Chapter 13: Policy Gradient Methods

Seungjae Ryan Lee
Preview: Policy Gradients

● **Action-value Methods**
  ○ Learn values of actions and select actions with estimated action values
  ○ Policy derived from action-value estimates

● **Policy Gradient Methods**
  ○ Learn parameterized policy that can select action without a value function
  ○ Can still use value function to *learn* the policy parameter
Policy Gradient Methods

- Define a performance measure $J(\theta)$ to maximize
- Learn policy parameter $\theta$ through *approximate gradient ascent*

$$\theta_{t+1} = \theta_t + \alpha \nabla J(\theta_t)$$

Stochastic estimate of $J(\theta)$
Soft-max in Action Preferences

- Numerical preference $h(s, a, \theta)$ for each state-action pair
- Action selection through soft-max

$$\pi(a|s, \theta) = \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,b,\theta)}}$$
Soft-max in Action Preferences: Advantages

1. Approximate policy can approach deterministic policy
   - No “limit” like ε-greedy methods
   - Using soft-max on action values cannot approach deterministic policy
Soft-max in Action Preferences: Advantages

2. ** Allow stochastic policy  
   ○ Best approximate policy can be stochastic in problems with significant function approximation

- Consider small corridor with -1 reward on each step  
  ○ States are indistinguishable  
  ○ Action transition is reversed in the second state
Soft-max in Action Preferences: Advantages

2. **Allow stochastic policy**
   - $\epsilon$-greedy methods ($\epsilon=0.1$) cannot find optimal policy

$$J(\theta) = v_{\pi_\theta}(S)$$

- $\epsilon$-greedy left
- $\epsilon$-greedy right
- Optimal stochastic policy
- Probability of right action

Soft-max in Action Preferences: Advantages

3. Policy may be simpler to approximate
   - Differs among problems

Theoretical Advantage of Policy Gradient Methods

- Smooth transition of policy for parameter changes
- Allows for stronger convergence guarantees

![Diagram showing smooth transition and values for Q and h]
Policy Gradient Theorem

- Define performance measure as value of the start state

\[ J(\theta) \equiv v_{\pi_\theta}(s_0) \]

- Want to compute \( \nabla J(\theta) \) w.r.t. policy parameter \( \theta \)
Policy Gradient Theorem

\[ \nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s, a) \nabla \pi(a|s, \theta) \]

On-policy state distribution

Episodic: Average episode length
Continuing: 1
Policy Gradient Theorem: Proof

\[ \nabla \nu_\pi(s) \]

\[ \nabla \left[ \sum_a \pi(a|s)q_\pi(s,a) \right] \]
Policy Gradient Theorem: Proof

\[ \nabla \left[ \sum_a \pi(a|s)q_\pi(s, a) \right] \]

\[ \sum_a \left[ \nabla \pi(a|s)q_\pi(s, a) + \pi(a|s)\nabla q_\pi(s, a) \right] \]
Policy Gradient Theorem: Proof

\[ \sum_a \left[ \nabla \pi(a|s)q_\pi(s, a) + \pi(a|s)\nabla q_\pi(s, a) \right] \]

\[ \sum_a \left[ \nabla \pi(a|s)q_\pi(s, a) + \pi(a|s)\nabla \sum_{s', r} p(s', r|s, a)(r + v_\pi(s')) \right] \]
Policy Gradient Theorem: Proof

\[
\sum_a \left[ \nabla \pi(a|s)q_\pi(s, a) + \pi(a|s) \nabla \sum_{s',r} p(s', r|s, a) (r + v_\pi(s')) \right]
\]

\[\nabla r = 0\]

\[
\sum_a \left[ \nabla \pi(a|s)q_\pi(s, a) + \pi(a|s) \sum_{s'} p(s'|s, a) \nabla v_\pi(s') \right]
\]
Policy Gradient Theorem: Proof

\[
\sum_a \left[ \nabla \pi(a|s) q_\pi(s, a) + \pi(a|s) \sum_{s'} p(s' | s, a) \nabla v_\pi(s') \right]
\]

Unrolling

\[
\sum_a \left[ \nabla \pi(a|s) q_\pi(s, a) + \pi(a|s) \sum_{s'} p(s' | s, a) \sum_{a'} \left[ \nabla \pi(a'|s') q_\pi(s', a') + \pi(a'|s') \sum_{s''} p(s'' | s', a') \nabla v_\pi(s'') \right] \right]
\]
Policy Gradient Theorem: Proof

\[
\sum_a \left[ \nabla \pi(a|s) q_\pi(s, a) + \pi(a|s) \sum_{s'} p(s'|s, a) \sum_{a'} \left[ \nabla \pi(a'|s') q_\pi(s', a') + \pi(a'|s') \sum_{s''} p(s''|s', a') \nabla v_\pi(s'') \right] \right] = \sum_{x \in S} \sum_{k=0} \Pr(s \to x, k, \pi) \sum_a \nabla \pi(a|x) q_\pi(x, a) \nabla v_\pi(s'')
\]
Policy Gradient Theorem: Proof

\[
\sum_{x \in S} \sum_{k=0}^{\infty} \Pr(s \to x, k, \pi) \sum_a \nabla \pi(a \mid x) q_\pi(x, a)
\]

\[
\sum_s \eta(s) \sum_a \nabla \pi(a \mid s) q_\pi(s, a)
\]
Policy Gradient Theorem: Proof

\[ \sum_s \eta(s) \sum_a \nabla \pi(a|s) q_\pi(s, a) \]

\[ \sum_{s'} \eta(s') \sum_s \frac{\eta(s)}{\sum_{s'} \eta(s')} \sum_a \nabla \pi(a|s) q_\pi(s, a) \]
Policy Gradient Theorem: Proof

\[
\sum_{s'} \eta(s') \sum_s \frac{\eta(s)}{\sum_{s'} \eta(s')} \sum_a \nabla \pi(a|s) q_\pi(s, a)
\]

\[
\sum_{s'} \eta(s') \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s, a)
\]
Policy Gradient Theorem: Proof

\[ \nabla v_\pi(s) = \sum_{s'} \eta(s') \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s, a) \]

\[ \sum_{s'} \eta(s') \text{ is a constant} \]

\[ \nabla v_\pi(s) \propto \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s, a) \]
Stochastic Gradient Descent

- Need samples with expectation $\nabla J(\theta)$

\[
\nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s, a) \nabla \pi(a | s, \theta)
\]
Stochastic Gradient Descent

$$\sum_s \mu(s) \sum_a q_\pi(s, a) \nabla \pi(a \mid s, \theta)$$

$$\mu$$ is an on-policy state distribution of $$\pi$$

$$\mathbb{E}_\pi \left[ \sum_a q_\pi(S_t, a) \nabla \pi(a \mid S_t, \theta) \right]$$
Stochastic Gradient Descent

\[
\mathbb{E}_{\pi} \left[ \sum_{a} q_{\pi}(S_t, a) \nabla \pi(a|S_t, \theta) \right]
\]

\[
\mathbb{E}_{\pi} \left[ \sum_{a} \pi(a|S_t, \theta) q_{\pi}(S_t, a) \frac{\nabla \pi(a|S_t, \theta)}{\pi(a|S_t, \theta)} \right]
\]
Stochastic Gradient Descent

\[
\mathbb{E}_{\pi} \left[ \sum_a \pi(a|S_t, \theta)q_{\pi}(S_t, a) \frac{\nabla \pi(a|S_t, \theta)}{\pi(a|S_t, \theta)} \right]
\]

Replace \( a \) with sample \( A_t \sim \pi \)

\[
\mathbb{E}_{\pi} \left[ q_{\pi}(S_t, A_t) \frac{\nabla \pi(A_t|S_t, \theta)}{\pi(A_t|S_t, \theta)} \right]
\]
Stochastic Gradient Descent: REINFORCE

\[ \mathbb{E}_\pi \left[ q_\pi(S_t, A_t) \frac{\nabla\pi(A_t | S_t, \theta)}{\pi(A_t | S_t, \theta)} \right] \]

\[ \mathbb{E}_\pi \left[ G_t \frac{\nabla\pi(A_t | S_t, \theta)}{\pi(A_t | S_t, \theta)} \right] \]
REINFORCE (1992)

- Sample return like Monte Carlo
- Increment proportional to return
- Increment inverse proportional to action probability
  - Prevent frequent actions dominating due to frequent updates

\[ \theta_{t+1} = \theta_t + \alpha G_t \frac{\nabla \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)}. \]
REINFORCE: Pseudocode

**REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for** $\pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \theta)$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\theta \in \mathbb{R}^{d'}$ (e.g., to 0)

Loop forever (for each episode):
- Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \theta)$
- Loop for each step of the episode $t = 0, 1, \ldots, T - 1$:
  - $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$  \hspace{1cm} (G_t)
  - $\theta \leftarrow \theta + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \theta)$  \hspace{1cm} Eligibility vector
REINFORCE: Results

$G_0$
Total reward on episode averaged over 100 runs

$\alpha = 2^{-13}$

$\alpha = 2^{-14}$

$\alpha = 2^{-12}$

$v_*(s_0)$

Episode

1 200 400 600 800 1000
REINFORCE with Baseline

- REINFORCE
  - Good theoretical convergence
  - Bad convergence speed due to high variance

\[ \nabla J(\theta) \propto \sum_s \mu(s) \sum_a \left( q_\pi(s, a) - b(s) \right) \nabla \pi(a|s, \theta) \]

\[ \theta_{t+1} = \theta_t + \alpha \left( G_t - b(S_t) \right) \frac{\nabla \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)} \]
## REINFORCE with Baseline (episodic), for estimating $\pi_\theta \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \theta)$
Input: a differentiable state-value function parameterization $\hat{v}(s, w)$
Algorithm parameters: step sizes $\alpha^\theta > 0$, $\alpha^w > 0$
Initialize policy parameter $\theta \in \mathbb{R}^{d'}$ and state-value weights $w \in \mathbb{R}^d$ (e.g., to $0$)

Loop forever (for each episode):
  - Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \theta)$
  - Loop for each step of the episode $t = 0, 1, \ldots, T - 1$:
    \[
    \begin{align*}
    G &\leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k \\
    \delta &\leftarrow G - \hat{v}(S_t, w) \\
    w &\leftarrow w + \alpha^w \delta \nabla \hat{v}(S_t, w) \\
    \theta &\leftarrow \theta + \alpha^\theta \gamma^t \delta \nabla \ln \pi(A_t|S_t, \theta)
    \end{align*}
    \]
REINFORCE with Baseline: Results

\[ G_0 \]
Total reward on episode averaged over 100 runs

REINFORCE with baseline \[ \alpha^g = 2^{-9}, \alpha^w = 2^{-6} \]

REINFORCE \[ \alpha = 2^{-13} \]

Episodes:

- S
- G
Actor-Critic Methods

- Learn approximations for both policy (Actor) and value function (Critic)
- Critic vs Baseline in REINFORCE
  - Critic is used for bootstrapping
  - Bootstrapping introduces bias and relies on state representation
  - Bootstrapping reduces variance and accelerates learning

\[
\pi(A, S, \theta) \quad \hat{v}(S, w)
\]

Actor \quad \text{Critic}
One-step Actor Critic

- Replace return with one-step return
- Replace baseline with approximated value function (Critic)
  - Learned with semi-gradient TD(0)

REINFORCE:
\[
\theta_{t+1} = \theta_t + \alpha \left( G_t - b(S_t) \right) \frac{\nabla \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)}.
\]

One-step AC:
\[
\theta_{t+1} = \theta_t + \alpha \left( G_{t:t+1} - \hat{v}(S_t, w) \right) \frac{\nabla \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)}
= \theta_t + \alpha \left( R_{t+1} + \gamma \hat{v}(S_{t+1}, w) - \hat{v}(S_t, w) \right) \frac{\nabla \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)}
= \theta_t + \alpha \delta_t \frac{\nabla \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)}.
\]
One-step Actor–Critic (episodic), for estimating $\pi_\theta \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \theta)$
Input: a differentiable state-value function parameterization $\hat{v}(s, w)$
Parameters: step sizes $\alpha^\theta > 0$, $\alpha^w > 0$
Initialize policy parameter $\theta \in \mathbb{R}^{d'}$ and state-value weights $w \in \mathbb{R}^d$ (e.g., to 0)
Loop forever (for each episode):
  Initialize $S$ (first state of episode)
  $I \leftarrow 1$
  Loop while $S$ is not terminal (for each time step):
    $A \sim \pi(\cdot|S, \theta)$
    Take action $A$, observe $S', R$
    $\delta \leftarrow R + \gamma \hat{v}(S', w) - \hat{v}(S, w)$
    (if $S'$ is terminal, then $\hat{v}(S', w) \equiv 0$)
    $w \leftarrow w + \alpha^w \delta \nabla \hat{v}(S, w)$
    $\theta \leftarrow \theta + \alpha^\theta I \delta \nabla \ln \pi(A|S, \theta)$
    $I \leftarrow \gamma I$
    $S \leftarrow S'$

Update Critic (value function) parameters
Update Actor (policy) parameters
Actor-Critic with Eligibility Traces

<table>
<thead>
<tr>
<th>Actor–Critic with Eligibility Traces (episodic), for estimating $\pi_\theta \approx \pi_*$</th>
</tr>
</thead>
</table>
| **Input:** a differentiable policy parameterization $\pi(a|s, \theta)$  
**Input:** a differentiable state-value function parameterization $\hat{v}(s, w)$  
**Parameters:** trace-decay rates $\lambda^\theta \in [0, 1]$, $\lambda^w \in [0, 1]$; step sizes $\alpha^\theta > 0$, $\alpha^w > 0$  
**Initialize** policy parameter $\theta \in \mathbb{R}^d$ and state-value weights $w \in \mathbb{R}^d$ (e.g., to 0)  
**Loop** forever (for each episode):  
  - **Initialize** $S$ (first state of episode)  
    - $z^\theta \leftarrow 0$ ($d'$-component eligibility trace vector)  
    - $z^w \leftarrow 0$ ($d$-component eligibility trace vector)  
    - $I \leftarrow 1$  
  - **Loop** while $S$ is not terminal (for each time step):  
    - $A \sim \pi(\cdot|S, \theta)$  
    - Take action $A$, observe $S', R$  
    - $\delta \leftarrow R + \gamma \hat{v}(S', w) - \hat{v}(S, w)$  
      (if $S'$ is terminal, then $\hat{v}(S', w) = 0$)  
    - $z^w \leftarrow \gamma \lambda^w z^w + \nabla \hat{v}(S, w)$  
    - $z^\theta \leftarrow \gamma \lambda^\theta z^\theta + I \nabla \ln \pi(A|S, \theta)$  
    - $w \leftarrow w + \alpha^w \delta z^w$  
    - $\theta \leftarrow \theta + \alpha^\theta \delta z^\theta$  
    - $I \leftarrow \gamma I$  
    - $S \leftarrow S'$
Average Reward for Continuing Problems

- Measure performance in terms of average reward

\[
J(\theta) \equiv r(\pi) = \lim_{h \to \infty} \frac{1}{h} \sum_{t=1}^{h} \mathbb{E}[R_t \mid S_0, A_{0:t-1} \sim \pi]
\]

\[
= \lim_{t \to \infty} \mathbb{E}[R_t \mid S_0, A_{0:t-1} \sim \pi]
\]

\[
= \sum_s \mu(s) \sum_a \pi(a \mid s) \sum_{s',r} p(s',r \mid s,a)r,
\]
Actor-Critic for Continuing Problems

Actor–Critic with Eligibility Traces (continuing), for estimating $\pi_\theta \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \theta)$
Input: a differentiable state-value function parameterization $\hat{v}(s, w)$
Algorithm parameters: $\lambda^w \in [0, 1]$, $\lambda^\theta \in [0, 1]$, $\alpha^w > 0$, $\alpha^\theta > 0$, $\alpha^R > 0$

Initialize $R \in \mathbb{R}$ (e.g., to 0)

Initialize state-value weights $w \in \mathbb{R}^d$ and policy parameter $\theta \in \mathbb{R}^{d'}$ (e.g., to 0)
Initialize $S \in S$ (e.g., to $s_0$)

$z^w \leftarrow 0$ (d-component eligibility trace vector)
$z^\theta \leftarrow 0$ (d'-component eligibility trace vector)

Loop forever (for each time step):

$A \sim \pi(\cdot|S, \theta)$
Take action $A$, observe $S', R$

$\delta \leftarrow R - \hat{R} + \hat{v}(S', w) - \hat{v}(S, w)$
$\bar{R} \leftarrow R + \alpha^R \delta$

$z^w \leftarrow \lambda^w z^w + \nabla \hat{v}(S, w)$
$z^\theta \leftarrow \lambda^\theta z^\theta + \nabla \ln \pi(A|S, \theta)$

$w \leftarrow w + \alpha^w \delta z^w$
$\theta \leftarrow \theta + \alpha^\theta \delta z^\theta$
$S \leftarrow S'$

endtoend.ai
Policy Gradient Theorem Proof (Continuing Case)

1. **Same procedure:**

   \[
   \nabla v_\pi(s) = \nabla \left[ \sum_a \pi(a|s)q_\pi(s,a) \right] \\
   = \sum_a \left[ \nabla \pi(a|s)q_\pi(s,a) + \pi(a|s)\nabla q_\pi(s,a) \right] \\
   = \sum_a \left[ \nabla \pi(a|s)q_\pi(s,a) + \pi(a|s)\nabla \sum_{s',r} p(s'|s,a)(r - r(\theta) + v_\pi(s')) \right] \\
   = \sum_a \left[ \nabla \pi(a|s)q_\pi(s,a) + \pi(a|s)[-\nabla r(\theta) + \sum_{s'} p(s'|s,a)\nabla v_\pi(s')] \right]
   \]

2. **Rearrange equation:**

   \[
   \nabla r(\theta) = \sum_a \left[ \nabla \pi(a|s)q_\pi(s,a) + \pi(a|s)\sum_{s'} p(s'|s,a)\nabla v_\pi(s') \right] - \nabla v_\pi(s). \\
   \downarrow \\
   \nabla J(\theta)
   \]
Policy Gradient Theorem Proof (Continuing Case)

$$\nabla J(\theta) = \sum_a \left[ \nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) \nabla v_\pi(s') \right] - \nabla v_\pi(s).$$

3. Sum over all states weighted by state-distribution $\mu(s)$
   a. Nothing changes since neither side depend on $s$ and $\sum_s \mu(s) = 1$

$$\nabla J(\theta) = \sum_s \mu(s) \left( \sum_a \left[ \nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) \nabla v_\pi(s') \right] - \nabla v_\pi(s) \right)$$

$$= \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s,a)$$

$$+ \sum_s \mu(s) \sum_a \pi(a|s) \sum_{s'} p(s'|s,a) \nabla v_\pi(s') - \sum_s \mu(s) \nabla v_\pi(s)$$

$$= \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s,a)$$

$$+ \sum_{s'} \sum_s \mu(s) \sum_a \pi(a|s) p(s'|s,a) \nabla v_\pi(s') - \sum_s \mu(s) \nabla v_\pi(s)$$
Policy Gradient Theorem Proof (Continuing Case)

\[ \nabla J(\theta) = \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s, a) \]
\[ + \sum_{s'} \sum_s \mu(s) \sum_a \pi(a|s) p(s'|s, a) \nabla v_\pi(s') - \sum_s \mu(s) \nabla v_\pi(s) \]

4. Use ergodicity: \[ \sum_s \mu(s) \sum_a \pi(a|s, \theta)p(s'|s, a) = \mu(s'), \text{ for all } s' \in S. \]

\[ \nabla J(\theta) = \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s, a) + \sum_{s'} \mu(s') \nabla v_\pi(s') - \sum_s \mu(s) \nabla v_\pi(s) \]
\[ = \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s, a). \]

Q.E.D.
Policy Parameterization for Continuous Actions

- Policy based methods can handle **continuous action spaces**
- Learn statistics of the probability distribution
  - ex) mean and variance of Gaussian
- Choose action from the learned distribution
  - ex) Gaussian distribution

\[
p(x) \equiv \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)
\]

\[
\pi(a|s, \theta) \equiv \frac{1}{\sigma(s, \theta) \sqrt{2\pi}} \exp\left(-\frac{(a - \mu(s, \theta))^2}{2\sigma(s, \theta)^2}\right)
\]
Policy Parametrization to Gaussian Distribution

- Divide policy parameter vector into mean and variance: \( \theta = [\theta_\mu, \theta_\sigma]^T \)
- Approximate mean with linear function:
  \[
  \mu(s, \theta) \doteq \theta_\mu^T x_\mu(s)
  \]
- Approximate variance with exponential of linear function:
  - Guaranteed positive
  \[
  \sigma(s, \theta) \doteq \exp\left( \theta_\sigma^T x_\sigma(s) \right)
  \]
- All PG algorithms can be applied to the parameter vector
Summary

- **Policy Gradient methods** have many advantages over action-value methods
  - Represent stochastic policy and approach deterministic policies
  - Learn appropriate levels of exploration
  - Handle continuous action spaces
  - Compute effect of policy parameter on performance with Policy Gradient Theorem

- **Actor-Critic** estimates value function for bootstrapping
  - Introduces bias but is often desirable due to lower variance
  - Similar to preferring TD over MC

“Policy Gradient methods provide a significantly different set of strengths and weaknesses than action-value methods.”
Thank you!

Original content from

- [Reinforcement Learning: An Introduction by Sutton and Barto](#)

You can find more content in

- [github.com/seungjaeryanlee](#)
- [www.endtoend.ai](#)