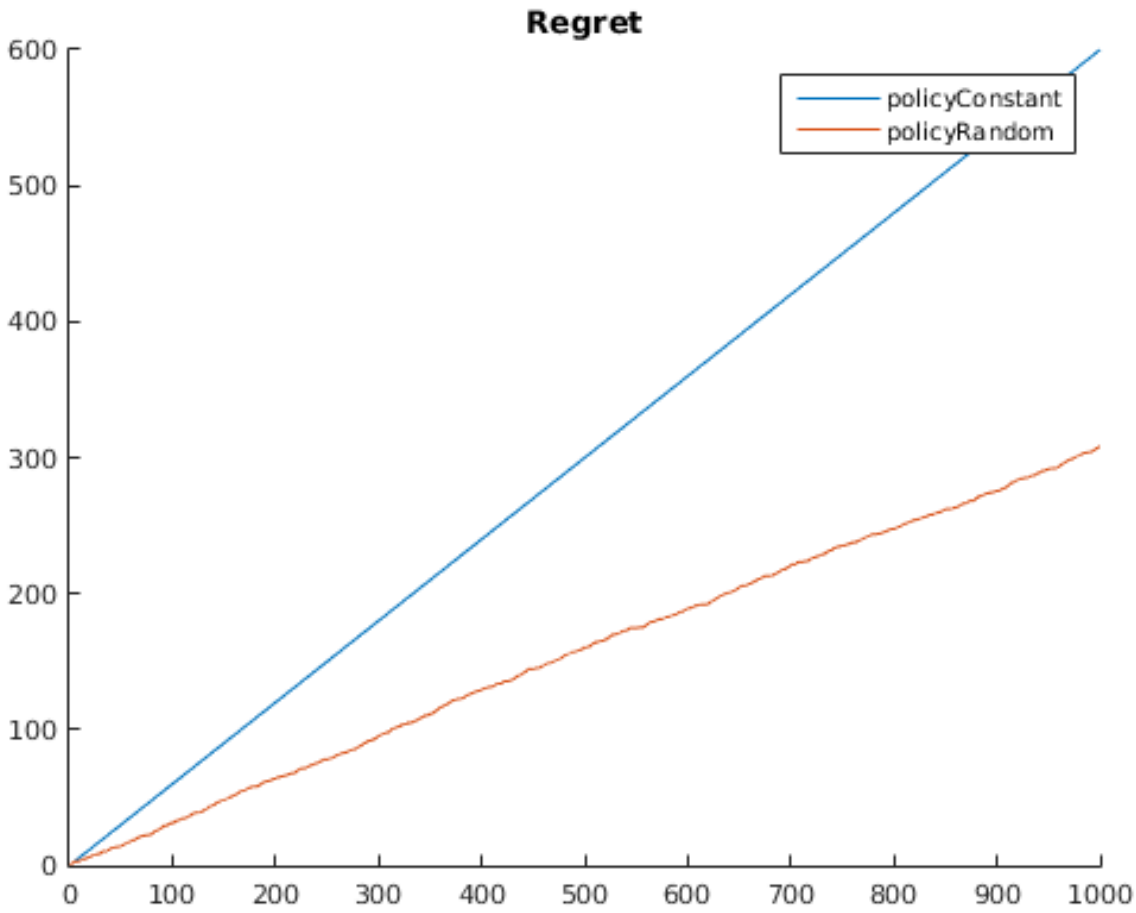
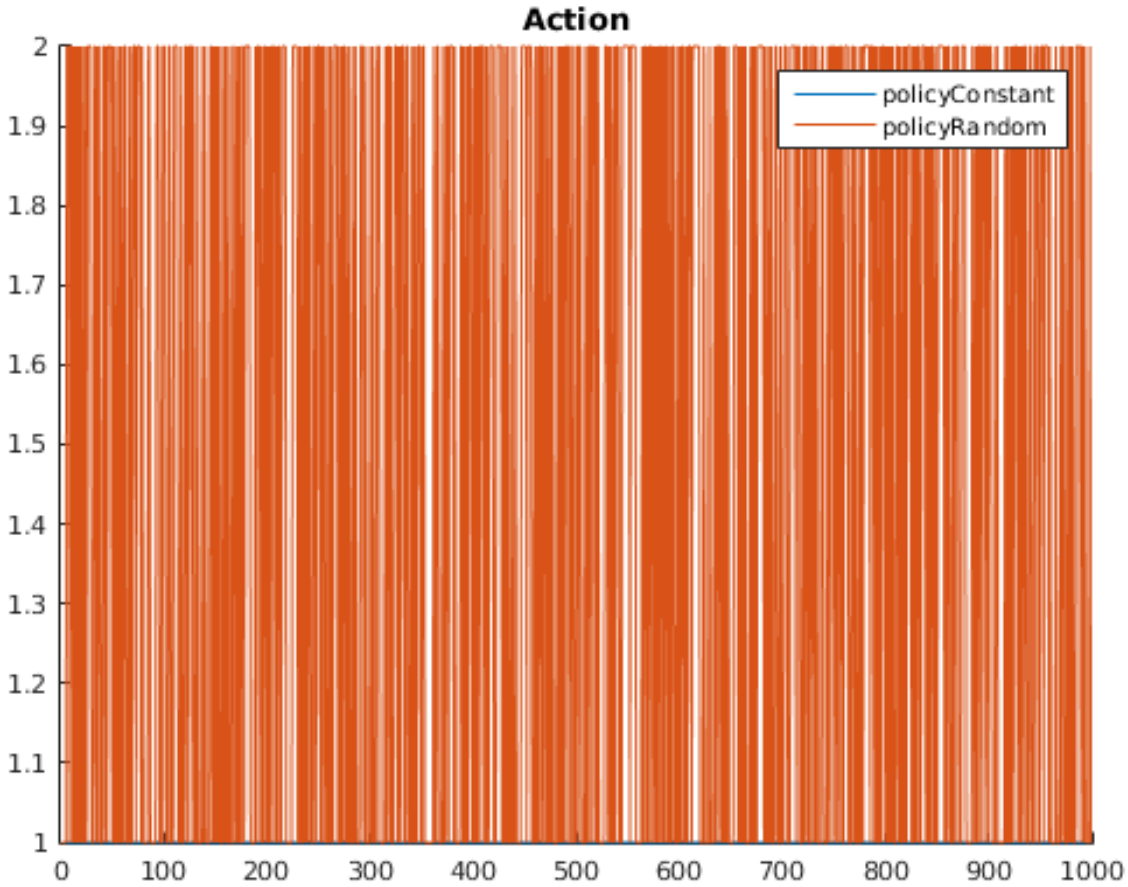
Ben Holden

Robotics 831, Homework #2

10/19/2015

1) Introduction
2) Framework
2.3.1) Fill in function cumulativeRewardBestActionHindsight
See code folder
2.5) policyConstant and policyRandom on gameConstant



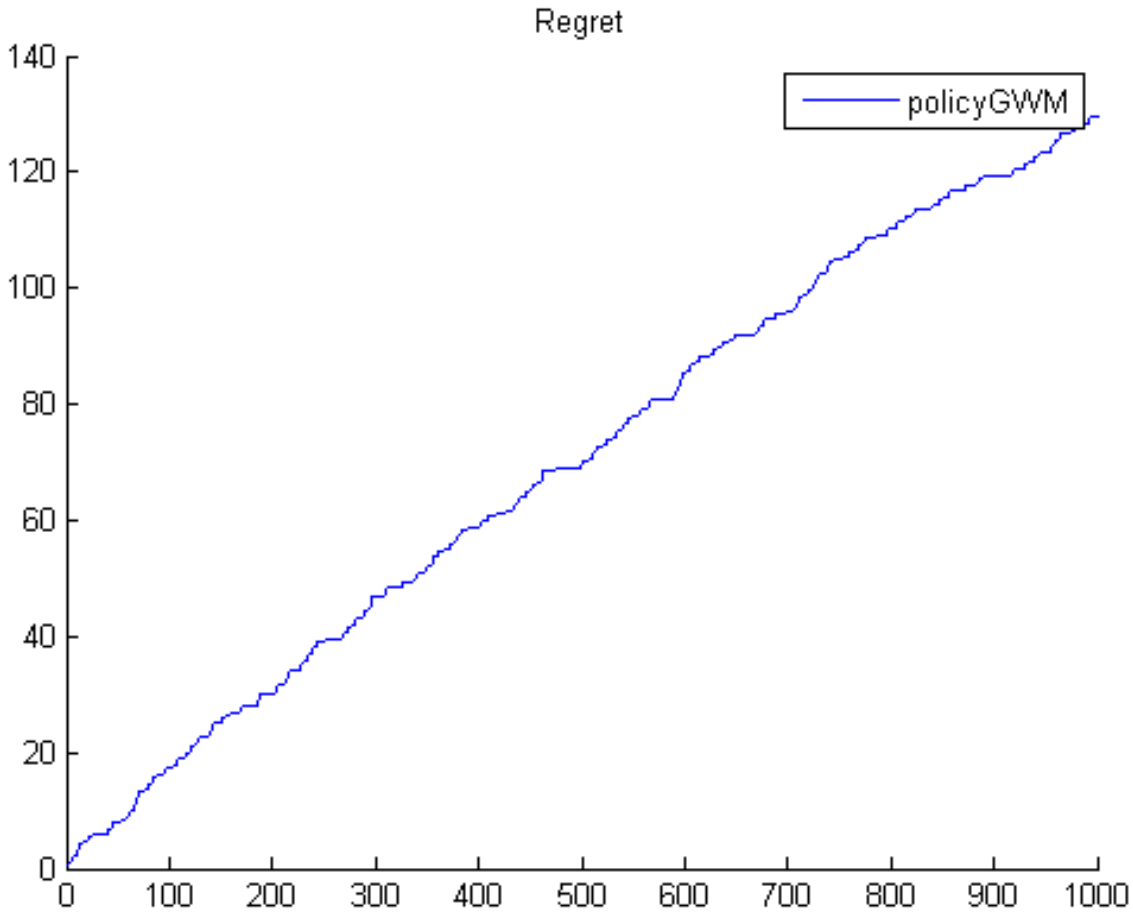


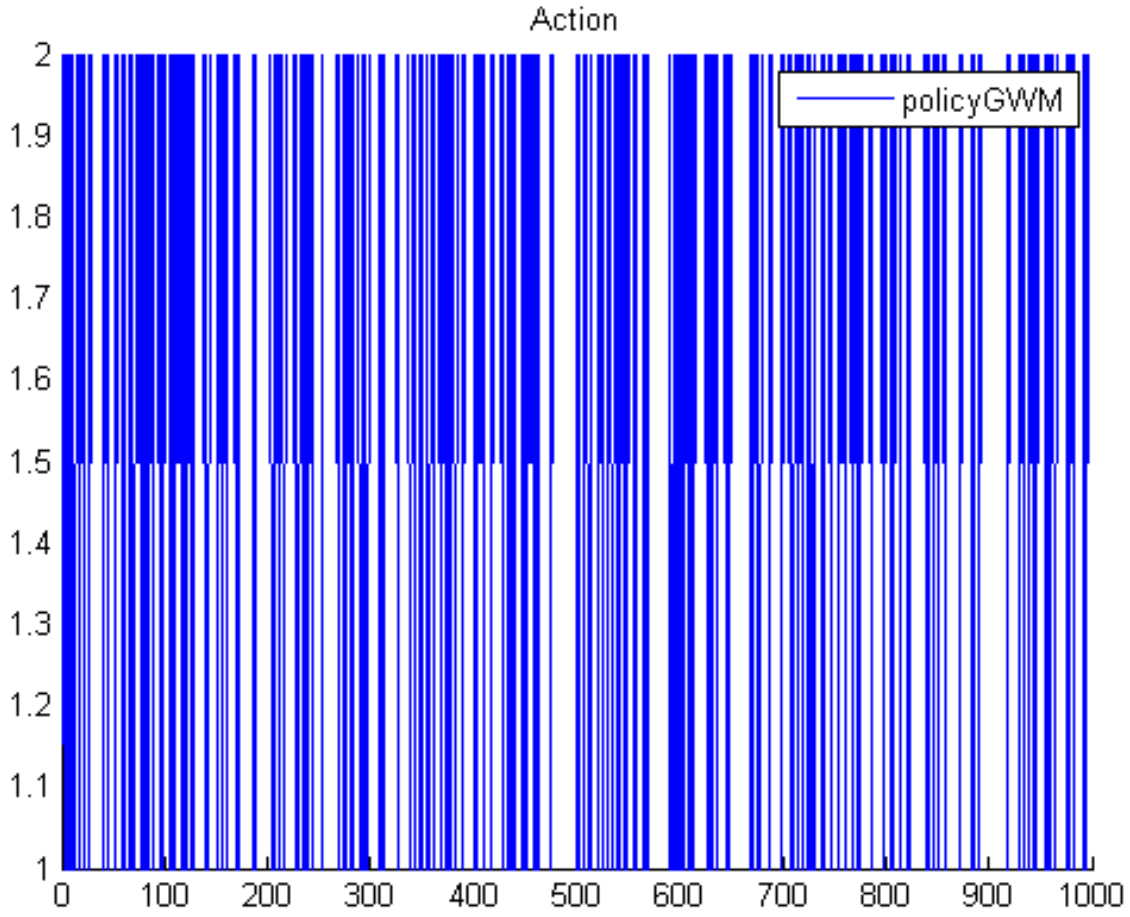
Random picks each action with an even distribution and Constant picks one and stays with it. So constant has a regret of 600 or 200 depending on the choice. Random has a regret of approximately the average of those two, or 300. These graphs support this.

3) EXP3

3.1.1) policyGWM on gameConstant

See code for implementation


Regret
policyGWVM
140
120
100
80
60
40
20
0
0
100
200
300
400
500
600
700
800
900
1000



GWM is no longer no-regret! This is because full information is not given to the policy only the loss of the selected action. This means that every action not selected is treated as if it had 0 loss even when it could have led to a loss of 1.

3.1.2) Derive the regret bound for GWM in the bandit setting

Potential Function:

$$\Phi^{(t+1)} = \log\left(\sum_{n=1}^{N} w_n^{t+1}\right)$$

Upper Bound:

$$\Phi^{(t+1)} = \log\left(\sum_{n=1}^{N} w_n^t e^{-\eta l_n^t}\right)$$

$$\Phi^{(t+1)} = \log\left(\sum_{i=1}^{N} w_i^t \sum_{n=1}^{N} \frac{w_n^t}{\sum_{i=1}^{N} w_i^t} e^{-\eta l_n^t}\right)$$

$$\Phi^{(t+1)} = \log\left(\sum_{i=1}^{N} w_i^t\right) + \log\left(\sum_{n=1}^{N} \frac{w_n^t}{\sum_{i=1}^{N} w_i^t} e^{-\eta l_n^t}\right)$$

$$\Phi^{(t+1)} = \Phi^t + \log\left(\sum_{n=1}^{N} p_n^t e^{-\eta l_n^t}\right)$$

Using $e^{-\eta x} \leq 1 + (e^{-\eta} - 1)\, x,\ x \in [0,1]$

$$\Phi^{(t+1)} \leq \Phi^t + \log\left(\sum_{n=1}^{N} p_n^t \left(1 + \left(e^{-\eta} - 1\right) l_n^t\right)\right)$$

$$\Phi^{(t+1)} \leq \Phi^t + \log\left(\sum_{n=1}^{N} p_n^t + \sum_{n=1}^{N} \left(e^{-\eta} - 1\right) l_n^t p_n^t\right)$$

$$\Phi^{(t+1)} \leq \Phi^t + \log\left(1 + \sum_{n=1}^{N} \left(e^{-\eta} - 1\right) l_n^t p_n^t\right)$$

Using $\log(1 - x) \leq -x$

$$\Phi^{(t+1)} \leq \Phi^t - \left(1 - e^{-\eta}\right) \sum_{n=1}^{N} l_n^t p_n^t$$

Telescoping:

$$\Phi^{T+1} \leq \Phi^1 - \left(1 - e^{-\eta}\right) \sum_{t=1}^{T} \sum_{n=1}^{N} l_n^t p_n^t$$

$$\Phi^{T+1} \leq \log N - \left(1 - e^{-\eta}\right) \sum_{t=1}^{T} \mathbb{E}_{a^t \sim p_n^t} l_{a^t}^t$$

Lower Bound:

$$\Phi^{T+1} = \log\left(\sum_{n=1}^{N} w_n^{t+1}\right)$$

$$\Phi^{T+1} \geq \log\left(w_{n^*}^{t+1}\right)$$

$$\Phi^{T+1} \geq \log\left(e^{-\eta \sum_{t=1}^{T} l_{n^*}^t}\right)$$

$$\Phi^{T+1} \geq -\eta \sum_{t=1}^{T} l_{n^*}^t$$

Combine:

$$-\eta \sum_{t=1}^{T} l_{n^*}^t \leq \log N - \left(1 - e^{-\eta}\right) \sum_{t=1}^{T} \mathbb{E}_{a^t \sim p_n^t} l_{a^t}^t$$

$$\sum_{t=1}^{T} \mathbb{E}_{a^t \sim p_n^t} l_{a^t}^t \leq \frac{\eta}{1 - e^{-\eta}} \sum_{t=1}^{T} l_{n^*}^t + \frac{\log N}{1 - e^{-\eta}}$$

Using $\frac{1}{1-e^{-\eta}} \leq \frac{1+\eta}{\eta}$

$$\sum_{t=1}^{T} \mathbb{E}_{a^t \sim p_n^t} l_{a^t}^t \leq (1 + \eta) \sum_{t=1}^{T} l_{n^*}^t + \frac{2 \log N}{\eta}$$

Regret:

$$R^T = \sum_{t=1}^{T} \mathbb{E}_{a^t \sim p_n^t} l_{a^t}^t - \min_{i=1,\dots,N} \sum_{t=1}^{T} l_i^t$$

$$R^T = (1+\eta) \sum_{t=1}^{T} l_{n^*}^t + \frac{2 \log N}{\eta} - \sum_{t=1}^{T} l_{n^*}^t$$

$$R^T = \eta \sum_{t=1}^{T} l_{n^*}^t + \frac{2 \log N}{\eta}$$

$$R^T \leq \frac{2 \log N}{\eta} + \eta \sum_{t=1}^{T} \sum_{n=1}^{N} l_{n^*}^t p_{n^*}^t$$

$$R^T \leq \frac{2 \log N}{\eta} + \eta \sum_{t=1}^{T} \sum_{n=1}^{N} l_{n^*}^t p_{n^*}^t$$

$$\mathbb{E}\left[R^T\right] \in O\left(\eta T^2 + \frac{1}{\eta} \log N\right)$$

So there is no magnitude of η, which makes this less the $O(T)$ so no-regret is not possible. In the code $\eta = O\left(\sqrt{\frac{\log N}{t}}\right)$ is used giving:

$$\mathbb{E}_{bandit}\left[R^T\right] = O\left(T\sqrt{NT} + \sqrt{T}\right)$$

So GWM will not work in the bandit setting.

3.1.3) Show that $\sum_{t=1}^{T} \tilde{l_n^t}$ is an unbiased estimator of $\sum_{t=1}^{T} l_n^t$ for any action n

Prove

$$E\left[\sum_{t=1}^{T} \tilde{l_n^t}\right] = E\left[\sum_{t=1}^{T} l_n^t\right]$$

Linear expectation can distribute the expectation

$$E\left[\sum_{t=1}^{T} \tilde{l_n^t}\right] = \sum_{t=1}^{T} E\left[\tilde{l_n^t}\right]$$

Using the law of total expectation

$$\sum_{t=1}^{T} E\left[\tilde{l_n^t}\right] = \sum_{t=1}^{T} E\left[E\left[\tilde{l_n^t} | p_n^t\right]\right]$$

By definition of $\tilde{l_n^t} = \frac{l_n^t}{p_n^t}$

$$E\left[\tilde{l_n^t} | p_n^t\right] = p_n^t \frac{l_n^t}{p_n^t} = l_n^t$$

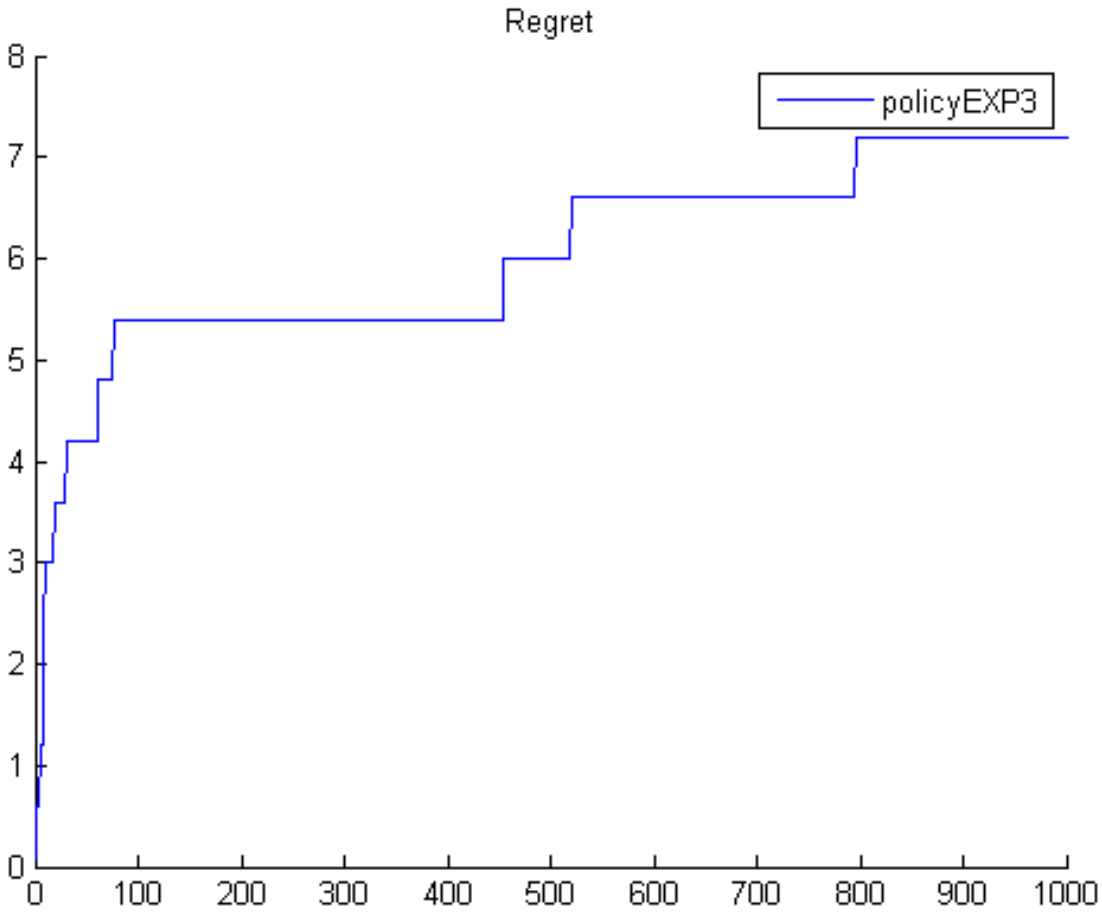
Plugging in the result from above

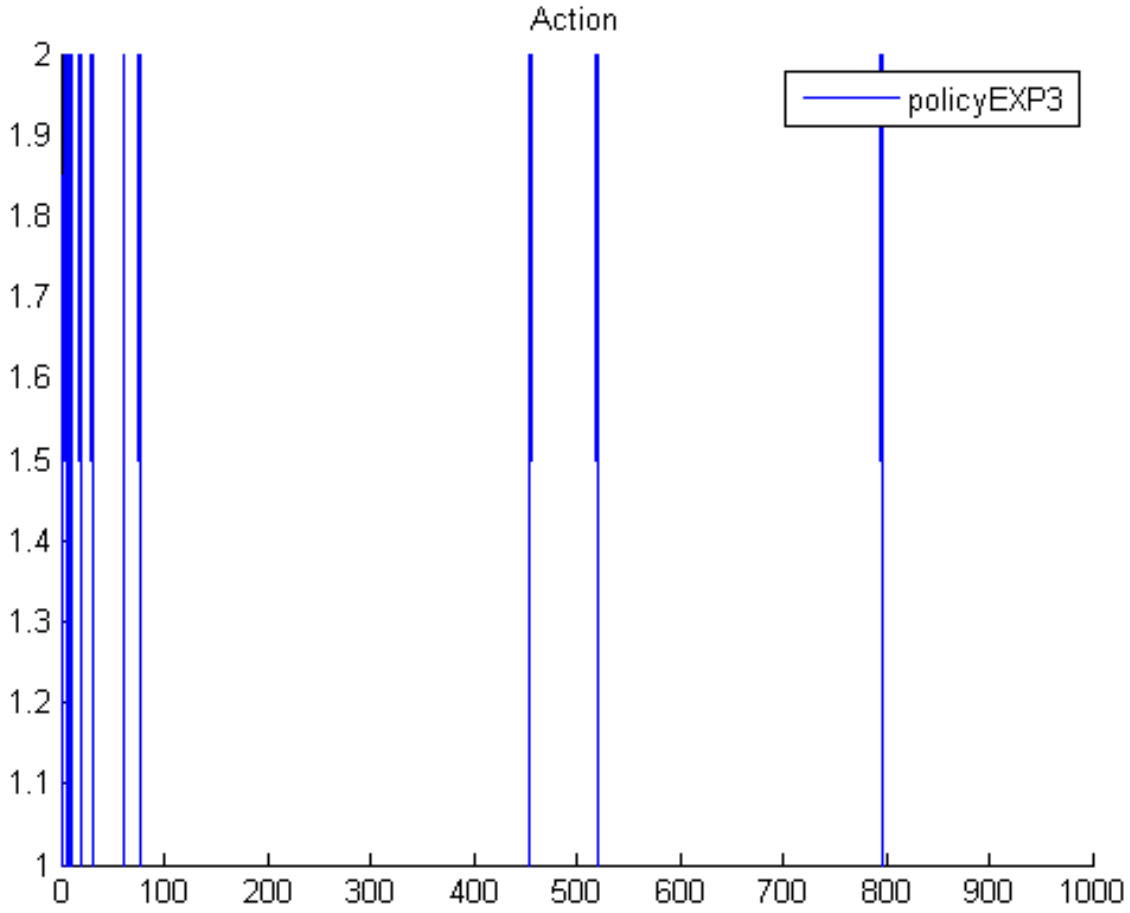
$$\sum_{t=1}^{T} E\left[E\left[\tilde{l_n^t} | p_n^t\right]\right] = \sum_{t=1}^{T} E\left[l_n^t\right]$$

Redistributing the expectation proves the equality

$$\sum_{t=1}^{T} E\left[l_n^t\right] = E\left[\sum_{t=1}^{T} l_n^t\right]$$

3.2.1) policyEXP3 on gameConstant
See code for implementation



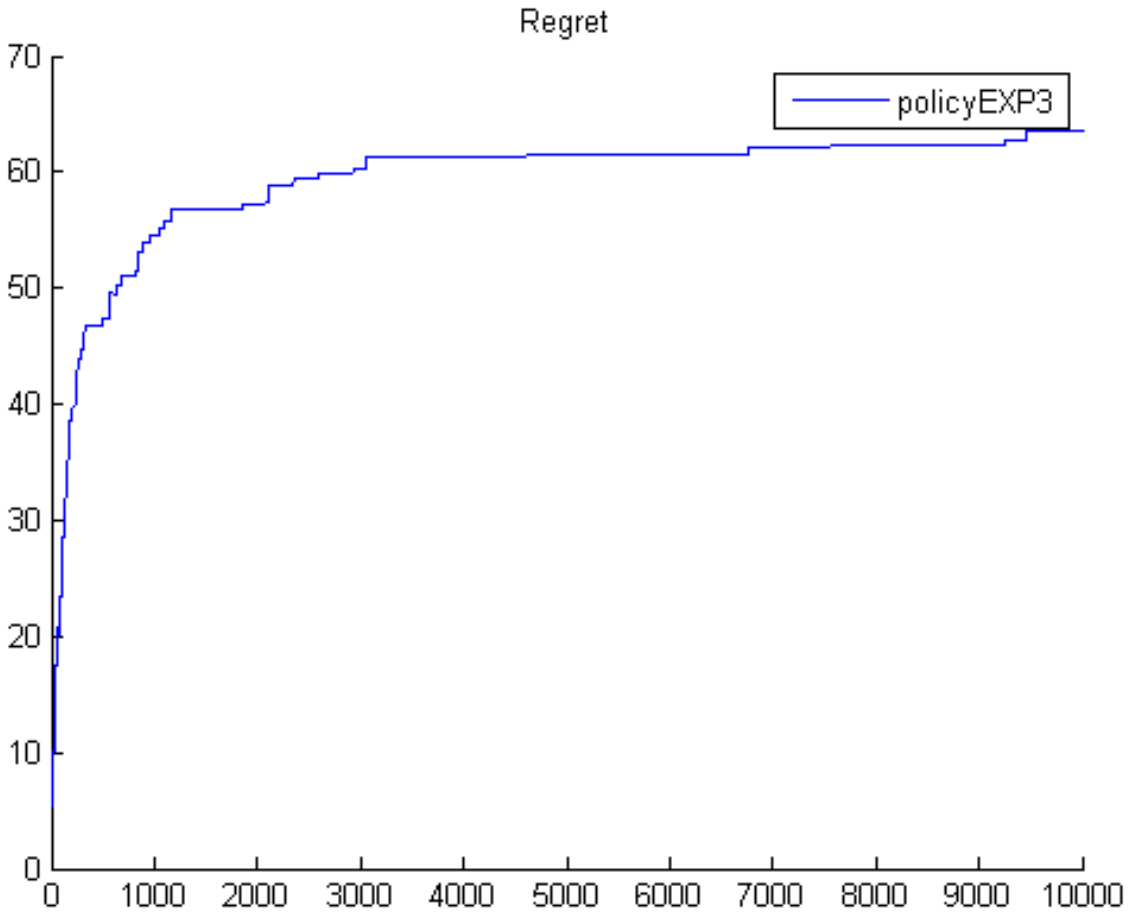


This shows that the changes from GWM to EXP3 make it no-regret. EXP3 works by scaling the weight of the action chosen by the probability that they were going to be chosen. This keeps the lack of information from ruining the prediction in the constant game.

3.3.1) Implement gameGaussian

See code for implementation

3.3.2) policyEXP3 on gameGaussian



The regret is more because the game is not constant and now has a gaussian distribution, which EXP3 doesn't know, but still levels out.

3.4.1) Derive the varience of the estimator $\sum_{t=1}^{T} \tilde{l_n^t}$. Is the variance bounded?

$$Var(X) = E\left[X^2\right] - (E\left[X\right])^2$$

Definition of variance

$$Var\left(\sum_{t=1}^{T} \tilde{l_n^t}\right) = E\left[\left(\sum_{t=1}^{T} \tilde{l_n^t}\right)^2\right] - \left(E\left[\sum_{t=1}^{T} \tilde{l_n^t}\right]\right)^2$$

From 3.1.3) above

$$E\left[\sum_{t=1}^{T} \tilde{l_n^t}\right] = \sum_{t=1}^{T} l_n^t$$

$$E\left[\left(\sum_{t=1}^{T} \tilde{l_n^t}\right)^2\right] = \left(\sum_{t=1}^{T} \tilde{l_n^t}\right)\left(\sum_{t=1}^{T} \tilde{l_n^t} p_n^t\right)$$

$$E\left[\left(\sum_{t=1}^{T} \tilde{l_n^t}\right)^2\right] = \left(\sum_{t=1}^{T} \frac{l_n^t}{p_n^t}\right)\left(\sum_{t=1}^{T} l_n^t\right)$$

$$Var\left(\sum_{t=1}^{T} \tilde{l_n^t}\right) = \left(\sum_{t=1}^{T} \frac{l_n^t}{p_n^t}\right)\left(\sum_{t=1}^{T} l_n^t\right) - \left(\sum_{t=1}^{T} l_n^t\right)^2$$

$$Var\left(\sum_{t=1}^{T} \tilde{l_n^t}\right) = \left(\sum_{t=1}^{T} l_n^t\right)\left(\left(\sum_{t=1}^{T} \frac{l_n^t}{p_n^t}\right) - \left(\sum_{t=1}^{T} l_n^t\right)\right)$$

The variance is not bounded. It will be very large if the probability is small.
4) Upper Confidence Bound (UCB) Algorithm
4.2.1) Show that the upper bound of the mean is the following with probability at least $1 - \delta$

$$\mu \leq \sum_{i=1}^{m} X_i + \sqrt{\frac{\log\left(\delta^{-1}\right)}{2m}}$$

By definition of the sampled mean

$$\hat{\mu} = \sum_{i=1}^{m} X_i$$

Rearranging the initial equation

$$\mu - \hat{\mu} \leq \sqrt{\frac{\log\left(\delta^{-1}\right)}{2m}}$$

Using Hoeffding's inequality

$$P(\mu - \hat{\mu} \geq \epsilon) \leq e^{-2m\epsilon^2}$$

Plugging in the desired probability $1 - \delta$

$$\delta \geq e^{-2m\epsilon^2}$$

Algebra

$$\epsilon \leq \sqrt{\frac{-\log \delta}{2m}}$$

Property of logarithms

$$\epsilon \leq \sqrt{\frac{\log \delta^{-1}}{2m}}$$

In terms of probability

$$P\left(\epsilon \leq \sqrt{\frac{\log \delta^{-1}}{2m}}\right) = \delta$$

Property of probability

$$P\left(\epsilon > \sqrt{\frac{\log \delta^{-1}}{2m}}\right) = 1 - \delta$$

Plugging in

$$P\left(\mu - \hat{\mu} \geq \epsilon > \sqrt{\frac{\log \delta^{-1}}{2m}}\right) = 1 - \delta$$

4.2.2) Show that the upper bound of the mean reward for an action n is the following

$$\mu_n^t \leq \hat{\mu}_n^t + \sqrt{\frac{\log t}{2C_n^t}}$$

By definition of t

$$t = \delta^{-1}$$

By definition of C_n^t

$$C_n^t = m$$

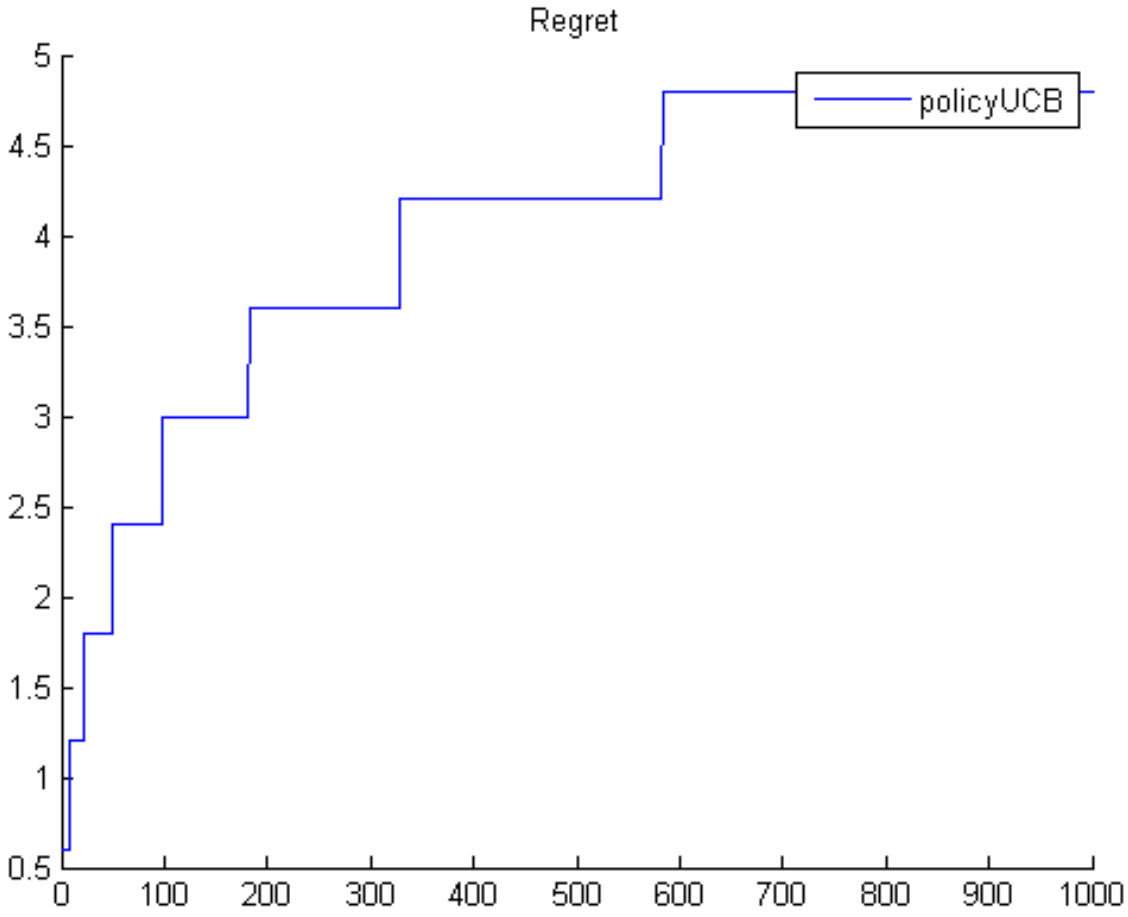
Using the same steps as above in 4.2.1) combined with definition of t

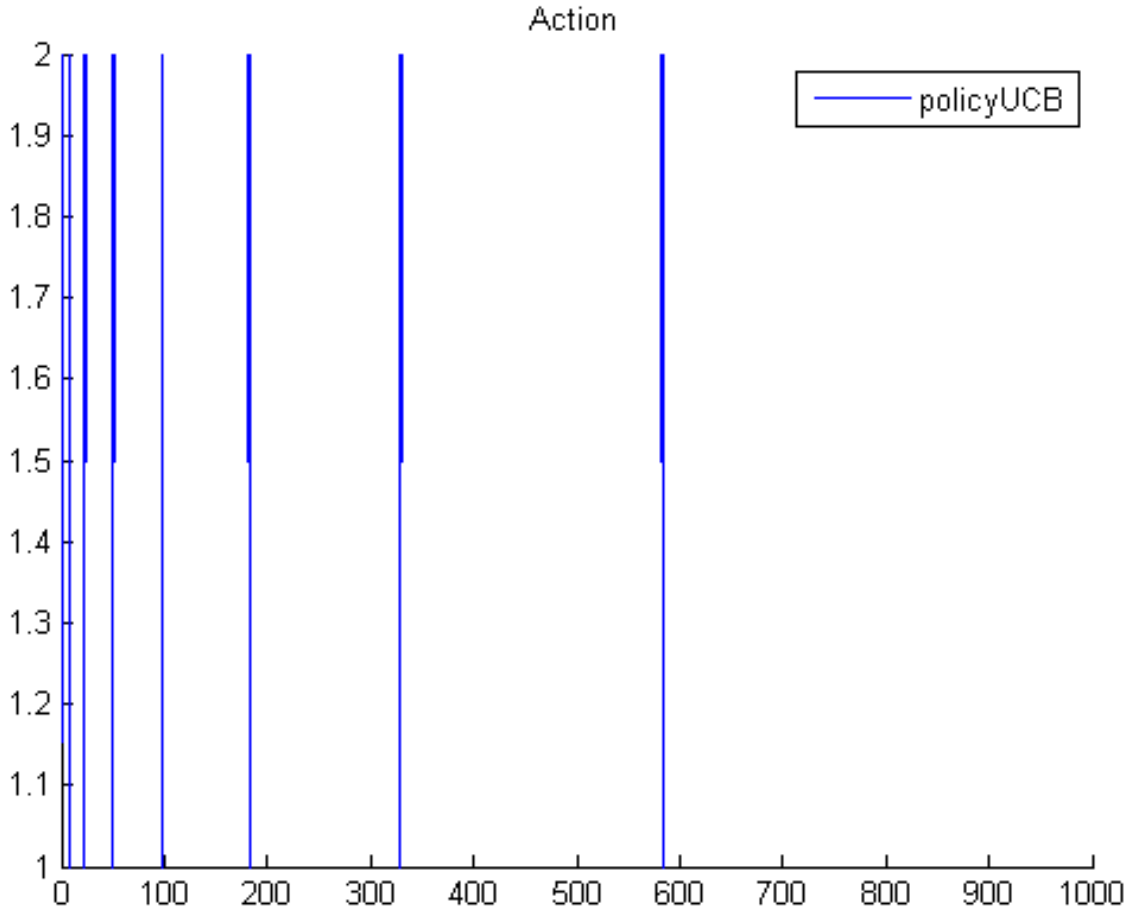
$$P\left(\mu - \hat{\mu} \geq \sqrt{\frac{\log t}{2C_n^t}}\right) = 1 - \frac{1}{t}$$

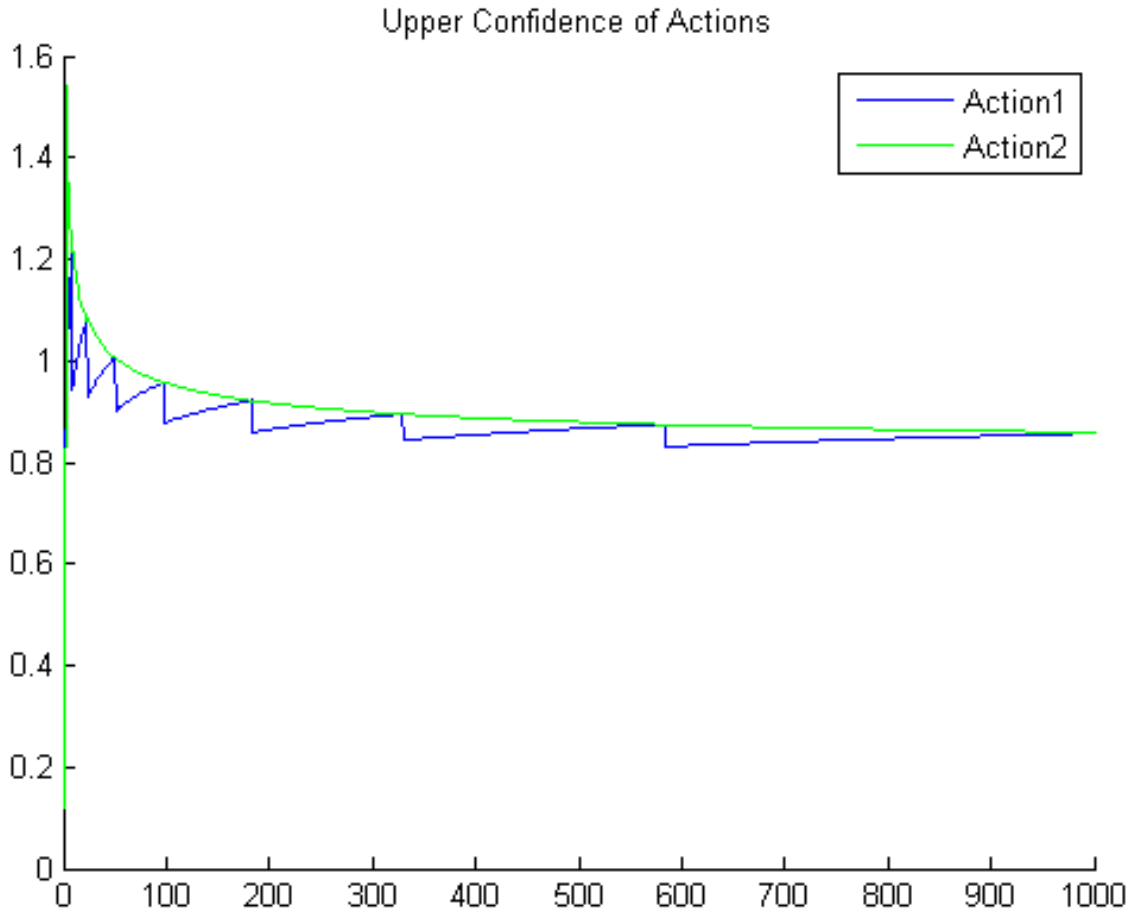
For large t this gives

$$P\left(\mu - \hat{\mu} \geq \sqrt{\frac{\log t}{2C_n^t}}\right) \approx 1$$

So it is the upper bounds when t gets large
4.3.1) policyUCB on gameConstant
see code for implementation

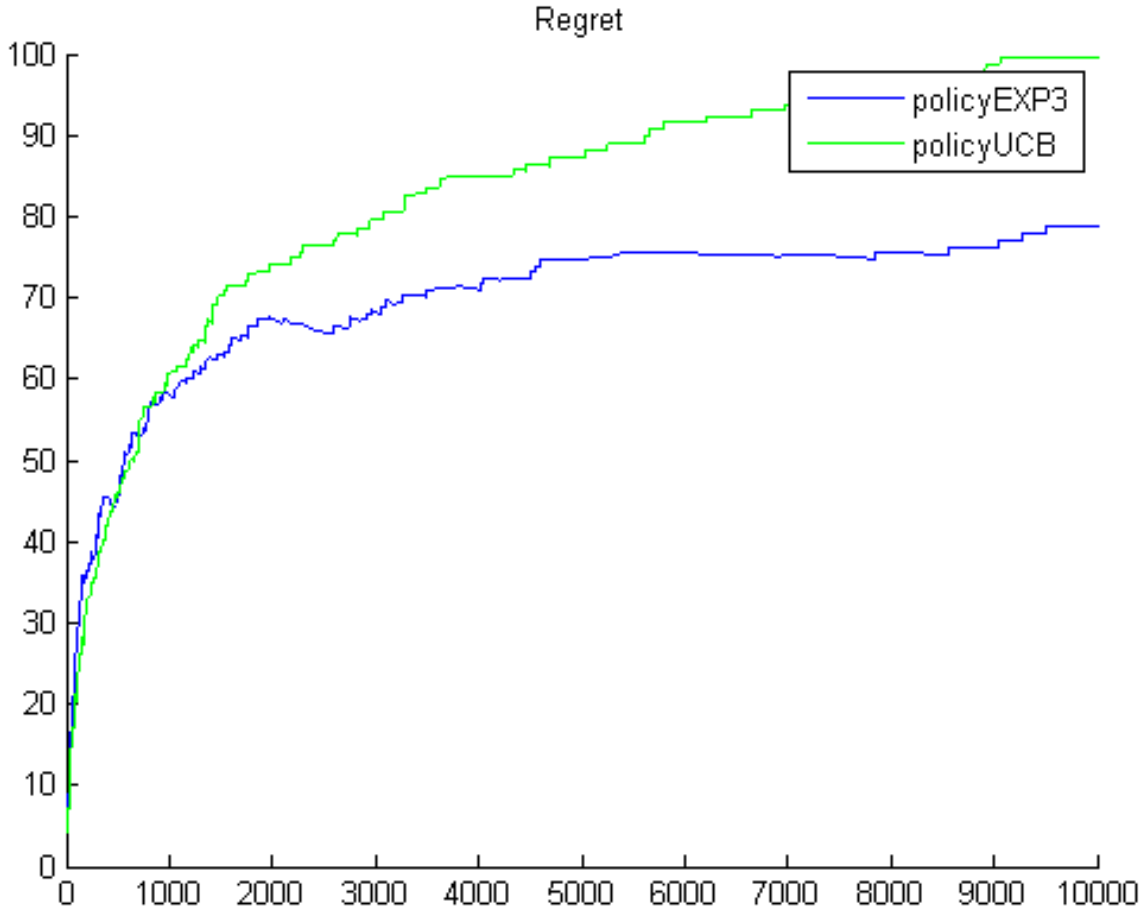



Action
policyUCB
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0
100
200
300
400
500
600
700
800
900
1000



UCB builds the confidence it has in each of the actions and chooses the action, which has the greatest estimated mean. Then based off the reward it updates the predicted mean for that particular action.

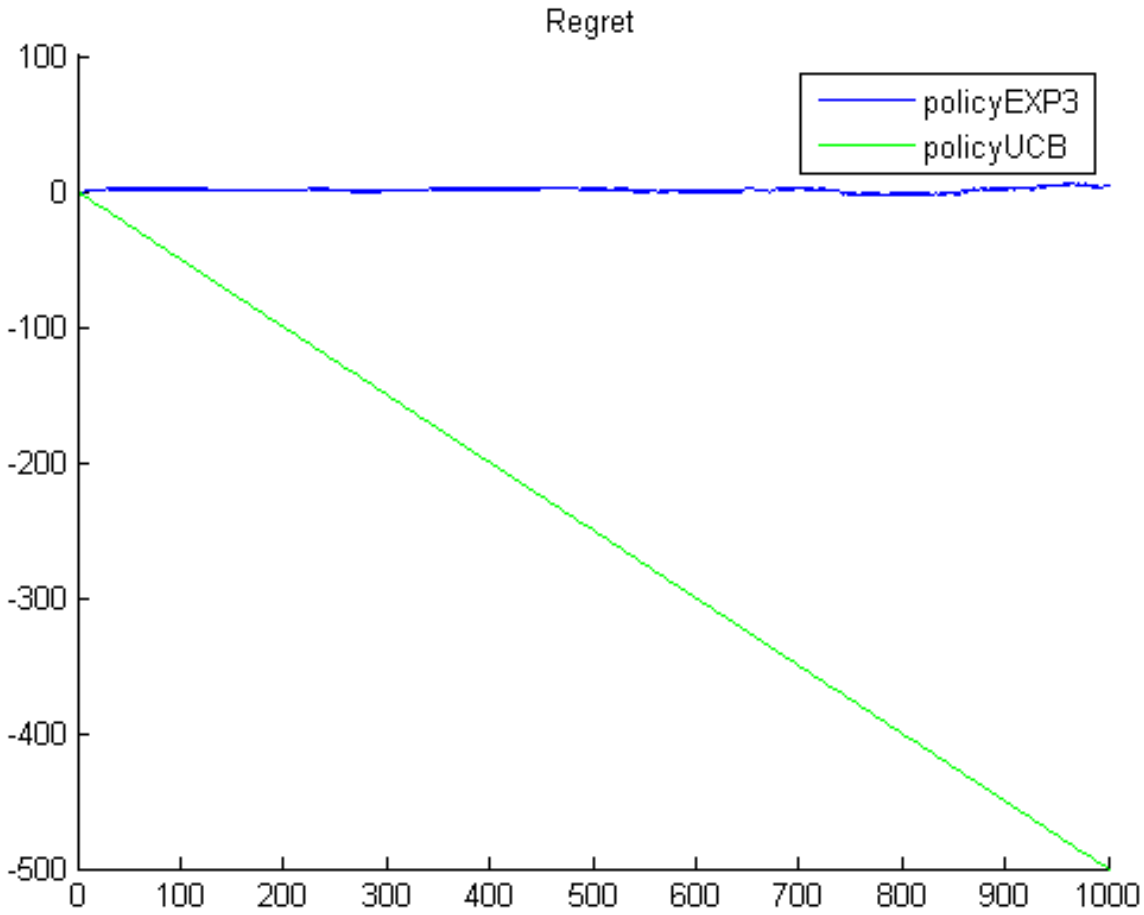
4.4.1) policyUCB and policyEXP3 on gameGaussian



EXP3 performs better in this case because UCB has an exploration portion of it, which can build more regret than the performance of the weights in EXP3.

4.5.1) policyUCB and policyEXP3 on gameAdversarial

See code for implementation



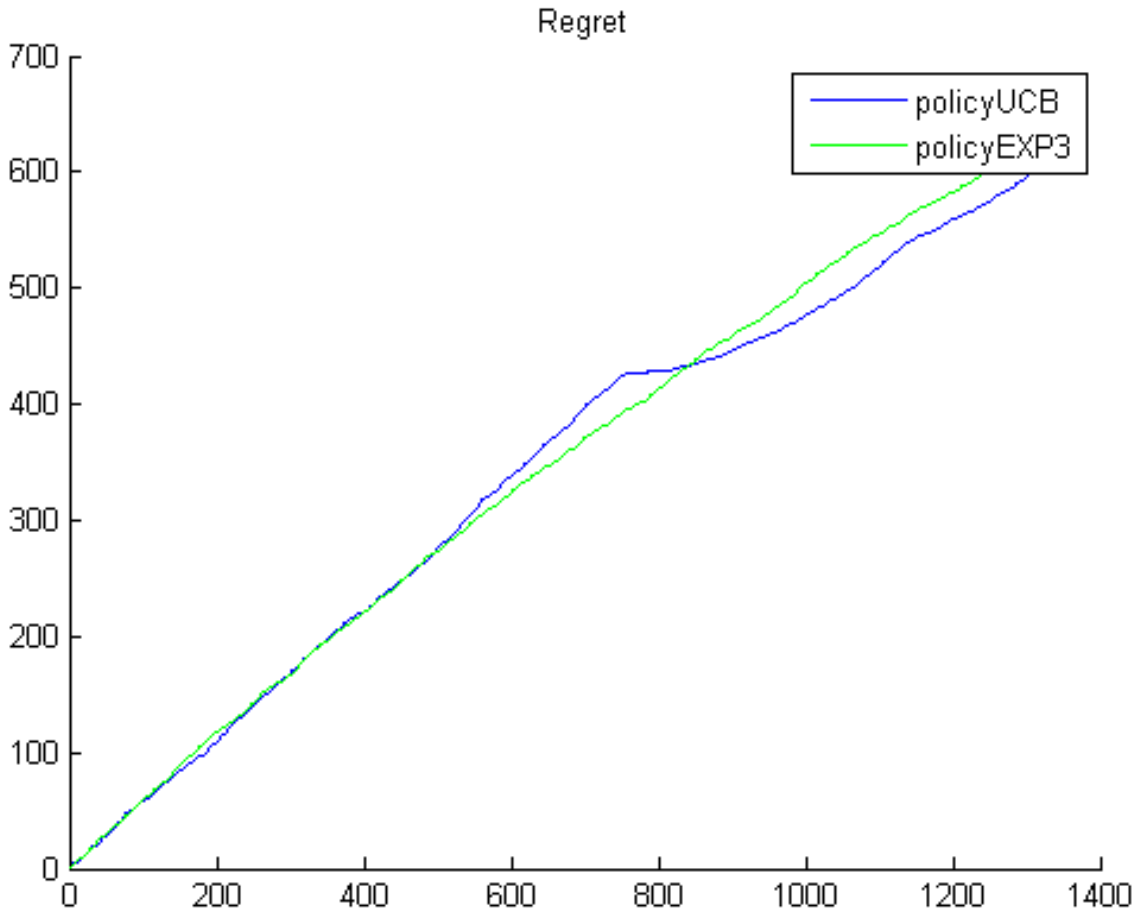
In this Adversarial case EXP3 maintains no-regret, but UCB does better because it switches off between actions and actually follows the Adversarial choices very well giving it negative regret.

5) Real Datasets

5.1.1) Implement gameLookupTable

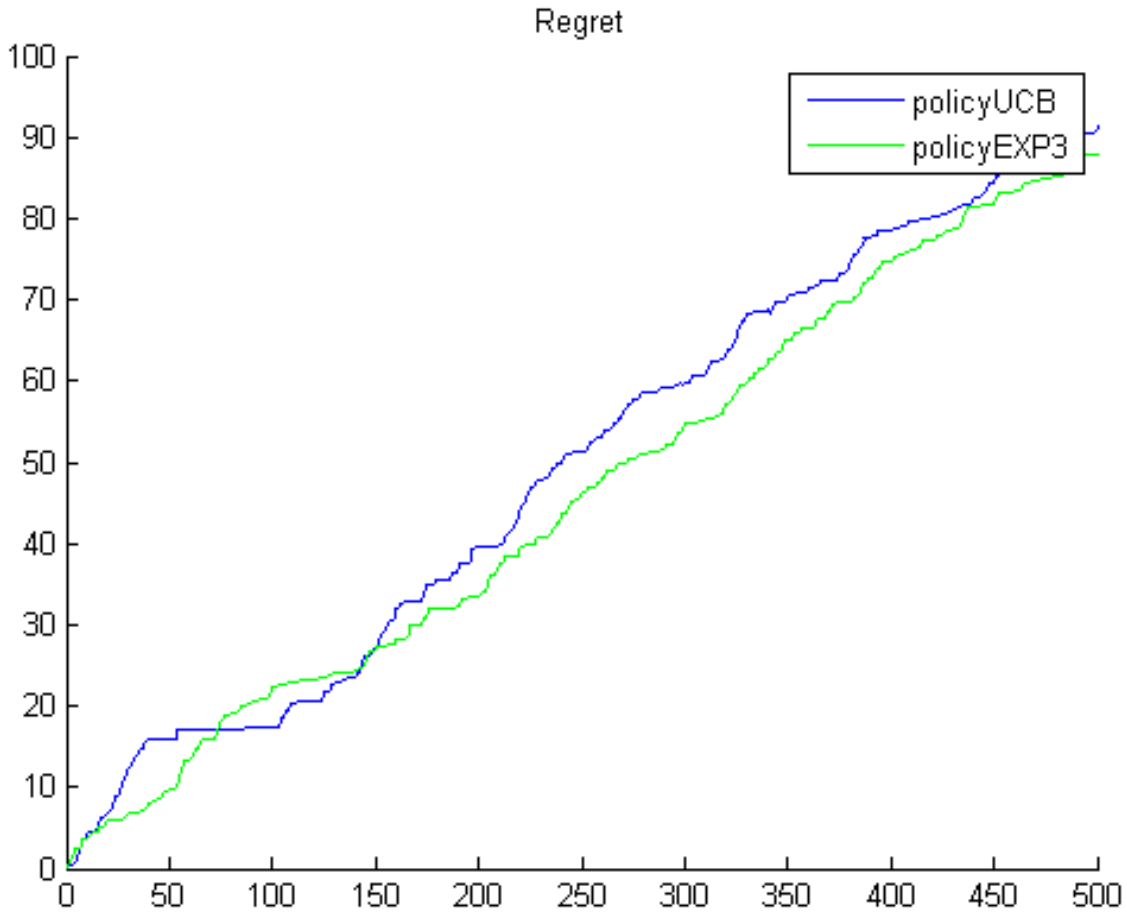
See code for implementation

5.2.1) policyUCB and policyEXP3 on gameLookupTable(univLatencies)



UCB does about as well as EXP3 because they each have so many actions and the time is too short for them to learn how to properly pick an action. It is important to note that UCB appears to be about to start gaining regret at a slower rate, while EXP3 is keeping a similar slope.

5.3.1) policyUCB and policyEXP3 on gameLookupTable(plannerPerformance)



This graph has a similar result as the one in 5.2.1, because of similar properties such as high ratio of number of actions to the total rounds in the game.