report on Qlearning algorithms

SXY,SYF

September 21, 2023

1 algorithm

for $iSessions \leftarrow 0$**to**10 **do**
  Initialization: $Q \leftarrow InitQ$
  Initialization: prime_strategy $\leftarrow (init\_action, num\_players)$
  Initialization: state $\leftarrow (init\_action, num\_players)$
  Parameters: $\alpha = 0.25, \beta = 0.00001, \delta = 0.95$
  iterations counter*;*
  for $iters \leftarrow 0$ **to** max_iters **do**
    for $players \leftarrow 0$ **to** $num\_players$ **do**
      $temp\_Q_t \leftarrow Q^{players}(action_t, :)$
      $max\_Val_t^{players} \leftarrow max_{\{A:actions\}}(temp\_Q_t)$
      $prime\_strategy \leftarrow argmax_{\{A:actions\}}(temp\_Q_t)$
    end
    for $players \leftarrow 0$ **to** $num\_players$ **do**
      if $Uniform(0,1) < exp(-\beta * iters)$ **then**
        $action_{t+1} \leftarrow$*Uniform(0,1)*action_space*
      end
      else $action_{t+1} \leftarrow$*prime_strategy*
      ;
    end
    $temp\_Q_{t+1} \leftarrow Q(:, action_{t+1}, :)$
    $max\_Val_{t+1} \leftarrow max_{\{A:actions\}}(temp\_Q_{t+1})$
    for $players \leftarrow 0$ **to** $num\_players$ **do**
      $old\_Q \leftarrow Q^{players}(action_t, action_{t+1})$
      new_Q
      $\leftarrow (1-\alpha) * old\_Q + \alpha * \left[Profits^{players}(action_{t+1}) + \delta * max\_Val_{t+1}^{players}(action_{t+1})\right]$
      $Q^{players}(action_t, action_{t+1}) \leftarrow new\_Q$
    end
    $action_{t+1} \leftarrow$*prime_strategy*
  end
end

Algorithm 1: Q_learning algorithm