Policy-Gradient Methods

- Policy-Gradient techniques attempt at **direct optimization of expected return**
  \[ \mathbb{E}_{\pi_\theta}[G_t] \]
  for **parameterized stochastic policy**
  \[ \pi_\theta(a|s) = P[A_t = a|S_t = s, \theta]. \]

- Policy-function is also called **actor**.
- We will discuss **actor-only** (optimize parametric policy) and **actor-critic** (learn both policy and critic parameters in tandem) methods.
**One-Step MDPs/Gradient Bandits**

Let $p_\theta(y)$ denote probability of an action/output, $\Delta(y)$ be the reward/quality of an output.

**Objective:** $\mathbb{E}_{p_\theta}[\Delta(y)]$

**Gradient:**

\[ \nabla_\theta \mathbb{E}_{p_\theta}[\Delta(y)] = \nabla_\theta \sum_y p_\theta(y) \Delta(y) \]

\[ = \sum_y \nabla_\theta p_\theta(y) \Delta(y) \]

\[ = \sum_y \frac{p_\theta(y)}{p_\theta(y)} \nabla_\theta p_\theta(y) \Delta(y) \]

\[ = \sum_y p_\theta(y) \nabla_\theta \log p_\theta(y) \Delta(y) \]

\[ = \mathbb{E}_{p_\theta}[\Delta(y) \nabla_\theta \log p_\theta(y)]. \]
Score Function Gradient Estimator for Bandit

- **Bandit Gradient Ascent:**
  - Sample \( y_i \sim p_\theta \),
  - Update \( \theta \leftarrow \theta + \alpha (\Delta(y_i) \nabla_\theta \log p_\theta(y_i)) \).

- Update by stochastic gradient \( g_i = \Delta(y_i) \nabla_\theta \log p_\theta(y_i) \) yields unbiased estimator of \( \mathbb{E}_{p_\theta}[\Delta(y)] \).

- Intuition: \( \nabla_\theta \log p_\theta(y) \) is called the **score function**.
  - Moving in the direction of \( g_i \) pushes up the score of the sample \( y_i \) in proportion to its reward \( \Delta(y_i) \).
  - In RL terms: High reward samples are weighted higher - *reinforced!*
  - Estimator is valid even if \( \Delta(y) \) is non-differentiable.
Score Function Gradient Estimator for MDPs

Let $y = S_0, A_0, R_1, \ldots, R_T \sim \pi_\theta$ be an episode, and $R(y) = R_1 + \gamma R_2 + \ldots + \gamma^{T-1} R_T = \sum_{t=1}^{T} \gamma^{t-1} R_t$ be its total discounted reward.

- **Objective:** $\mathbb{E}_{\pi_\theta} [R(y)]$.
- **Gradient:** $\mathbb{E}_{\pi_\theta} [R(y) \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta (A_t | S_t)]$.

**Reinforcement Gradient Ascent:**

- Sample episode $y = S_0, A_0, R_1, \ldots, R_T \sim \pi_\theta$,
- Obtain reward $R(y) = \sum_{t=1}^{T} \gamma^{t-1} R_t$,
- Update $\theta \leftarrow \theta + \alpha (R(y) \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta (A_t | S_t))$. 

---

Reinforcement Learning, Winter 2017/19 32(40)
General Form of Policy Gradient Algorithms

Formalized for expected per time-step reward with respect to action-value $q_{\pi_\theta}(S_t, A_t)$.

- **Objective:** $E_{\pi_\theta} [q_{\pi_\theta}(S_t, A_t)]$.
- **Gradient:** $E_{\pi_\theta} [q_{\pi_\theta}(S_t, A_t) \nabla_\theta \log \pi_\theta(A_t|S_t)]$.

**Policy Gradient Ascent:**
- Sample episode $y = S_0, A_0, R_1, \ldots, R_T \sim \pi_\theta$.
- For each time step $t$:
  - Obtain reward $q_{\pi_\theta}(S_t, A_t)$,
  - Update $\theta \leftarrow \theta + \alpha (q_{\pi_\theta}(S_t, A_t) \nabla_\theta \log \pi_\theta(A_t|S_t))$. 
Policy Gradient Algorithms

- General form for expected per time-step return $q_{\pi_\theta}(S_t, A_t)$ is known as Policy Gradient Theorem [Sutton et al., 2000].
- Since $q_{\pi_\theta}(s, a)$ is normally not known, one can use the actual discounted return $G_t$ at time step $t$, calculated from sampled episode. This leads to the REINFORCE algorithm [Williams, 1992].
- Problems of Policy Gradient Algorithms, esp. REINFORCE:
  - Large variance in discounted returns calculated from sampled episodes.
  - Each gradient update is done independently of past gradient estimates.
Variance Reduction by Baselines

- Variance of REINFORCE can be reduced by comparison of actual return $G_t$ to a baseline $b(s)$ for state $s$ that is constant with respect to actions $a$. Example: average return so far.
- Update:

$$\theta \leftarrow \theta + \alpha (G_t - b(S_t)) \nabla_{\theta} \log \pi_{\theta}(A_t \mid S_t).$$

- Can be interpreted as **Control Variate** [Ross, 2013]:
  - Goal is to augment random variable $X$ (= stochastic gradient) with highly correlated variable $Y$ such that $\text{Var}(X - Y) = \text{Var}(X) + \text{Var}(Y) - 2\text{Cov}(X, Y)$ is reduced.
  - Gradient remains unbiased since $\mathbb{E}[X - Y + \mathbb{E}[Y]] = \mathbb{E}[X]$.

Exercise: Show that $\mathbb{E}[Y] = 0$ for constant baselines.
Actor-Critic Methods

- Learning a critic in order to get an improved estimate of the expected return will also reduce variance.
  - **Critic:** TD(0) update for linear approximation
    \[ q_{\pi_\theta}(s, a) \approx q_w(s, a) = \phi(s, a)^\top w. \]
  - **Actor:** Policy gradient update reinforced by \( q_w(s, a) \).

- **Simple Actor-Critic** [Konda and Tsitsiklis, 2000]:
  - Sample \( a \sim \pi_\theta \).
  - For each step \( t \):
    - Sample reward \( r \sim R_s \), transition \( s' \sim P_{s, a} \), action \( a' \sim \pi_\theta(s', \cdot) \),
    - \( \delta \leftarrow r + \gamma q_w(s', a') - q_w(s, a) \),
    - \( \theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a|s) q_w(s, a) \),
    - \( w \leftarrow w + \beta \delta \phi(s, a) \),
    - \( a \leftarrow a', s \leftarrow s' \).

**Exercise:** What is the difference between REINFORCE and Actor-Critic in terms of number of updates per step?
Bias and Compatible Function Approximation

- Approximating $q_{\pi_\theta}(s, a) \approx q_w(s, a)$ introduces bias. Unless
  1. Value approximator is compatible with the policy, i.e., the change in value equals the score function s.t.

$$\nabla_w q_w(s, a) = \nabla \theta \log \pi_\theta(s, a),$$

  2. Parameters $w$ are set to minimize the squared error

$$\epsilon = \mathbb{E}_{\pi_\theta}[(q_{\pi_\theta}(s, a) - q_w(s, a))^2],$$

- Then policy gradient is exact:

$$\mathbb{E}_{\pi_\theta}[q_{\pi_\theta}(s, a) \nabla \theta \log \pi_\theta(a|s)] = \mathbb{E}_{\pi_\theta}[q_w(s, a) \nabla \theta \log \pi_\theta(a|s)].$$

Exercise: Prove the compatible function approximation property!
Advantage Actor-Critic

- Combine idea of baseline with actor-critic by using **advantage function** that compares action-value function $q_{\pi \theta}(s, a)$ to state-value function $v_{\pi \theta}(s) = \mathbb{E}_{a \sim \pi}[q_{\pi \theta}(s, a)]$.
- Use approximate TD error
  \[
  \delta_w = r + \gamma v_w(s') - v_w(s),
  \]
  where state-value is approximated by $v_w(s)$, and action-value is approximated by sample $q_w(s') = r + \gamma v_w(s')$.
- Update Actor: $\theta \leftarrow \theta + \alpha \nabla_{\theta} \log \pi_{\theta}(a|s)(q_w(s') - v_w(s))$.
- Update Critic: $w = \text{arg min}_w (q_w(s') - v_w(s))^2$. 

Summary: Policy-Gradient Methods

- Build upon huge knowledge in stochastic optimization which provides excellent theoretical understanding of convergence properties.
- Gradient-based techniques are model-free since MDP transition matrix is not dependent on $\theta$.
- Directly applicable to continuous output spaces and stochastic policies.
- Problem of high variance in actor-only methods can be mitigated by the critic’s low-variance estimate of expected return.
Overall Summary and Outlook

What have we covered:

- **Policy evaluation** (a.k.a. prediction) using **DP**

- **Policy optimization** (a.k.a. control) using **Value-based** techniques of **DP**, **MC**, or both: **TD**.

- **Policy-gradient** techniques for direct stochastic optimization of parametric policies.

What did we leave out:

- Proofs: See Bertsekas & Tsitsiklis and papers on reading list.

- Subleties of exploration/exploitation (selecting random start states in MC vs. random actions in PG), on/off policy learning (SARSA vs. Q-learning),...

- See papers on reading list.
References


