

```
% R_LEARN_ACQ_SCRIPT - R learning for the access control que problem
%
% Written by:
% --
% John L. Weatherwax          2007-12-03
%
% email: wax@alum.mit.edu
%
% Please send comments and especially bug reports to the
% above email address.
%
%-----

clear all;
close all;
clc;

addpath( genpath( '../.../FullBNT-1.0.4/KPMstats/' ) );

n_servers = 10;      % the number of servers
h          = 0.5;     % probability of getting a high paying customer
p          = 0.06;    % probability a machine in service finished during this timestep
%alpha     = 0.1;     % learning rate for Q
alpha      = 0.01;    % learning rate for Q
beta       = 0.01;    % learning rate for rho

%MAX_N_EPISODES = 1000;
%MAX_N_EPISODES = 1e4;
%MAX_N_EPISODES = 1e5;
MAX_N_EPISODES = 2e6;
MAX_N_EPISODES = 10e6;

[rho,Q,Qmax,Act] = R_learn_acq(alpha,beta,h,p,n_servers,MAX_N_EPISODES);

fprintf('rho = %f\n',rho);
fprintf('max optimal action function Qmax is given by ...\n');
fprintf('priority increases downward; number of free servers increase going right; \n');
fprintf('[1,2,3,4] x [1,2,3,4,5,6,7,8,9,,10]\n');
disp(Qmax);
fprintf('optimal (greedy) action is given by ...\n');
disp(Act);

figure; imagesc( 1:n_servers, [1, 2, 4, 8], Qmax ); colorbar;
xlabel( 'number of free servers' ); ylabel( 'cust. priority' );

figure; imagesc( 1:n_servers, [1, 2, 4, 8], Act ); colorbar;
xlabel( 'number of free servers' ); ylabel( 'cust. priority' );
saveas( gcf, 'optimal_greedy_policy', 'png' );

figure; hold on; grid on;
ph1=plot( 1:10, Qmax(1,:), '-b' );
ph2=plot( 1:10, Qmax(2,:), '-r' );
ph3=plot( 1:10, Qmax(3,:), '-g' );
ph4=plot( 1:10, Qmax(4,:), '-m' );
legend( [ ph1, ph2, ph3, ph4 ], { 'priority 1', 'priority 2', 'priority 4', 'priority 8'
} );
saveas( gcf, 'value_of_best_action', 'png' );
```