

```

function [V] = rw_offline_ntd_learn(n,alpha,n_rw_states,nEpisodes)
% RW_OFFLINE_NTD_LEARN - Performs n-step learning of the random walk example
%
% Written by:
% --
% John L. Weatherwax          2007-12-07
%
% email: wax@alum.mit.edu
%
% Please send comments and especially bug reports to the
% above email address.
%
%-----

% the number of episodes (in general this would need to be much larger to guarantee
% complete learning):
%nEpisodes = 1000;

% the number of non terminal states plus two terminal states:
nStates=n_rw_states+2;

% initialize the nonterminal states ... terminal states are defined to be ZERO
%V = 0.5*ones(1,nStates); V(1)=0; V(end)=0;
V = zeros(1,nStates); V(1)=0; V(end)=0;

gamma      = 1.0;
%gamma     = 0.9;
gammaPowV = gamma.^[ 0:(n-1) ];

for ei=1:nEpisodes,

    % perform a random walk getting the sequence of rewards recieved and the
    % sequence of states we transition through:
    %
    [rewseen,stateseen] = rw_episode(n_rw_states);

    % compute  $R_t^{(n)}$  for  $t=0,1,2,\dots,T-1$  the discounted n-step return for these t's
    %
    %  $R_t^{(n)} = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots + \gamma^{n-1} r_{t+n} +$ 
    %  $\gamma^n V_t(s_{t+n})$ 
    %
    % and update  $V(\cdot)$ 
    %
    T = length(rewseen); Rtn = zeros(1,T); deltaV = zeros(1,nStates);
    for t=0:(T-1), % <- STATE index ...
        % get the rewards recieved  $r_{t+1}$  to  $r_{t+n}$ :
        n_rews = min(n,T-t);
        rs = rewseen( (t+1) : (t+n_rews) );
        % get the state value function at the state at  $s_{t+n+1}$ :
        st_tpn = stateseen( t+1+n_rews );
        Vst_tpn = V(st_tpn);
        % compute the n-step reward:
        Rtn(t+1) = dot( rs, gammaPowV(1:n_rews) ) + (gamma^n_rews)*Vst_tpn;

        % this is an offline algorithm so save all V updates (to be applied when the episode
    ends):
        st = stateseen(t+1);
        deltaV(st) = deltaV(st) + alpha*( Rtn(t+1)-V(st) );
    end
end

```

```
end

% apply the update stored in deltaV accumulated from this episode:
V = V + deltaV;
end

% remove the terminal states:
V = V(2:end-1);
```