Actor Critic

Santiago Paternain and Miguel Calvo-Fullana
Electrical and Systems Engineering, University of Pennsylvania
{spater,cfullana}@seas.upenn.edu

October 24 —November 4, 2019

- A Markov Decision Process is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, P)$
- P is a Markov transition probability if for any $\mathcal{S}' \subseteq \mathcal{S}$ and $\mathcal{R}' \subseteq \mathcal{R}$

$$P[S_{t+1} \in \mathcal{S}', R_{t+1} \in \mathcal{R}' | S_t, A_t, \ldots, S_0, A_0] = P[S_{t+1} \in \mathcal{S}', R_{t+1} \in \mathcal{R}' | S_t, A_t]$$

- We select the actions based on parameterized policies $\pi_\theta(a|s)$
  - $\Rightarrow$ We cannot work with general continuous functions
  - $\Rightarrow$ Parameterization is necessary
- Find the best policy within the functions that our parameterization defines
  - $\Rightarrow$ "Best" is defined by the value function

$$v_{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \Big| S_0 = s\right]$$

  - $\Rightarrow$ Recall that the expectation is with respect to all the rewards seen
  - $\Rightarrow$ We write $\mathbb{E}_{\pi_\theta}$ to denote the we are following the policy $\pi_\theta(a|s)$

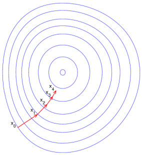
Maximizing the Value function

- Expanding the expectation, the value function is written as

$$v_{\pi_\theta}(s) = \sum_{k=0}^{\infty} \int_{\mathcal{R}^k \mathcal{A}^k \mathcal{S}^{k-1}} \gamma^k r_k \Pi_{j=0}^{k-1} p(s_{j+1}, r_{j+1} | s_j, a_j) \pi_\theta(a_j | s_j) \, d\mathbf{s}_k d\mathbf{a}_{k-1} d\mathbf{r}_k$$

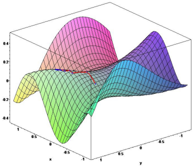
- Where the policy is parameterized by $\theta \in \mathbb{R}^d$
- We will update the parameters via gradient ascent
- Computing the gradient can be tricky (Today's lecture)

  $\Rightarrow$ Policy Gradient Theorem



- The gradient of v with respect to $\theta \in \mathbb{R}^d$ is

$$\nabla_\theta v_{\pi_\theta}(s) = \left(\frac{\partial v(\theta)}{\partial \theta_1}, \ldots, \frac{\partial v(\theta)}{\partial \theta_d} \right)^T$$



- To find the maximum, we update the parameters θ

$$\theta_{k+1} = \theta_k + \alpha_k \nabla_\theta v(\theta), \text{ with } \alpha_k > 0$$

- Recall that the expression of the gradient is

$$\nabla_\theta v_{\pi_\theta}(s) = (1-\gamma)^{-1}\mathbb{E}_{S\sim\rho_\theta, A\sim\pi_\theta}\left[q_{\pi_\theta}(S,A)\nabla\log\pi_\theta(A|S)\right]$$

- Where the distribution $\rho_\theta(s', s)$ is given by

$$\rho_\theta(s', s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p(S_t = s \mid S_0 = s')$$

- And $q(s,a)$ is the sate-action value function

$$q_{\pi_\theta}(s,a) = \mathbb{E}_{\pi_\theta}\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right]$$

- To compute the gradient we are required to have good estimates of q_{π_θ}

- REINFORCE is a Monte Carlo type method
- It uses one trajectory as a sample
- Basically run one trajectory and compute

$$\theta_{k+1} = \theta_k + \alpha_k \sum_{t=0}^{T-1} \nabla_\theta \log \pi_{\theta_k}(A_t|S_t) G_t$$

- Works because $\sum_{t=0}^{T-1} \nabla_\theta \log \pi_{\theta_k}(A_t|S_t) G_t$ is unbiased

$$\mathbb{E}_{\pi_{\theta_k}} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_{\theta_k}(A_t|S_t) G_t \right] = \nabla_\theta v(\theta_k)$$

- However the variance of an estimate is important for convergence

- In this example we try to maximize the following function

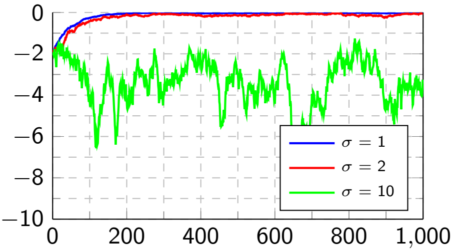
$$f(\mathbf{x}) = -\frac{1}{2}\|\mathbf{x}\|^2$$

- The gradient of this function is $\nabla f(\mathbf{x}) = -\mathbf{x}$

$\Rightarrow$ Estimate of the gradient is $-\mathbf{x}_k + \xi_k$ with $\xi_k \sim \mathcal{N}(0, \sigma^2)$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha(-\mathbf{x}_k + \xi_k)$$

- The estimate is unbiased $\mathbb{E}[\xi_k] = 0$
- Convergence is influenced by the variance
- We want estimates with small variance



- We introduced baselines to the estimate $\nabla_\theta \log \pi_\theta(A_t|S_t)\left(G_t - v(S_t)\right)$
- The new estimate is unbiased because $\mathbb{E}\left[\nabla_\theta \log \pi_{\theta_k}(A_t|S_t) v(S_t)\right] = 0$
- We argued that since $v_{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta}\left[G_t \mid S_t = s\right]$

  $\Rightarrow$ It is reasonable to expect that the variance is reduced

  $\Rightarrow$ But we did not prove it
- Since q_{π_θ} is also related to the return

$$q_{\pi_\theta}(s, a) = \mathbb{E}_{\pi_\theta}\left[G_t \mid S_t = s, A_t = a\right]$$

- If instead of looking at the return we were to look at at q_{π_θ}

  $\Rightarrow$ we should get a better estimate since we are "eliminating the noise "

What can we gain from knowing the q-function?

- Recall that the expression for the policy gradient for episodic tasks

$$\nabla_\theta v_{\pi_\theta}(s) = \mathbb{E}\left[G_t \nabla_\theta \log \pi_\theta(A_t|S_t) \mid S_t = s\right]$$

- The estimate that we have been using so far is

$$\hat{\nabla}_\theta v(s) = G_t \nabla_\theta \log \pi_\theta(A_t|S_t)$$

- However notice that we can also write

$$\begin{aligned}\nabla_\theta v_{\pi_\theta}(s) &= \mathbb{E}\left[\mathbb{E}\left[G_t \mid S_t, A_t\right] \nabla_\theta \log \pi_\theta(A_t|S_t) \mid S_t = s\right] \\ &= \mathbb{E}\left[Q(S_t, A_t) \nabla_\theta \log \pi_\theta(A_t|S_t) \mid S_t = s\right]\end{aligned}$$

- So we can also use the following estimate

$$\hat{\nabla}_\theta v_q = Q(S_t, A_t) \nabla_\theta \log \pi_\theta(A_t|S_t)$$

- It is also an unbiased estimate $\mathbb{E}\left[\hat{\nabla}_\theta v_q | S_t = s\right] = \nabla_\theta v_{\pi_\theta}(s)$

- Is this new estimate better in any sense?

$$\hat{\nabla}_\theta v_q(\theta) = Q(S_t, A_t) \nabla_\theta \log \pi_\theta(A_t|S_t)$$

- Let us compute the difference in covariance of the two estimates

$$\begin{aligned} Cov\left[\hat{\nabla}_\theta v(\theta)\right] - Cov\left[\hat{\nabla}_\theta v_q(\theta)\right] &= \mathbb{E}\left[G_t^2 \nabla_\theta \log \pi_\theta(A_t|S_t) \nabla_\theta \log \pi_\theta(A_t|S_t)^\top\right] \\ &\quad - \nabla_\theta v(\theta) \nabla_\theta v(\theta)^\top \\ &\quad - \mathbb{E}\left[Q(S_t, A_t)^2 \nabla_\theta \log \pi_\theta(A_t|S_t) \nabla_\theta \log \pi_\theta(A_t|S_t)^\top\right] \\ &\quad + \nabla_\theta v(\theta) \nabla_\theta v(\theta)^\top \end{aligned}$$

- Then it follows that

$$\begin{aligned} &Cov\left[\hat{\nabla}_\theta v(\theta)\right] - Cov\left[\hat{\nabla}_\theta v_q(\theta)\right] \\ &\quad = \mathbb{E}\left[\left(G_t^2 - Q(S_t, A_t)^2\right) \nabla_\theta \log \pi_\theta(A_t|S_t) \nabla_\theta \log \pi_\theta(A_t|S_t)^\top\right] \end{aligned}$$

- Working with $\hat{\nabla}_\theta v_q(\theta)$ is better if

$$Cov\left[\hat{\nabla}_\theta v(\theta)\right] - Cov\left[\hat{\nabla}_\theta v_q(\theta)\right] \geq 0$$

- Conditioning on S_t, A_t the previous expression yields

$$\Delta Var = \mathbb{E}\left[\mathbb{E}\left[G_t^2 - q_{\pi_\theta}(S_t, A_t)^2 \mid S_t, A_t\right] \nabla_\theta \log \pi_\theta(A_t|S_t) \nabla_\theta \log \pi_\theta(A_t|S_t)^\top\right]$$

- Let us show that the red expression is always non-negative

$$\mathbb{E}\left[G_t^2 - q_{\pi_\theta}(S_t, A_t)^2 \mid S_t, A_t\right] = \mathbb{E}\left[G_t^2 \mid S_t, A_t\right] - q_{\pi_\theta}(S_t, A_t)^2$$

- We have used that $q_{\pi_\theta}(S_t, A_t)$ is a deterministic function given S_t, A_t
- By definition we have that $q_{\pi_\theta}(S_t, A_t) = \mathbb{E}\left[G_t \mid S_t, A_t\right]$

$$\begin{aligned}\mathbb{E}\left[G_t^2 - q_{\pi_\theta}(S_t, A_t)^2 \mid S_t, A_t\right] &= \mathbb{E}\left[G_t^2 \mid S_t, A_t\right] - \mathbb{E}\left[G_t \mid S_t, A_t\right]^2 \\ &= Var(G_t \mid S_t, A_t) \geq 0\end{aligned}$$

- The variance of the estimate with the q_{π_θ} function is always reduced

- Having access to the q-function reduces the variance of our estimate
- This implies faster-convergence
- Why are they called actor-critic?
  - $\Rightarrow$ There is an actor: the agent choosing the policy
  - $\Rightarrow$ The critic is represented by the Q-function
  - $\Rightarrow$ It gives feedback on how good the action is for the given state
- Estimating the q-function is as easy as estimating the v-function
  - $\Rightarrow$ Monte-Carlo updates
  - $\Rightarrow$ TD updates
  - $\Rightarrow$ n-step and λ returns
- All of these methods can be used as well
  - $\Rightarrow$ Nothing really changes so we will go fast over them

- The q-function for the state-action $(s, a) \in \mathcal{S} \times \mathcal{A}$ and policy π is

$$q_\pi(s) = \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right] = \mathbb{E}_\pi \left[\sum_{k=0}^{T-1} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]$$

- Instead of computing the expectation we can consider the average return
  - $\Rightarrow$ Every time we visit the state s we estimate its return
  - $\Rightarrow$ And we average all these
  - $\Rightarrow$ Law of large numbers guarantees convergence to the expected value
- The first algorithm that we will see is First Visit Monte Carlo
  - $\Rightarrow$ This is a tabular method $\Rightarrow$ discrete state space

Input: Policy $\pi(A|S)$ and starting distribution $p(S_0)$
Initialize: $q(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ ▷ (Value function is set to zero)
$n(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ ▷ (Counter for visits set to zero)
for *episode* $k = 0, 1, 2, \ldots$ **do**
  Generate an episode following $\pi : S_0, A_0, R_1, S_1, A_1, \ldots, S_{T-1}, A_{T-1}, R_T$
  Set $G = 0$
  for *time* $t = T-1, \ldots, 1, 0$ **do**
    $G = \gamma G + R_{t+1}$ ▷ (Compute return of states $S_{T-1}, \ldots, S_1, S_0$)
    if $(S_t, A_t) \notin \{(S_0, A_0), (S_1, A_1) \ldots, (S_{t-1}, A_{t-1})\}$ **then**
      $n(S_t, A_t) = n(S_t, A_t) + 1$ ▷ (Increase counter for visit)
      $q(S_t, A_t) = q(S_t, A_t)\left(n(S_t, A_t) - 1\right)/n(S_t, A_t) + G/n(S_t, A_t)$ ▷ (Update mean)
    end
  end
end

Algorithm 1: First visit Monte Carlo

- For each state-action (s, a) we compute the return G_t given that $S_t = s, A_t = s$
- First visit $\Rightarrow$ we consider the return only the first time that we visit s, a
  - $\Rightarrow$ This means that for every episode we get a different return for s, a
  - $\Rightarrow$ And these returns are therefore i.i.d
  - $\Rightarrow$ They also have bounded variance
- So the law of the large numbers proves the convergence of the algorithm
- We can also do every-visit Monte Carlo
  - $\Rightarrow$ The returns are not independent in this case
  - $\Rightarrow$ We can still write it as a Stochastic Approximation problem
  - $\Rightarrow$ Similar to constant step Monte Carlo with a diminishing step-size

- Instead of computing the average we can do SGD
- Let us define the following error for each state

$$F(q) = \frac{1}{2}\left\|q(s,a) - q_\pi(s,a)\right\|^2$$

- Where $q(s,a)$ is an estimate of the value function under the policy π
- We have found the q function when $q(s,a) = q_\pi(s,a) \Rightarrow F(q) = 0$
- We can use SGD to minimize the function $F(q)$
- Compute the gradient with respect to $q(s,a)$

$$\frac{\partial F(q)}{q(s,a)} = q(s,a) - q_\pi(s,a) = q(s,a) - \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = s\right]$$

- So we can do Stochastic Approximation

$$q_{k+1}(S_t,A_t) = q_k(S_t,A_t) - \alpha\left(q_k(S_t,A_t) - G_t\right) = (1-\alpha)q_k(S_t,A_t) + \alpha G_t$$

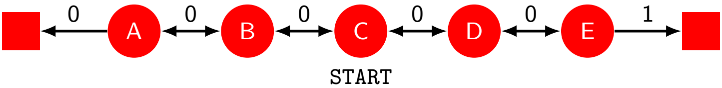
Input: Stepsize α, Policy $\pi(A|S)$ and starting distribution $p(S_0)$
Initialize: $q(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ ▷ (q function is set to zero)
for *episode* $k = 0, 1, 2, \ldots$ **do**
    Generate an episode following $\pi : S_0, A_0, R_1, S_1, A_1, \ldots, S_{T-1}, A_{T-1}, R_T$
    Set $G = 0$
    for *time* $t = T-1, \ldots, 1, 0$ **do**
        $G = \gamma G + R_{t+1}$ ▷ (Compute return of state-action $(S_{T-1}, A_{T-1}), \ldots, (S_1, A_1), (S_0, A_0)$)
        $q(S_t, A_t) = (1-\alpha) q(S_t, A_t) + \alpha G$ ▷ (Update using SGD)
    end
end

Algorithm 2: Constant step Monte Carlo

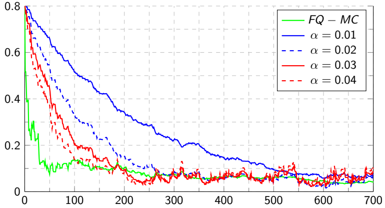
- Consider the MDP with uniform policy, i.e., for all $s \in \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}\}$

$$\pi(a = \mathtt{left}) = \pi(a = \mathtt{right}) = 0.5$$

- All transitions have zero rewards except from $s = \mathtt{E}$, with $a = \mathtt{right}$



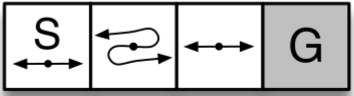
- The $q-$function for each state-action pair is the probability of reaching the terminal state on the right before the one on the left from the neighboring state.



Input: Parametric Policy $\pi_\theta(A|S)$, distribution $p(S_0)$,step-sizes α_θ,α_q
Initialize: $q(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ ▷ (Value function is set to zero)
Initialize: $\theta_0 = \theta$
for *episode* $k = 0, 1, 2, \ldots$ **do**
    Generate an episode following $\pi : S_0, A_0, R_1, S_1, A_1, \ldots, S_{T-1}, A_{T-1}, R_T$
    for *time* $t = T-1, \ldots, 1, 0$ **do**
        $G_t = \sum_{t'=t}^{T-1} R_{t'+1}$ ▷ (Compute return of states $S_{T-1}, \ldots, S_1, S_0$)
        $q(S_t, A_t) = q(S_t, A_t)(1 - \alpha_q) + G_t\alpha_q$
        $\nabla_\theta v(\theta) = \nabla_\theta v(\theta) + q(S_t, A_t)\nabla_\theta \log \pi_{\theta_k}(A_t|S_t)$
    end
    Update: $\theta_{k+1} = \theta_k + \alpha_\theta \nabla_\theta v(\theta)$
end

Algorithm 3: Monte Carlo Actor-Critic

- Consider the following short corridor
  - $\Rightarrow$ For each state there are two actions `left` or `right`
  - $\Rightarrow$ Transitions are normal but in the middle state they are reversed
  - $\Rightarrow$ All transitions give reward -1
  - $\Rightarrow$ Episode terminates when we reach G

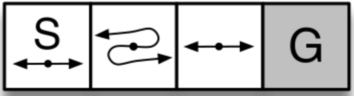


- We want to solve this problem using a very simple parameterization

$$x(s, \texttt{left}) = [1, 0] \quad x(s, \texttt{right}) = [0, 1]$$

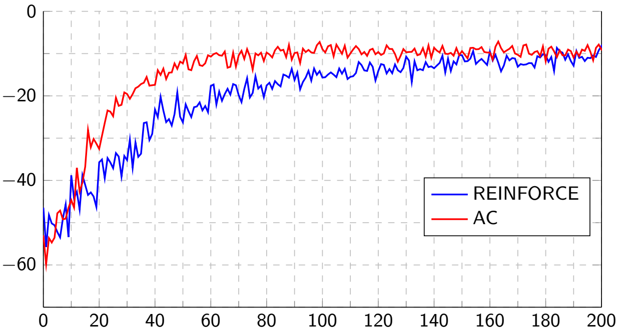
- Basically we follow the same policy regardless of the state

- If we follow the same policy regardless of the state



- There is no deterministic policy that is optimal
- The optimal policy is something around 50% on each direction
- However it has to be biased to the `right`
- We start with a bad policy defined by $\theta_1 = 0$ and $\theta_2 = 3$
- This gives us $\pi_\theta(\texttt{right}) \approx 0.05$

- We solve the previous example using REINFORCE and Actor Critic
- We select the step sizes to be $\alpha_\theta = 0.001$ and $\alpha_q = 0.01$
- We trained 100 examples and averaged the learning curves



- In both cases we get $\pi(\texttt{right}) \approx 0.54$
- Actor critic has better convergence properties

- We need to wait until the end of an episode to update the value function
  - $\Rightarrow$ Problem is that we can have episodes that are very long
  - $\Rightarrow$ What about continuing tasks? No episode at all
- We would like to operate step-by-step instead of episode-by-episode
  - $\Rightarrow$ This could accelerate learning but not possible with Monte Carlo
- Monte Carlo methods are simple to understand and use
  - $\Rightarrow$ Serve as good building blocks to more complex methods

- The q-function for the state-action $(s, a) \in \mathcal{S} \times \mathcal{A}$ and policy π is

$$q_\pi(s,a) = \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{T-1} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right]$$

- Recall that Monte Carlo methods need to wait until the end of the episode to update the value function
  - $\Rightarrow$ They operate in an episode-by-episode sense
- Now we look at Temporal Difference (TD) methods
  - $\Rightarrow$ They work in a step-by-step sense
- They update their estimated based on previous estimates
  - $\Rightarrow$ There is no need to wait for the final outcome of the episode
  - $\Rightarrow$ This concept is known as bootstraping

- The q-function also satisfies the Bellman's equation

$$q_\pi(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right]$$

- Recall that it is the only function that satisfies Bellman's equation
- Let us define the operator

$$\mathcal{B}(q)\Big|_{(s,a)} = \mathbb{E}_\pi\left[R_{t+1} + \gamma q(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right]$$

- So we have that $\mathcal{B}(q_\pi)|_{(s,a)} = q_\pi(s,a) \;\Rightarrow q_\pi$ is the only fixed point
- We can show that the operator is a contraction, i.e., for any q, q'

$$\left\|\mathcal{B}(q) - \mathcal{B}(q')\right\|_\infty \leq \gamma \left\|q - q'\right\|$$

- If the operator is a contraction and we apply it k times we have

$$\left\|\mathcal{B}(q)^k - \mathcal{B}(q')^k\right\|_\infty \leq \gamma \left\|\mathcal{B}(q)^{k-1} - \mathcal{B}(q')^{k-1}\right\|$$

- Recursively this yields

$$\left\|\mathcal{B}(q)^k - \mathcal{B}(q')^k\right\|_\infty \leq \gamma^k \left\|q - q'\right\|$$

- We have defined the operator

$$\mathcal{B}(q)\big|_{(s,a)} = \mathbb{E}_\pi\left[R_{t+1} + \gamma q(S_{t+1}) \mid S_t = s, A_t = a\right]$$

- For which q_π is a fixed point $\mathcal{B}(q_\pi) = q_\pi$
- And if it is a contraction (left to be shown) then we have that

$$\left\|\mathcal{B}(q)^k - \mathcal{B}(q')^k\right\|_\infty \leq \gamma^k \left\|q - q'\right\|$$

- Replacing q' by q_π in the previous equation yields

$$\left\|\mathcal{B}(q)^k - q_\pi\right\|_\infty \leq \gamma^k \left\|q - q_\pi\right\|$$

- Taking the limit of $k \to \infty$ establishes convergence for $\gamma \leq 1$
- If we apply the Bellman operator to any q we will converge to q_π

- We have defined the operator

$$\mathcal{B}(q)\Big|_{(s,a)} = \mathbb{E}_\pi\left[R_{t+1} + \gamma q(S_{t+1}) \mid S_t = s, A_t = a\right]$$

- We need to show that it is a contraction

$\Rightarrow$ The proof is the same as the proof for TD(0) estimation of v

- We have defined the operator

$$\mathcal{B}(q)\Big|_{(s,a)} = \mathbb{E}_\pi \left[R_{t+1} + \gamma q(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right]$$

- By its recursive application we can estimate q_π
- To compute the Bellman operator we need to compute an expectation

  $\Rightarrow$ Not efficient $\Rightarrow$ Let us try a stochastic approximation
- We want to find the fixed point of the Bellman operator $\mathcal{B}(q) - q = 0$
- Let us use Robbins-Monro $\Rightarrow$ Define $F(q) = \mathcal{B}(q) - q$
- Say that we have $S_t = s, A_t = a$ then the estimate of F is given by

$$\hat{F}(q_k)\Big|_{(s,a)} = R_{t+1} + \gamma q_k(S_{t+1}, A_{t+1}) - q_k(S_t, A_t)$$

- If we are able to get estimates of all the variables at the same time

$$q_{k+1} = q_k + \alpha \hat{F}(q_k)$$

- Which is the classic stochastic approximation

  $\Rightarrow$ We have convergence guarantees

Input: Policy $\pi(A|S)$, starting distribution $p(S_0)$, step-size α
Initialize: $q(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ ▷ (value function is set to zero)
for *episode* $k = 0, 1, 2, \ldots$ **do**
  Initialize S_0
  Choose $A \sim \pi(A|S)$
  for *each step of the episode* $t = 0, 1, \ldots, T-1$ **do**
    Take action A and observe R and S'
    Choose $A' \sim \pi(A'|S')$
    $q(S,A) = q(S,A) + \alpha[R + \gamma q(S',A') - q(S,A)]$ ▷ (Stochastic Approx)
    $S = S'$ ▷ (Update State)
    $A = A'$ ▷ (Update Action)
  end
end

Algorithm 4: Tabular TD(0)

- Notice that the previous algorithm is actually asynchronous
- We only update one of the entries at the time

$$\begin{aligned} q_{k+1}(S_t, A_t) &= q_k(S_t, A_t) + \alpha \left(R_{t+1} + \gamma q_k(S_{t+1}, A_{t+1}) - q_k(S_t, A_t)\right) \\ &= q_k(S_t, A_t) + \alpha \hat{F}(q(S_t, A_t)) \end{aligned}$$

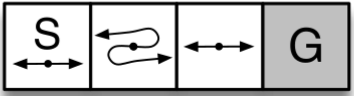
- The proof assumes that we compute $\hat{F}(q(S_t, A_t))$ for all states and actions
- Nonetheless, the proof can be extended for asynchronous updates[1]

[1] J.N. Tsitsiklis, "Asynchronous stochastic approximation and Q-learning", in *Machine Learning*, vol. 16, no. 1, pp. 185-202, 1994.

Input: Parametric Policy $\pi_\theta(A|S)$, distribution $p(S_0)$,step-sizes α_θ,α_q
Initialize: $q(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ ▷ (q function is set to zero)
Initialize: $\theta_0 = \theta$
for *episode* $k = 0, 1, 2, \ldots$ **do**
  Initialize S
  Choose $A \sim \pi_{\theta_k}(A|S)$
  for *each step of the episode* $t = 0, 1, \ldots, T-1$ **do**
    Take action A and observe R and S'
    Choose $A' \sim \pi_{\theta_k}(A'|S')$
    $q(S,A) = q(S,A) + \alpha_q\left(R + \gamma q(S',A') - q(S,A)\right)$
    $\nabla_\theta v(\theta) = \nabla_\theta v(\theta) + q(S,A)\nabla_\theta \log \pi_{\theta_k}(A|S)$
    $S = S'$
    $A = A'$
  end
  Update: $\theta_{k+1} = \theta_k + \alpha_\theta \nabla_\theta v(\theta)$
end

Algorithm 5: TD Actor-Critic

- Consider the following short corridor
  - $\Rightarrow$ For each state there are two actions `left` or `right`
  - $\Rightarrow$ Transitions are normal but in the middle state they are reversed
  - $\Rightarrow$ All transitions give reward -1
  - $\Rightarrow$ Episode terminates when we reach G

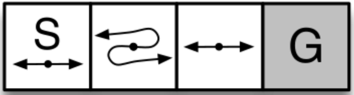


- We want to solve this problem using a very simple parameterization

$$x(s, \texttt{left}) = [1, 0] \quad x(s, \texttt{right}) = [0, 1]$$

- Basically we follow the same policy regardless of the state

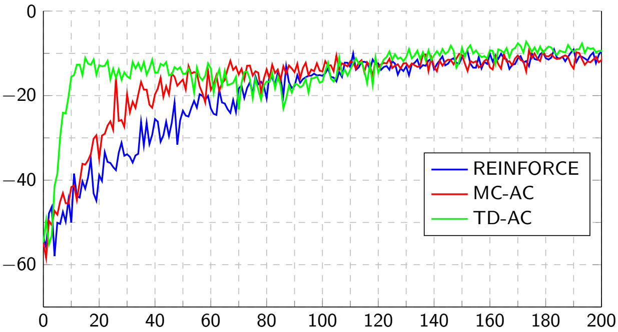
- If we follow the same policy regardless of the state



- There is no deterministic policy that is optimal
- The optimal policy is something around 50% on each direction
- However it has to be biased to the `right`
- We start with a bad policy defined by $\theta_1 = 0$ and $\theta_2 = 3$
- This gives us $\pi_\theta(\mathtt{right}) \approx 0.05$

- We solve the previous example using REINFORCE and Actor Critic
- We select the step sizes to be $\alpha_\theta = 0.001$ for all algorithms

  $\Rightarrow \alpha_q = 0.01$ for Montecarlo and $\alpha_q = 0.005$ for TD
- We trained 100 examples and averaged the learning curves



- In both cases we get $\pi(\texttt{right}) \approx 0.54$
- Actor critic with *TD* has better convergence properties

- So far we have been talking about using the q-function to reduce the variance of the estimate
- But baselines were used for the same reason $\Rightarrow$ Can we use both?

  $\Rightarrow$ Nothing prevents us from considering the following estimate

$$\hat{\nabla}_\theta v_a = \nabla_\theta \log \pi_\theta(A_t|S_t)\left(q(S_t, A_t) - v(S_t)\right)$$

- The reason for that is that $\mathbb{E}\left[\nabla_\theta \log \pi_\theta(A_t|S_t) v(S_t)\right] = 0$
- The difference between q and v is called the advantage function

$$a(S_t, A_t) = q(S_t, A_t) - v(S_t)$$

- It is a normalization with respect to the state

  $\Rightarrow$ How much an action can improve over the value of the current state

  $\Rightarrow$ Or the advantage of choosing a specific action

- Does that mean that we need to keep track both of q and v?
- Not really thanks to Bellman's equation

$$\begin{aligned} a(s,a) = q(s,a) - v(s) &= \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right] - v(s) \\ &= \mathbb{E}_\pi \left[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s, A_t = a \right] - v(s) \end{aligned}$$

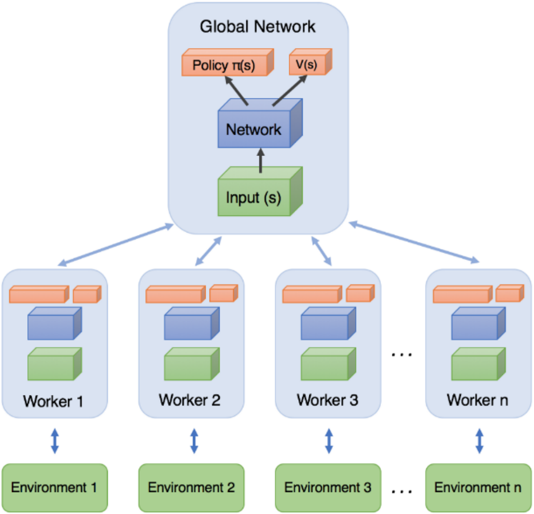
- Then we can estimate the gradient using

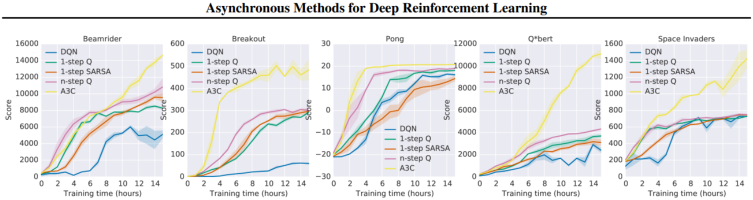
$$\begin{aligned} \hat{\nabla}_\theta v_a &= a(S_t, A_t) \nabla_\theta \log \pi_\theta(A_t | S_t) \\ &= \left(R_{t+1} + \gamma v(S_{t+1}) - v(S_t) \right) \nabla_\theta \log \pi_\theta(A_t | S_t) \end{aligned}$$

Input: Parametric Policy $\pi_\theta(A|S)$, distribution $p(S_0)$,step-sizes η_α,α_v
Initialize: $v(s) = 0$ for all $s \in \mathcal{S}$ ▷ (Value function is set to zero)
Initialize: $\theta_0 = \theta$
for *episode* $k = 0, 1, 2, \ldots$ **do**
  Initialize S
  Choose $A \sim \pi_{\theta_k}(A|S)$
  for *each step of the episode* $t = 0, 1, \ldots, T-1$ **do**
    Take action A and observe R and S'
    $v(S) = v(S) + \alpha_v \left(R + \gamma v(S') - v(S)\right)$
    $\nabla_\theta v(\theta) = \nabla_\theta v(\theta) + \left(R + \gamma v(S') - v(S)\right) \nabla_\theta \log \pi_{\theta_k}(A|S)$
    $S = S'$
    Choose $A \sim \pi_{\theta_k}(A|S)$
  end
  Update: $\theta_{k+1} = \theta_k + \alpha_\theta \nabla_\theta v(\theta)$
end

Algorithm 6: A2C

Asynchronos Advantage Actor-Critic (A3C)




Asynchronous Methods for Deep Reinforcement Learning
Beamrider
Breakout
Pong
Q*bert
Space Invaders
DQN
1-step Q
1-step SARSA
n-step Q
A3C
Score
Training time (hours)

- So far we have been doing On-Policy learning
- We use the same policy for actuation and training
- Learns about the policy that it is executing
- It is more natural as a framework
- Analysis is easier $\Rightarrow$ so it is a better place to start

- Off-Policy considers a different policy for training
- Executes one policy but it learns another one
- Learn about a policy while executing an exploratory policy
- Learn from demonstration or previous experience
- Learning multiple tasks from a single interaction with an environment
- Requires compensating for shift between behavior and target policy
  - $\Rightarrow$ It is called importance sampling
  - $\Rightarrow$ this increases variance, the more so when using multi-step updates

- Say we have termination at time $t+T$, then the value function is

$$v^{\pi}(s) = \mathbb{E}\left[R_{t+1} + \ldots + R_{t+T} \mid S_t = s\right]$$

- Let us denote by $b(a|s)$ the behavior policy
- Assume that the MDP is ergodic

$\Rightarrow$ There exists a steady state distribution under b

$$d_b(s) = \lim_{t\to\infty} P(S_t = s|s_0, b)$$

$\Rightarrow$ Intuition is that decisions have only a temporary effect

$\Rightarrow$ In the long run only the policy and the transition probability matters

- Under the assumption of said distribution we want to maximize

$$J(\theta) = \sum_{s\in\mathcal{S}} d_b(s) v_{\pi_\theta}(s)$$

- Sum of value functions weighted by how often we visit each state

Off-Policy Actor Critic

- If our goal is to maximize the objective

$$J(\theta) = \sum_{s \in \mathcal{S}} d_b(s) v_{\pi_\theta}(s)$$

- We can use a gradient ascent scheme

$$\nabla_\theta J(\theta) = \nabla_\theta \left(\sum_{s \in \mathcal{S}} d_b(s) v_{\pi_\theta}(s) \right) = \sum_{s \in \mathcal{S}} d_b(s) \nabla_\theta v_{\pi_\theta}(s)$$

- The behavior policy is independent of the learned policy
- Recall that the v-function satisfies that

$$v_{\pi_\theta}(s) = \sum_{a \in \mathcal{A}} \pi_\theta(a|s) q_{\pi_\theta}(s, a)$$

- Therefore the gradient of the v-function yields

$$\nabla_\theta v_{\pi_\theta}(s) = \sum_{a \in \mathcal{A}} \nabla_\theta \pi_\theta(a|s) q_{\pi_\theta}(s, a) + \sum_{a \in \mathcal{A}} \pi_\theta(a|s) \nabla_\theta q_{\pi_\theta}(s, a)$$

- The second term is difficult to estimate in an off-policy setting

$$g(\theta) = \sum_{s \in \mathcal{S}} d_b(s) \sum_{a \in \mathcal{A}} \nabla_\theta \pi_\theta(a|s) q_{\pi_\theta}(s, a)$$

- We have defined the following approximation of the gradient

$$g(\theta) = \sum_{s\in\mathcal{S}} d_b(s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi_\theta(a|s) q_{\pi_\theta}(s,a)$$

- And we will use it to update the policy as

$$\theta_{k+1} = \theta_k + \alpha g(\theta_k)$$

Theorem (Off-Policy Improvement[2])

For small enough step-size $\alpha > 0$ it follows that

$$J(\theta_{k+1}) \geq J(\theta_k) \quad \textit{and} \quad v_{\pi_{\theta_{k+1}}}(s) \geq v_{\pi_{\theta_k}}(s).$$

- Although we are not using the gradient it still improves the value function

[2] T. Degris, M. White and R. S. Sutton, "Off-Policy Actor-Critic", In Proceedings ICML 2012

- We have defined the following approximation of the gradient

$$g(\theta) = \sum_{s \in \mathcal{S}} d_b(s) \sum_{a \in \mathcal{A}} \nabla_\theta \pi_\theta(a|s) q_{\pi_\theta}(s, a)$$

- And we will use it to update the policy as

$$\theta_{k+1} = \theta_k + \alpha g(\theta_k)$$

- Use Taylor's theorem to write

$$\pi_{\theta_{k+1}}(a|s) = \pi_{\theta_k}(a|s) + \nabla_\theta \pi_{\theta_k}(a|s)^\top \alpha g(\theta_k) + o(\alpha^2)$$

- Therefore we have that

$$\begin{aligned}\pi_{\theta_{k+1}}(a|s) q_{\pi_{\theta_k}}(s, a) = {} & q_{\pi_{\theta_k}}(s, a) \pi_{\theta_k}(a|s) \\ & + q_{\pi_{\theta_k}}(s, a) \nabla_\theta \pi_{\theta_k}(a|s)^\top \alpha g(\theta_k) + o(\alpha^2)\end{aligned}$$

- Notice that we have

$$
\begin{aligned}
& q_{\pi_{\theta_k}}(s,a)\nabla_\theta \pi_{\theta_k}(a|s)^\top \alpha g(\theta_k) \\
&= \alpha q_{\pi_{\theta_k}}(s,a)\nabla_\theta \pi_{\theta_k}(a|s)^\top \sum_{s\in\mathcal{S}} d_b(s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi_{\theta_k}(a|s) q_{\pi_{\theta_k}}(s,a) \\
&= \alpha \sum_{s\in\mathcal{S}} d_b(s) \sum_{a\in\mathcal{A}} q_{\pi_{\theta_k}}(s,a)^2 \left\|\nabla_\theta \pi_{\theta_k}(a|s)\right\|^2 \geq 0
\end{aligned}
$$

- Because for tabular problems all the updates are independent
- Putting everything together we have that

$$
\begin{aligned}
\pi_{\theta_{k+1}}(a|s) q_{\pi_{\theta_k}}(s,a) &= q_{\pi_{\theta_{k+1}}}(s,a)\pi_{\theta_k}(a|s) \\
&\quad + q_{\pi_{\theta_k}}(s,a)\nabla_\theta \pi_{\theta_k}(a|s)^\top (\alpha g(\theta_k)) + o(\alpha^2)
\end{aligned}
$$

- Therefore, for small enough α we have that

$$
\pi_{\theta_{k+1}}(a|s) q_{\pi_{\theta_k}}(s,a) \geq \pi_{\theta_k}(a|s) q_{\pi_{\theta_k}}(s,a)
$$

- To show that $J(\theta_{k+1}) \geq J(\theta_k)$ and $v_{\pi_{\theta_{k+1}}}(s) \geq v_{\pi_{\theta_k}}(s)$ we can use

$$\pi_{\theta_{k+1}}(a|s) q_{\pi_{\theta_k}}(s,a) \geq \pi_{\theta_k}(a|s) q_{\pi_{\theta_k}}(s,a)$$

- We will do only one of the proofs, they are the same

$$\begin{aligned} v_{\pi_{\theta_k}}(s) &= \sum_{a \in \mathcal{A}} \pi_{\theta_k}(a|s) q_{\pi_{\theta_k}}(s,a) \leq \sum_{a \in \mathcal{A}} \pi_{\theta_{k+1}}(a|s) q_{\pi_{\theta_k}}(s,a) \\ &= \sum_{a \in \mathcal{A}} \pi_{\theta_{k+1}}(a|s) \mathbb{E}\left[R_{t+1} + \gamma v_{\pi_{\theta_k}}(S_{t+1}) \mid S_t = s\right] \\ &= \mathbb{E}_{A_t \sim \pi_{\theta_{k+1}}}\left[R_{t+1} + \gamma v_{\pi_{\theta_k}}(S_{t+1}) \mid S_t = s\right] \end{aligned}$$

- Applying the relationship recursively

$$v_{\pi_{\theta_k}}(s) \leq \mathbb{E}_{A \sim \pi_{\theta_{k+1}}}\left[G_t \mid S_t = s\right] = v_{\pi_{\theta_{k+1}}}(s)$$

- We have defined the following approximation of the gradient

$$g(\theta) = \sum_{s \in \mathcal{S}} d_b(s) \sum_{a \in \mathcal{A}} \nabla_\theta \pi_\theta(a|s) q_{\pi_\theta}(s, a)$$

Theorem (Off-Policy Policy-Gradient Theorem[3])

Let us define the set of critical points of $g(\theta)$ and $\nabla_\theta J(\theta)$

$$\mathcal{Z} = \{\theta \mid \nabla_\theta J(\theta) = 0\} \quad \textit{and} \quad \tilde{\mathcal{Z}} = \{\theta \mid g(\theta) = 0\}.$$

Then it follows that

$$\tilde{\mathcal{Z}} = \mathcal{Z}$$

[3] T. Degris, M. White and R. S. Sutton, "Off-Policy Actor-Critic", In Proceedings ICML 2012

- Recall the definitions

$$\mathcal{Z} = \{\theta \mid \nabla_\theta J(\theta) = 0\} \quad \text{and} \quad \tilde{\mathcal{Z}} = \{\theta \mid g(\theta) = 0\}.$$

- We will first show that $\mathcal{Z} \subset \tilde{\mathcal{Z}}$
- Assume that there exists some $\theta^\star \in \mathcal{Z}$ such that $\theta^\star \notin \tilde{\mathcal{Z}}$
- By the Policy Gradient Improvement Theorem it follows that

$$J(\theta^\star + \alpha g(\theta^\star)) > J(\theta^\star)$$

- So, $\theta^\star$ cannot be a local maximum of $J(\theta)$

- To prove the other inclusion let us show that if $\theta^\star \in \tilde{\mathcal{Z}}$ then $\nabla_\theta J(\theta^\star) = 0$
- Without loss of generality assume that we have m weights for state s_i then

$$g(\theta^\star)_{i,j} = \sum_{s' \in \mathcal{S}} d_b(s') \sum_{a \in \mathcal{A}} \frac{\partial}{\partial \theta_{i,j}} \pi_\theta(a|s') q_{\pi_\theta}(s', a)$$
$$= d_b(s_i) \sum_{a \in \mathcal{A}} \frac{\partial}{\partial \theta_{i,j}} \pi_\theta(a|s_i) q_{\pi_\theta}(s_i, a) = 0$$

- Assume that for s_i we have some k such that $\nabla_\theta J(\theta^\star) \neq 0$

$$\nabla_\theta J(\theta^*)_{ik} - g(\theta^\star)_{ik} = \sum_{s' \in \mathcal{S}} d_b(s') \sum_{a \in \mathcal{A}} \pi_\theta(a|s') \frac{\partial}{\partial \theta_{i,k}} q_{\pi_\theta}(s', a) \neq 0$$

- This term is the one that we decided not to consider

- We have from the previous slide that

$$\nabla_\theta J(\theta^*)_{ik} - g(\theta^\star)_{ik} = \sum_{s' \in \mathcal{S}} d_b(s') \sum_{a \in \mathcal{A}} \pi_\theta(a|s') \frac{\partial}{\partial \theta_{i,k}} q_{\pi_\theta}(s', a) \neq 0$$

- Which implies that

$$\nabla_\theta J(\theta^*)_{ik} = d_b(s_i) \sum_{a \in \mathcal{A}} \pi_\theta(a|s_i) \frac{\partial}{\partial \theta_{i,k}} q_{\pi_\theta}(s_i, a) \neq 0$$

- This means that we can improve $v_{\theta^\star}(s_i)$ by modifying the probabilities
- $\theta_{i,k}$ only influences state s_i hence to improve the value at state s_i

$$\sum_{j=1}^{m} \sum_{a \in \mathcal{A}} \frac{\partial}{\partial \theta_{i_s,j}} \pi_\theta(a|s) q_{\pi_\theta}(s, a) \neq 0$$

- Contradiction $\Rightarrow$ So $\tilde{\mathcal{Z}} \subset \mathcal{Z}$

- Recall that we are using the following approximation of the gradient

$$g(\theta) = \sum_{s \in \mathcal{S}} d_b(s) \sum_{a \in \mathcal{A}} \nabla_\theta \pi_\theta(a|s) q_{\pi_\theta}(s, a)$$

- And let us rewrite it as

$$g(\theta) = \mathbb{E}_{s \sim d_b} \left[\sum_{a \in \mathcal{A}} \nabla_\theta \pi_\theta(a|s) q_{\pi_\theta}(s, a) \right]$$

- We can write then

$$g(\theta) = \mathbb{E}_{s \sim d_b} \left[\sum_{a \in \mathcal{A}} b(a|s) \frac{\pi_\theta(a|s)}{b(a|s)} \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)} q_{\pi_\theta}(s, a) \right]$$

- Defining $\rho(s, a) = \pi_\theta(a|s)/b(a|s)$ and using the log trick

$$g(\theta) = \mathbb{E}_{s \sim d_b, a \sim b} \left[\rho(s, a) q_{\pi_\theta}(s, a) \nabla_\theta \log \pi_\theta(a|s) \right]$$

- From the previous slide with $\rho(s,a) = \pi_\theta(a|s)/b(a|s)$

$$g(\theta) = \mathbb{E}_{s\sim d_b, a\sim b}\left[\rho(s,a) q_{\pi_\theta}(s,a) \nabla_\theta \log \pi_\theta(a|s)\right]$$

- Similar to the policy gradient but are including the importance sampling
- Introducing a baseline and a stochastic approximation we have that

$$\theta_{k+1} = \theta_k + \alpha\rho(S_t,A_t)\psi(S_t,A_t)\left(G_t^\lambda - v(s_t)\right)$$

- where G_t^λ is the λ-return

$$G_t^\lambda = R_{t+1} + (1-\lambda)v(S_{t+1}) + \lambda\rho(S_{t+1},A_{t+1})G_{t+1}^\lambda$$

- This only means that we are using $TD(\lambda)$ for the estimation of the critic
- If we want to use $TD(0)$ just set $\lambda = 0$ and then

$$G_t = R_{t+1} + v(S_{t+1})$$

Input: Policies $\pi_\theta(A|S)$, $b(A|S)$ starting distribution $p(S_0)$, step-sizes α_v,α_θ
Initialize: $v(s) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ ▷ (value function is set to zero)
$\theta_0 = \theta$ ▷ (Initial parameters)
for *episode* $k = 0, 1, 2, \ldots$ **do**
    Initialize S_0
    Choose $A \sim b(A|S)$
    for *each step of the episode* $t = 0, 1, \ldots, T-1$ **do**
        Take action A and observe R and S'
        $v(S) = v(S) + \alpha_v[R + \gamma v(S') - v(S)]$ ▷ (Stochastic Approx)
        $\nabla_\theta v(\theta) = \nabla_\theta v(\theta) + \rho(S, A)(R + v(S') - v(S))\nabla_\theta \log \pi_{\theta_k}(A|S)$
        $S = S'$ ▷ (Update State)
        Choose $A \sim b(A|S)$
    end
    $\theta_{k+1} = \theta_k + \alpha_\theta \nabla_\theta v(\theta)$
end

Algorithm 7: Off-Policy AC $TD(0)$

- The algorithm is a stochastic approximation of of the defined function

$$g(\theta) = \sum_{s\in\mathcal{S}} d_b(s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi_\theta(a|s) q_{\pi_\theta}(s,a)$$

- And we have established two important results

$$v_{\pi_{\theta_{k+1}}}(s) \geq v_{\pi_{\theta_k}}(s)$$

$\Rightarrow$ Which means that in expectation the v-function increases with

$$\theta_{k+1} = \theta_k + \alpha \hat{g}(\theta_k)$$

- Because the value function is upper bounded then v converges
- It will converge to the points where $g(\theta) = 0$
- Since the critical points of $g(\theta)$ are the same as those of $J(\theta)$
- The algorithm converges to the set of critical points of $J(\theta)$

Outline

- So far we have been considering mainly random policies
  - $\Rightarrow$ Gaussian $\pi_\theta(a|s) = \frac{1}{\sqrt{(2\pi)}} exp(-\|a - \mu_\theta(s)\|^2/2)$
  - $\Rightarrow$ Soft-max $\pi_\theta(a|s) = \frac{e^{f(a,s,\theta)}}{\sum_{a' \in \mathcal{A}} f(a',s,\theta)}$
- Random policies help with exploration
- They are more robust to modeling errors
- If we are sure our system is an MDP why not using deterministic policies?
- For exploration we can do off-policy training

- For stochastic policies we have derived the policy gradient theorem

$$\nabla_\theta v(\theta) = (1-\gamma)^{-1} \mathbb{E}_{s\sim\rho_\theta, a\sim\pi_\theta} \left[\nabla_\theta \log \pi_\theta(a|s) q_{\pi_\theta}(s,a)\right]$$

- where the distribution ρ is defined as

$$\rho_\theta(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t p(s_t = s | s_0, \theta)$$

- We have discussed how to sample from the distribution ρ_θ
- And how to estimate the q-function $\Rightarrow$ Actor-Critic Algorithms
  - $\Rightarrow$ We studied the off-policy Actor-Critic
  - $\Rightarrow$ Use an off-policy stochastic actor-critic to learn a determinstic policy

- Let us consider a deterministic policy $a = \mu(s, \theta)$ and define as usual

$$v_\theta(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | s_0 = s\right]$$

- Let $\rho_\mu(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t p(s_t = s | s_0)$
- Then the gradient of the value function with respect to μ yields

Theorem (Deterministic Policy Gradient [4])

$$\nabla_\theta v(\theta) = (1-\gamma)^{-1} \mathbb{E}_{s \sim \rho_\mu}\left[\nabla_\theta \mu_\theta(s) \nabla_a q_{\mu_\theta}(s, a)\big|_{a=\mu(s)}\right]$$

[4] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra and M. Riedmiller "Deterministic Policy Gradient Algorithms" In Proceedings ICML 2014

- Let us start by using the Bellman's equation to write

$$v_\theta(s) = \mathbb{E}\left[R_{t+1} + \gamma v_\theta(S_{t+1}) \mid S_t = s, A_t = \mu_\theta(s)\right]$$

- Which in integral form yields

$$v_\theta(s) = \int_{\mathcal{R}\times\mathcal{S}} \left(r + \gamma v_\theta(s')\right) p(r, s'|s, \mu_\theta(s))\, ds'\, dr$$

- Let us compute the gradient with respect to θ

$$\begin{aligned}\nabla_\theta v_\theta(s) = &\int_{\mathcal{R}\times\mathcal{S}} \nabla_\theta \left(r + \gamma v_\theta(s')\right) p(r, s'|s, \mu_\theta(s))\, ds'\, dr \\ &+ \int_{\mathcal{R}\times\mathcal{S}} \left(r + \gamma v_\theta(s')\right) \nabla_\theta p(r, s'|s, \mu_\theta(s))\, ds'\, dr\end{aligned}$$

- The first term just yields $\gamma \nabla_\theta v_\theta(s')$

$$\begin{aligned}\nabla_\theta v_\theta(s) = &\gamma \int_{\mathcal{S}} \nabla_\theta v_\theta(s') p(r, s'|s, \mu_\theta(s))\, ds' \\ &+ \int_{\mathcal{R}\times\mathcal{S}} \left(r + \gamma v_\theta(s')\right) \nabla_\theta p(r, s'|s, \mu_\theta(s))\, ds'\, dr\end{aligned}$$

- From the previous slide we have that

$$\nabla_\theta v_\theta(s) = \gamma \int_{\mathcal{S}} \nabla_\theta v_\theta(s') p(r, s' | s, \mu_\theta(s)) \, ds' + \int_{\mathcal{R}\times\mathcal{S}} (r + \gamma v_\theta(s')) \, \nabla_\theta p(r, s' | s, \mu_\theta(s)) \, ds' dr$$

- Using the chain rule we have that

$$\nabla_\theta p(r, s' | s, \mu_\theta(s)) = \nabla_a p(r, s' | s, a)|_{a=\mu_\theta(s)} \nabla_\theta \mu_\theta(s)$$

- Rearrenging terms we have that

$$\nabla_\theta v_\theta(s) = \gamma \int_{\mathcal{S}} \nabla_\theta v_\theta(s') p(r, s' | s, \mu_\theta(s)) \, ds' + \nabla_a \left(\int_{\mathcal{R}\times\mathcal{S}} (r + \gamma v_\theta(s')) \, p(r, s' | s, a) \, ds' dr \right) \Big|_{a=\mu_\theta(s)} \nabla_\theta \mu_\theta(s)$$

- By Bellman's equation the term in the parenthesis is $q_\theta(s, a)$

$$\nabla_\theta v_\theta(s) = \gamma \int_{\mathcal{S}} \nabla_\theta v_\theta(s') p(r, s' | s, \mu_\theta(s)) \, ds' + \nabla_\theta \mu_\theta(s) \nabla_a q_\theta(s, \mu_\theta(s))$$

- From the previous slide we have that Rearrenging terms we have that

$$\nabla_\theta v_\theta(s) = \gamma \int_{\mathcal{S}} \nabla_\theta v_\theta(s') p(r, s'|s, \mu_\theta(s))\, ds' + \nabla_\theta \mu_\theta(s) \nabla_a q_\theta(s, \mu_\theta(s))$$

- Is a linear integral system of equations

  $\Rightarrow$ same ideas as the previous policy gradient proof apply

$$\nabla_\theta v(s_0) = \int_{\mathcal{S}} \nabla_\theta \mu_\theta(s') \nabla_a q_{\mu_\theta}(s', a)\big|_{a=\mu(s')} \sum_{t=0}^{\infty} \gamma^t p(s_t = s'|s_0 = s)\, ds'$$

- This sum appears from applying the recursion
- Defining $\rho_\theta(s) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t p(s_t = s'|s_0 = s)$

$$\nabla_\theta v(s_0) = (1-\gamma)^{-1} \int_{\mathcal{S}} \nabla_\theta \mu_\theta(s') \nabla_a q_{\mu_\theta}(s', a)\big|_{a=\mu(s')} \rho_\theta(s')\, ds'$$
$$= (1-\gamma)^{-1} \mathbb{E}_{s \sim \rho_\mu} \left[\nabla_\theta \mu_\theta(s) \nabla_a q_{\mu_\theta}(s, a)\big|_{a=\mu(s)} \right]$$

- The goal is to understand the relationship between
  - $\Rightarrow$ the stochastic policy gradient

$$\nabla_\theta v(\theta) = (1-\gamma)^{-1}\mathbb{E}_{s\sim\rho_\theta, a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) q_{\pi_\theta}(s,a)\right]$$
$$= (1-\gamma)^{-1}\mathbb{E}_{s\sim\rho_\theta}\left[\mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) q_{\pi_\theta}(s,a) \mid s\right]\right]$$

  - $\Rightarrow$ and the deterministic policy gradient

$$\nabla_\theta v(s_0) = (1-\gamma)^{-1}\mathbb{E}_{s\sim\rho_\mu}\left[\nabla_\theta \mu_\theta(s) \nabla_a q_{\mu_\theta}(s,a)\Big|_{a=\mu(s)}\right]$$

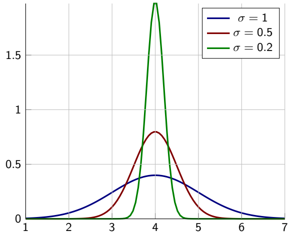
- Look similar but not exactly the same, the red terms are different

- We want to understand better the relationship between

$$\mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) q_{\pi_\theta}(s,a) \mid s\right] \quad \text{and} \quad \nabla_\theta \mu_\theta(s) \nabla_a q_{\mu_\theta}(s,a)\Big|_{a=\mu(s)}$$

- Let us consider for simplicity Gaussian policies

$$\pi_\theta(a|s) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(a-\mu_\theta(s))^2/(2\sigma^2)}$$



- We can think of a deterministic policy as a gaussian with $\sigma = 0$
- More formally as a δ distribution

- A δ is an operator defined as $\displaystyle\int f(x)\delta(x)\,dx = f(0)$

- So let us consider a stochastic policy and then take $\sigma \to 0$
- For a Gaussian distribution

$$\pi_\theta(a|s) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(a-\mu_\theta(s))^2/(2\sigma^2)}$$

- The gradient of the log yields

$$\nabla_\theta \log \pi_\theta(a|s) = \nabla_\theta \left(-(a-\mu_\theta(s))^2/(2\sigma^2)\right) = \frac{a-\mu_\theta(s)}{\sigma^2}\nabla_\theta \mu_\theta(s)$$

- Recall that we are looking at the following two terms

$$\mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) q_{\pi_\theta}(s,a) \mid s\right] \quad \text{and} \quad \nabla_\theta \mu_\theta(s) \nabla_a q_{\mu_\theta}(s,a)\Big|_{a=\mu_\theta(s)}$$

- Replacing $\nabla_\theta \log \pi_\theta(a|s)$ in the first expression yields

$$\mathbb{E}_{a\sim\pi_\theta}\left[\frac{a-\mu_\theta(s)}{\sigma^2} q_{\pi_\theta}(s,a) \mid s\right] \nabla_\theta \mu_\theta(s)$$

- We reduce the analysis of the two expressions to compare

$$\mathbb{E}_{a\sim\pi_\theta}\left[\frac{a-\mu_\theta(s)}{\sigma^2}q_{\pi_\theta}(s,a)\mid s\right] \quad \text{and} \quad \nabla_a q_{\mu_\theta}(s,a)\Big|_{a=\mu_\theta(s)}$$

- We are interpreting deterministic policies as the limit of a Gaussians

$$\lim_{\sigma\to 0}\mathbb{E}_{a\sim\pi_\theta}\left[\frac{a-\mu_\theta(s)}{\sigma^2}q_{\pi_\theta}(s,a)\mid s\right] =$$

$$\lim_{\sigma\to 0}\int_{\mathcal{A}}\frac{a-\mu_\theta(s)}{\sigma^2}q_{\pi_\theta}(s,a)\frac{e^{-(a-\mu_\theta(s))^2/(2\sigma^2)}}{\sqrt{2\pi\sigma^2}}\,da$$

- Let us define $\eta = a-\mu_\theta(s)$ and define

$$\lim_{\sigma\to 0}\mathbb{E}_{a\sim\pi_\theta}\left[\frac{a-\mu_\theta(s)}{\sigma^2}q_{\pi_\theta}(s,a)\mid s\right] =$$

$$\lim_{\sigma\to 0}\int\frac{\eta}{\sigma^2}q_{\pi_\theta}(s,\eta+\mu_\theta(s))\frac{e^{-(\eta)^2/(2\sigma^2)}}{\sqrt{2\pi\sigma^2}}\,d\eta$$

- We are now comparing

$$\lim_{\sigma\to 0} \mathbb{E}_{a\sim\pi_\theta}\left[\frac{a-\mu_\theta(s)}{\sigma^2} q_{\pi_\theta}(s,a) \mid s\right] \quad \text{and} \quad \nabla_a q_{\mu_\theta}(s,a)\Big|_{a=\mu_\theta(s)}$$

- From the previous slide we had that

$$\lim_{\sigma\to 0} \mathbb{E}_{a\sim\pi_\theta}\left[\frac{a-\mu_\theta(s)}{\sigma^2} q_{\pi_\theta}(s,a) \mid s\right] =$$
$$\lim_{\sigma\to 0} \int \frac{\eta}{\sigma^2} q_{\pi_\theta}(s,\eta+\mu_\theta(s)) \frac{e^{-(\eta)^2/(2\sigma^2)}}{\sqrt{2\pi\sigma^2}}\, d\eta$$

- Define $\phi(\eta) = \frac{e^{-(\eta)^2/(2\sigma^2)}}{\sqrt{2\pi\sigma^2}}$ and notice that $\nabla_\eta \phi(\eta) = -\frac{\eta}{\sigma^2}\phi(\eta)$
- Integrate by parts

$$\lim_{\sigma\to 0} \mathbb{E}_{a\sim\pi_\theta}\left[\frac{a-\mu_\theta(s)}{\sigma^2} q_{\pi_\theta}(s,a) \mid s\right] = \lim_{\sigma\to 0} -\phi(\eta) q(s,\eta+\mu_\theta(s))\Big|_{-\infty}^{\infty}$$
$$+ \int \phi(\eta) \nabla_a q_{\pi_\theta}(s,a)\Big|_{a=\eta+\mu_\theta(s)}\, d\eta$$

- Recall that we are looking at the following two terms

$$\lim_{\sigma \to 0} \mathbb{E}_{a \sim \pi_\theta} \left[\frac{a - \mu_\theta(s)}{\sigma^2} q_{\pi_\theta}(s, a) \mid s \right] \quad \text{and} \quad \nabla_a q_{\mu_\theta}(s, a)\big|_{a=\mu_\theta(s)}$$

- From the previous slide we had that

$$\lim_{\sigma \to 0} \mathbb{E}_{a \sim \pi_\theta} \left[\frac{a - \mu_\theta(s)}{\sigma^2} q_{\pi_\theta}(s, a) \mid s \right] = \lim_{\sigma \to 0} -\phi(\eta) q(s, \eta + \mu_\theta(s))\big|_{-\infty}^{\infty} + \int \phi(\eta) \nabla_a q_{\pi_\theta}(s, a)\big|_{a=\eta+\mu_\theta(s)} d\eta$$

- $\lim_{\eta \to \infty} \eta(\eta) = 0$ and q is bounded
- The gaussian converges to the δ so the previous integral is

$$\lim_{\sigma \to 0} \mathbb{E}_{a \sim \pi_\theta} \left[\frac{a - \mu_\theta(s)}{\sigma^2} q_{\pi_\theta}(s, a) \mid s \right] = \nabla_a q_{\pi_\theta}(s, a)\big|_{a=\mu_\theta(s)}$$

- In summary the deterministic policy gradient

$$\nabla_\theta v(\theta) = (1-\gamma)^{-1}\mathbb{E}_{s\sim\rho_{\mu_\theta}}\left[\nabla_\theta \mu_\theta(s)\nabla_a q_{\mu_\theta}(s,a)\Big|_{a=\mu(s)}\right]$$

- Can be understood as the limit of the stochastic policy gradient

$$\nabla_\theta v(\theta) = (1-\gamma)^{-1}\mathbb{E}_{s\sim\rho_\theta, a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) q_{\pi_\theta}(s,a)\right]$$

- How can we get the estimate of the gradient?
  - $\Rightarrow$ There is an expectation $\Rightarrow$ Stochastic Approximations
  - $\Rightarrow$ We would still need to compute $\nabla_a q_{\mu_\theta}(s,a)\Big|_{a=\mu_\theta(s)}$
  - $\Rightarrow$ Learn q_{μ_θ} using function approximations
  - $\Rightarrow$ Computing the derivative of q with respect to a is easy
  - $\Rightarrow$ Use Off policy Actor-Critic to ensure proper exploration

- As we did before define the cost for the behavior policy $b(A|S)$

$$J_b(\mu_\theta) = \int_{\mathcal{S}} \rho_b(s) v_{\mu_\theta}(s)\, ds = \int_{\mathcal{S}} \rho_b(s) q_{\mu_\theta}(s, \mu_\theta(s))\, ds$$

- Taking the gradient it follows that

$$\nabla_\theta J_b(\mu_\theta) = \int_{\mathcal{S}\times\mathcal{A}} \rho_b(s) \nabla_\theta \mu_\theta(a|s) q_{\mu_\theta}(s, a)\, dsda$$

$$+ \int_{\mathcal{S}\times\mathcal{A}} \rho_b(s) \nabla_\theta \mu_\theta(a|s) q_{\mu_\theta}(s, a)\, dsda$$

- Because the policy is deterministic the expression yields

$$\nabla_\theta J_b(\mu_\theta) \approx \int_{\mathcal{S}} \rho_b(s) \nabla_\theta \mu_\theta(s) \nabla_a q_{\mu_\theta}(s, a)\Big|_{a=\mu_\theta(s)}\, ds$$

- We don't need the impotance sampling