Excercise 2:

Contents

question

Day 3, Exercise # 2: RL

A bee is foraging among two flowers (yellow and blue) in search of nectar. The amount of nectar for each flower is stochastic, following normal distribution, blue with mean 1 and yellow with mean 0.5, both with 0.25 standard deviation.

The model bee has a stochastic policy, which means that it chooses blue and yellow flowers with probabilities that we write as *P*[b] and *P*[y] respectively. A convenient way to parameterize these probabilities is to use the softmax distribution:

$$P[\mathrm{b}] = \frac{\exp(\beta m_{\mathrm{b}})}{\exp(\beta m_{\mathrm{b}}) + \exp(\beta m_{\mathrm{y}})} \qquad P[\mathrm{y}] = \frac{\exp(\beta m_{\mathrm{y}})}{\exp(\beta m_{\mathrm{b}}) + \exp(\beta m_{\mathrm{y}})}$$

If the bee chooses a blue flower on a trial and receives nectar volume r_b, it should update the action value m_b according to the prediction error by $m_b \rightarrow m_b + \varepsilon\,\delta$ with $\delta = r_b - m_b$, and leave m_y unchanged. If it lands on a yellow flower, m_y is changed to $m_y \rightarrow m_y + \varepsilon\,\delta$ with $\delta = r_y - m_y$, and m_b is unchanged.
Simulate this model bee for 100 number of actions.
a) Draw the reward distribution for each flower using MATLAB “hist” function for 1000 samples to compare their average rewards.
b) Use $\beta = 1$ and $\varepsilon = 0.1$, and draw the cumulative visits to yellow and blue flowers versus history of actions.
c) Draw the action values (m_y, m_b) versus history of actions.
d) Change the parameter β to 0.1 and then 10. What do you observe?

Answer

```
clc
clear
close all

N_samples = 1000;
Reward_y = normrnd(0.5, 0.25, 1, N_samples);
Reward_b = normrnd(1, 0.25, 1, N_samples);


N_actions = 100;

beta = 1;  %this is a trade-off between exploration and exploitation (something like a learning rate!)
% very large beta values may lead to the wrong decision (choosing yellow)
epsilon = 0.1;

my_exp = zeros(1,N_actions);
mb_exp = zeros(1,N_actions);
ch = zeros(1,N_actions);

ch(1) = round (rand);
for i = 1:N_actions-1

    if ch(i) == 0
        my_exp(i+1) = my_exp(i) + epsilon * (Reward_y(i) - my_exp(i));
        mb_exp(i+1) = mb_exp(i);
    elseif ch(i) == 1
        mb_exp(i+1) = mb_exp(i) + epsilon * (Reward_b(i) - mb_exp(i));
        my_exp(i+1) = my_exp(i);
    end


    P_dens = exp(beta*my_exp(i+1)) + exp(beta*mb_exp(i+1));
    py = exp(beta * my_exp(i+1)) / P_dens;
    pb = exp(beta * mb_exp(i+1)) / P_dens;

    xy = py * rand(1);
```

```
    xb = pb * rand (1);
    if xy > xb
        ch(i+1) = 0;
    elseif xb > xy
        ch(i+1) = 1;
    end

end

% - plotting

bins = -2:0.1:3;

figure
subplot(2,2,1); hold on
hist(Reward_y, bins)
xlabel('Reward value (Yellow)')
ylabel('Probability')

subplot(2,2,2); hold on
hist(Reward_b, bins)
xlabel('Reward value (Blue)')
ylabel('Probability')

subplot(2,2,3); hold on
plot(cumsum(ch==0), 'LineWidth',2, 'Color','y')
plot(cumsum(ch==1), 'LineWidth',2, 'Color','b')
xlabel('Action history')
ylabel('Sum action')


subplot(2,2,4); hold on
plot(my_exp, 'LineWidth',2, 'color', 'y')
plot(mb_exp, 'LineWidth',2, 'color', 'b')
xlabel('Action history')
ylabel('Expected reward')
```