

```
function [] = exercise_2_5(nB,nA,nP,sigma)
%
% The 10-armed bandit testbed, with greedy action selection and incremental
% computation of the action values (\alpha = 1/k)
%
% Inputs:
%   nB: the number of bandits
%   nA: the number of arms
%   nP: the number of plays (times we will pull a arm)
%   sigma: the standard deviation of the return from each of the arms
%
% Written by:
% --
% John L. Weatherwax          2007-11-13
%
% email: wax@alum.mit.edu
%
% Please send comments and especially bug reports to the
% above email address.
%
%-----

%close all;
%clc;
%clear;

if( nargin<1 ) % the number of bandits:
    nB = 2000;
end
if( nargin<2 ) % the number of arms:
    nA = 10;
end
if( nargin<3 ) % the number of plays (times we will pull a arm):
    nP = 2000;
end
if( nargin<4 ) % the standard deviation of the return from each of the arms:
    sigma = 1.0;
end

%randn('seed',0);

% generate the TRUE reward  $Q^{\star}$ :
qStarMeans = mvnrnd( zeros(nB,nA), eye(nA) );

% run an experiment for each epsilon:
% 0 => fully greedy
% 1 => explore on each trial
epsArray = [ 0, 0.01, 0.1 ]; % , 1 ];

avgReward      = zeros(length(epsArray),nP);
perOptAction   = zeros(length(epsArray),nP);
cumReward      = zeros(length(epsArray),nP);
cumProb        = zeros(length(epsArray),nP);
for ei=1:length(epsArray),
    tEps = epsArray(ei);

    allRewards      = zeros(nB,nP);
    pickedMaxAction = zeros(nB,nP);
    for bi=1:nB, % pick a bandit
```

```

qT = zeros(1,nA); % <- initialize value function to zero (no knowledge)
qN = zeros(1,nA); % <- keep track of the number draws on each arm
for pi=1:nP, % make a play
    % determine if this move is exploratory or greedy:
    if( rand(1) <= tEps ) % pick a RANDOM arm:
        [dum,arm] = histc(rand(1),linspace(0,1+eps,nA+1)); clear dum;
    else % pick the GREEDY arm:
        [dum,arm] = max( qT ); clear dum;
    end
    % determine if the arm selected is the best possible:
    [dum,bestArm] = max( qStarMeans(bi,:) );
    if( arm==bestArm ) pickedMaxAction(bi,pi) = 1; end
    % get the reward from drawing on that arm:
    reward = qStarMeans(bi,arm) + sigma*randn(1);
    allRewards(bi,pi) = reward;
    % update qN,qT incrementally:
    qT(arm) = qT(arm) + ( reward-qT(arm) )/(qN(arm)+1);
    qN(arm) = qN(arm) + 1;
end
end

avgRew = mean(allRewards,1);
avgReward(ei,:) = avgRew(:).';
percentOptAction = mean(pickedMaxAction,1);
perOptAction(ei,:) = percentOptAction(:).';
csAR = cumsum(allRewards,2); % do a cumulative sum across plays for each
bandit
csRew = mean(csAR,1);
cumReward(ei,:) = csRew(:).';
csPA = cumsum(pickedMaxAction,2)./cumsum(ones(size(pickedMaxAction)),2);
csProb = mean(csPA,1);
cumProb(ei,:) = csProb(:).';
end

% produce the average rewards plot:
%
figure; hold on; clrStr = 'brkc'; all_hnds = [];
for ei=1:length(epsArray),
    %all_hnds(ei) = plot( [ 0, avgReward(ei,:) ], [clrStr(ei)] );
    all_hnds(ei) = plot( 1:nP, avgReward(ei,:), [clrStr(ei),'-'] );
end
legend( all_hnds, { '0', '0.01', '0.1' }, 'Location', 'SouthEast' );
axis tight; grid on;
xlabel( 'plays' ); ylabel( 'Average Reward' );

% produce the percent optimal action plot:
%
figure; hold on; clrStr = 'brkc'; all_hnds = [];
for ei=1:length(epsArray),
    %all_hnds(ei) = plot( [ 0, avgReward(ei,:) ], [clrStr(ei)] );
    all_hnds(ei) = plot( 1:nP, perOptAction(ei,:), [clrStr(ei),'-'] );
end
legend( all_hnds, { '0', '0.01', '0.1' }, 'Location', 'SouthEast' );
axis( [ 0, nP, 0, 1 ] ); axis tight; grid on;
xlabel( 'plays' ); ylabel( '% Optimal Action' );

% produce the cumulative average rewards plot:
%
figure; hold on; clrStr = 'brkc'; all_hnds = [];
for ei=1:length(epsArray),

```

```
    all_hnds(ei) = plot( 1:nP, cumReward(ei,:), [clrStr(ei),'-'] );
end
legend( all_hnds, { '0', '0.01', '0.1' }, 'Location', 'SouthEast' );
axis tight; grid on;
xlabel( 'plays' ); ylabel( 'Cummulative Average Reward' );

% produce the cummulative percent optimal action plot:
%
figure; hold on; clrStr = 'brkc'; all_hnds = [];
for ei=1:length(epsArray),
    all_hnds(ei) = plot( 1:nP, cumProb(ei,:), [clrStr(ei),'-'] );
end
legend( all_hnds, { '0', '0.01', '0.1' }, 'Location', 'SouthEast' );
axis( [ 0, nP, 0, 1 ] ); axis tight; grid on;
xlabel( 'plays' ); ylabel( 'Cummulative % Optimal Action' );
```