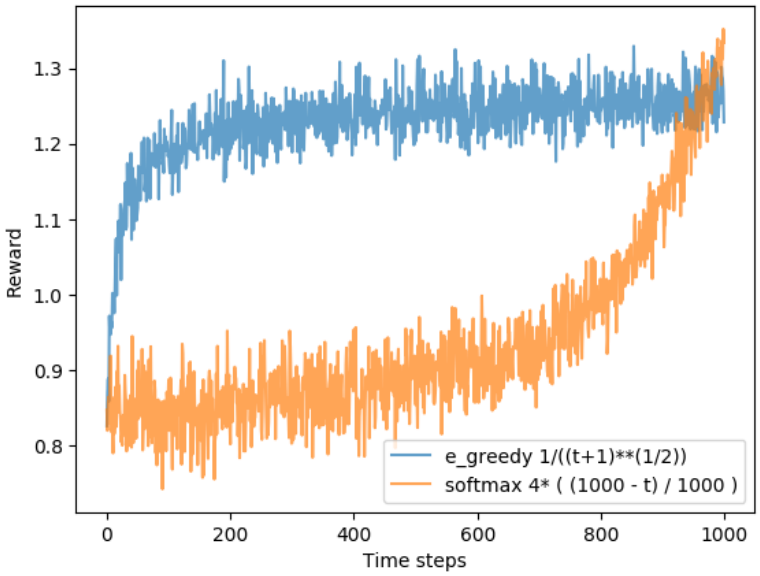
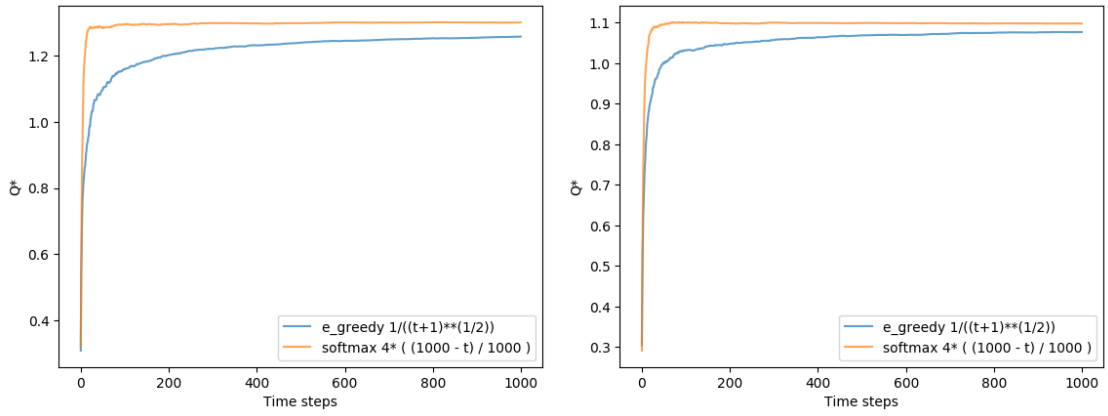


Figure 7: Average rewards for all algorithms

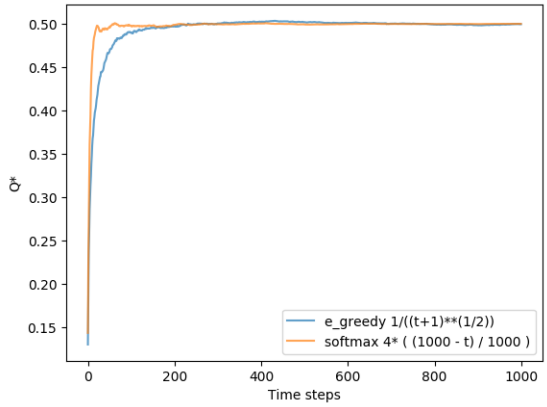
Dynamic egreedy doesn't seems to perform better that the statics one, the resulting rewards tends to be identical. Observing the graph shows us that it takes more time to stabilize to the best arm.

Dynamic softmax seems to be better, if you consider to long term reward. It takes to the 1000 epoch for the algorithm to find the best arm in opposition to the softmax using a a temperature of 0.1, where it is close to the best arm at around epoch 200, but not as close as the dynamic softmax, where it is closer to the best arm, when it is stabilized.

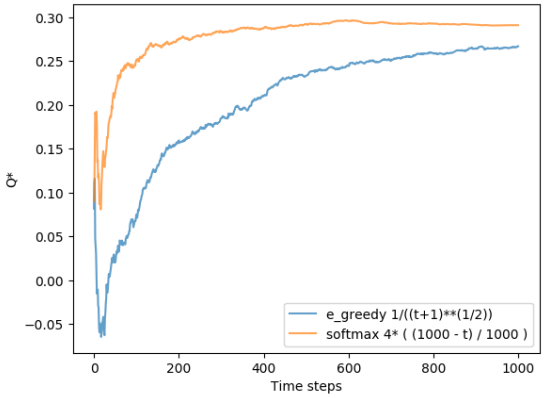


(a) $Q_{a_0}(t)$

(b) $Q_{a_1}(t)$

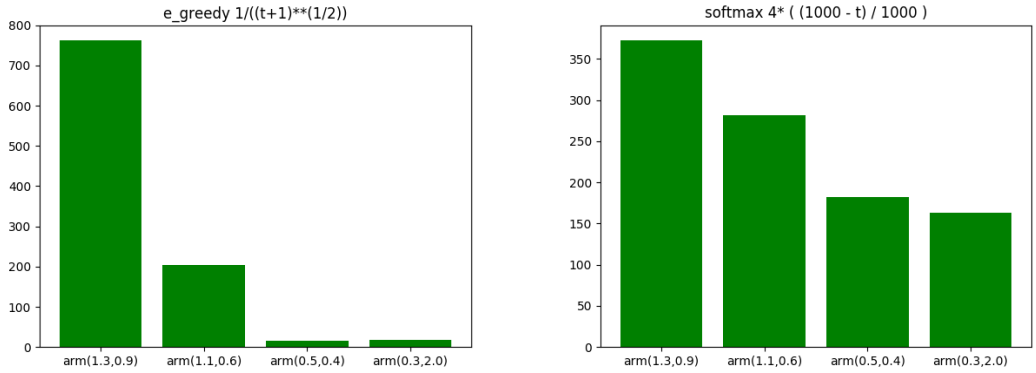


(c) $Q_{a_2}(t)$



(d) $Q_{a_3}(t)$

Figure 8: Plot per arm showing the $Q^*_{a_i}$ of that action along with the actual Q_{a_i} a i estimate over time



(a) (b)

Figure 9: Histograms showing the number of times each action is selected per selection strategy

2 Climbing game

When using the formula, it is important to have a big tau, because exp(Q/tau) will give errors if Q is too big. (See how EV(a) is calculated)

I also remarked that when using EV with $\max_r ewards or max_q when min_t au = 0.001 the 11 is good, but when 0.1 or 1, not.$