

```

% WINDY_GR_Script - Performs on-policy sarsa iterative action value funtion estimation
for the windy grid world example.
%
% 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.
%
%-----

%close all;

alpha = 1e-1;

sideII = 7; sideJJ = 10;

% the wind in each column:
wind = [ 0 0 0 1 1 1 2 2 1 0 ];

% the beginning and terminal states (in matrix notation):
s_start = [ 4, 1 ];
s_end = [ 4, 8 ];

MAX_N_EPISODES=30;
MAX_N_EPISODES=1e3; % 28 episodes to get around 8000 timesteps
MAX_N_EPISODES=1e4;
%MAX_N_EPISODES=1e5;
%MAX_N_EPISODES=1e6;
MAX_N_EPISODES=10e6;

[Q,ets] = windy_gw(alpha,sideII,sideJJ,s_start,s_end,wind,MAX_N_EPISODES);

pol_pi = zeros(sideII,sideJJ); V = zeros(sideII,sideJJ);
for ii=1:sideII,
    for jj=1:sideJJ,
        sti = sub2ind( [sideII,sideJJ], ii, jj );
        [V(ii,jj),pol_pi(ii,jj)] = max( Q(sti,:) );
    end
end

plot_gw_policy(pol_pi,s_start,s_end,wind);
title( 'policy (1=>up,2=>down,3=>right,4=>left)' );
fn = sprintf('windy_gw_policy_nE_%d',MAX_N_EPISODES); saveas( gcf, fn, 'png' );

figure; imagesc( V ); colormap(flipud(jet)); colorbar;
title( 'state value function' );
fn = sprintf('windy_gw_state_value_fn_nE_%d',MAX_N_EPISODES); saveas( gcf, fn, 'png' );

figure; plot( 1:length(ets), ets, '-x' ); windy_gw_ets = ets;
%fn = sprintf('windy_gw_learning_rate_nE_%d',MAX_N_EPISODES); saveas( gcf, fn, 'fig' );

```