```
function [] = n_armed_testbed_softmax(nB,nA,nP,sigma)
%
% Generates the 10-armed bandit testbed with action selection using the
% softmax method and a Gibbs probability distribution with several temperatures.
%
% 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 = 10000;
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 temperature:
% temp << 1 => fully greedy method
% temp >> 1 => uniform selction fully exploritory
temperatureArray = [ 0.01, 0.1, 1 ];

% assume we have at least ONE draw from each "arm" (initialize with use the qStarMeans
matrix):
qT0 = mvnrnd( qStarMeans, eye(nA) );

avgReward    = zeros(length(temperatureArray),nP);
perOptAction = zeros(length(temperatureArray),nP);
cumReward    = zeros(length(temperatureArray),nP);
cumProb      = zeros(length(temperatureArray),nP);
for ei=1:length(temperatureArray),
  Temp = temperatureArray(ei);
```

```
  %qT = qT0;  % <- initialize to one draw per arm
  qT = zeros(size(qT0));  % <- initialize to zero draws per arm (no knowledge)
  qN = ones( nB, nA ); % keep track of the number draws on this arm
  qS = qT;             % keep track of the SUM of the rewards (qT = qS./qN)

  allRewards      = zeros(nB,nP);
  pickedMaxAction = zeros(nB,nP);
  for bi=1:nB, % pick a bandit
    for pi=1:nP, % make a play
      % determine the move to make based on past experience:
      %
      qt        = qT(bi,:);                              % <- pick the current belief on
the rewards to come from each action
      smp       = exp(qt/Temp) ./ sum(exp(qt/Temp));    % <- compute the softmax
probabilities (use Gibbs)
      [dum,arm] = histc(rand(1),[0,cumsum(smp)]); clear dum;
      if( arm<1 || arm>nA )
        arm
        qt
        smp
      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,qS,qT:
      qN(bi,arm) = qN(bi,arm)+1;
      qS(bi,arm) = qS(bi,arm)+reward;
      qT(bi,arm) = qS(bi,arm)/qN(bi,arm);
    end
  end

  avgRew          = mean(allRewards,1);
  avgReward(ei,:) = avgRew(:).';
  percentOptAction   = mean(pickedMaxAction,1);
  perOptAction(ei,:) = percentOptAction(:).';
  csAR            = cumsum(allRewards,2); % do a cummulative 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(temperatureArray),
  fprintf( 'working on temperature = %10.6f...\n', temperatureArray(ei) );
  %all_hnds(ei) = plot( [ 0, avgReward(ei,:) ], [clrStr(ei)] );
  all_hnds(ei) = plot( 1:nP, avgReward(ei,:), [clrStr(ei),'-'] );
end
legend( all_hnds, { '0.01:greedy', '0.1', '1:random' }, '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(temperatureArray),
  %all_hnds(ei) = plot( [ 0, avgReward(ei,:) ], [clrStr(ei)] );
  all_hnds(ei) = plot( 1:nP, perOptAction(ei,:), [clrStr(ei),'-'] );
end
legend( all_hnds, { '0.01:greedy', '0.1', '1:random' }, 'Location', 'SouthEast' );
axis( [ 0, nP, 0, 1 ] ); axis tight; grid on;
xlabel( 'plays' ); ylabel( '% Optimal Action' );

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