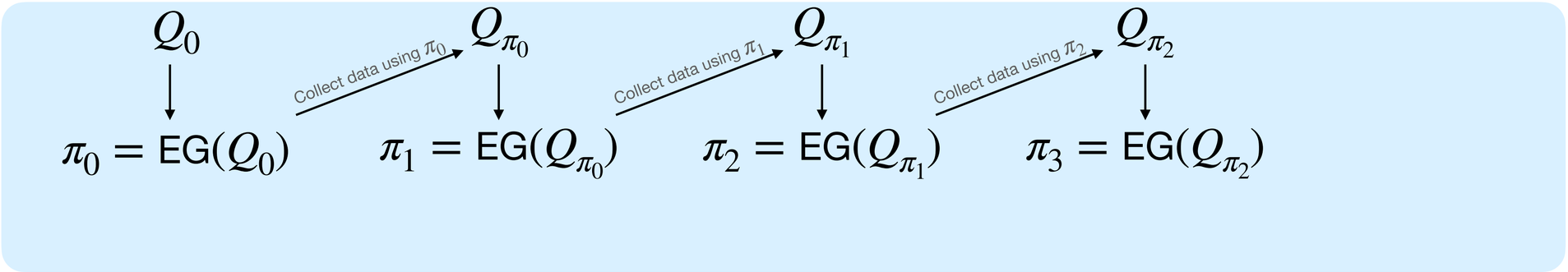
Review: REINFORCE, RwB, A2C

Recitation 7 10-403

Conor Igoe — April 2 2021

REINFORCE, RwB, A2C

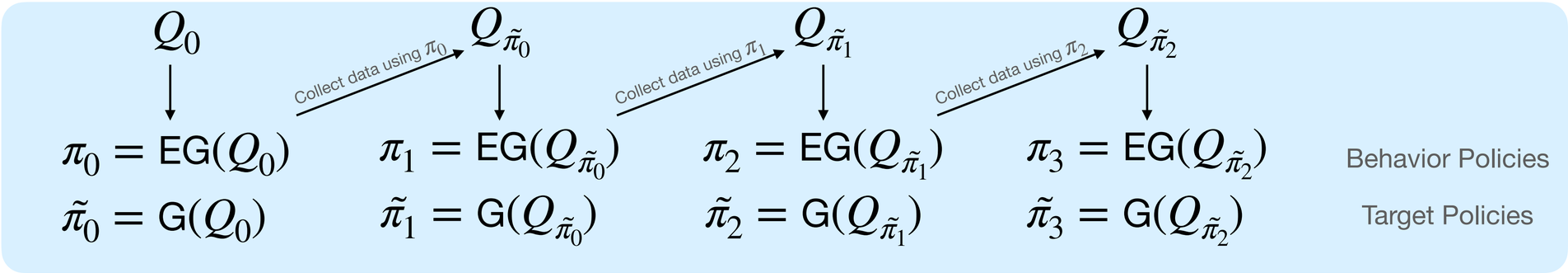
Overview

- Shift in approach: most of the course up until REINFORCE focused on action-value methods for RL:
  - 

  - Different version based on: replacement frequency, Q function estimation (e.g. with or w/o function approximation, with or w/o bootstrapping), behaviour & target policy construction
  - Some example algorithms include tabular SARSA, tabular Q-Learning, semi-gradient N-step SARSA
  - **The important point is that the policy is always defined in terms of some sort of greedification of a q function**

REINFORCE, RwB, A2C

Overview

- Shift in approach: most of the course up until REINFORCE focused on action-value methods for RL:
  - 

  - Different version based on: replacement frequency, Q function estimation (e.g. with or w/o function approximation, with or w/o bootstrapping), behaviour & target policy construction
  - Some example algorithms include tabular SARSA, tabular Q-Learning, semi-gradient N-step SARSA
  - **The important point is that the policy is always defined in terms of some sort of greedification of a q function**

REINFORCE, RwB, A2C, N-step A2C

Overview

- π_θ vs Q_θ

- Advantages:
  - trivial to consider continuous action spaces
  - gives us an opportunity to inject domain knowledge into the problem

REINFORCE, RwB, A2C, N-step A2C

Overview

Towards Generalization and Simplicity in Continuous Control

Aravind Rajeswaran* **Kendall Lowrey*** **Emanuel Todorov** **Sham Kakade**

University of Washington Seattle

{ aravraj, klowrey, todorov, sham } @ cs.washington.edu

Abstract

This work shows that policies with simple linear and RBF parameterizations can be trained to solve a variety of widely studied continuous control tasks, including the OpenAI gym benchmarks. The performance of these trained policies are competitive with state of the art results, obtained with more elaborate parameterizations such as fully connected neural networks. Furthermore, the standard training and testing scenarios for these tasks are shown to be very limited and prone to over-fitting, thus giving rise to only trajectory-centric policies. Training with a diverse initial state distribution induces more global policies with better generalization. This allows for interactive control scenarios where the system recovers from large on-line perturbations; as shown in the supplementary video.

REINFORCE, RwB, A2C, N-step A2C

Overview

- Formal Setup
- Policy Gradient Theorem Recap
- Obtaining Estimators from the Policy Gradient Theorem
- REINFORCE
- REINFORCE with Baseline (RwB)
- (N-step) Advantage Actor Critic (A2C)

REINFORCE, RwB, A2C, N-step A2C

Formal Setup

- Finite episodic MDP
  - $|\mathcal{S}| < \infty$, $|\mathcal{A}| < \infty$, $|\mathcal{R}| < \infty$, $\max(\mathcal{R}) < \infty$, $\min(\mathcal{R}) > -\infty$
  - For every initial condition $s_0 \in \mathcal{S}$, no matter what sequence of actions taken, guaranteed to reach terminal state $s \in \mathcal{S}_{\text{term}} \subset \mathcal{S}$ in finitely many timesteps
- "Nice" stochastic policy class Π parameterised by θ
  - $\pi_\theta(a \mid s)$ differentiable in θ for every (s, a)
  - All policies in Π output non-zero probabilities for all actions in all states

REINFORCE, RwB, A2C, N-step A2C

Formal Setup

- Objective function:

$$J(\theta) = \mathbb{E}_{\tau \sim \mathbb{P}_{\pi_\theta}} \left\{ \sum_{t=0}^{T} \gamma^t R_t \right\}, \gamma \in (0,1)$$

- $\mathbb{P}_{\pi_\theta}$ is a probability distribution over trajectories $\tau = S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
  - $S_0 \sim \rho_0(\cdot)$
  - $A_t \sim \pi_\theta(\cdot \mid S_t) \quad \forall t \geq 0$
  - $(S_{t+1}, R_{t+1}) \sim p(\cdot, \cdot \mid S_t, A_t) \quad \forall t \geq 0$
  - T is random (depends on whether $S_t \in \mathcal{S}_{\text{term}}$, but guaranteed to be finite)

REINFORCE, RwB, A2C, N-step A2C

Policy Gradient Theorem Recap

- Policy Gradient Theorem (version 1)
  - $$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \mathbb{P}_{\pi_\theta}} \left\{ G_0 \sum_{t=0}^{T-1} \nabla_\theta \log \left(\pi_\theta(A_t \mid S_t) \right) \right\}$$
    - $$G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$
      - We derived this in class; relies on the log ratio trick and decomposition of $\mathbb{P}_{\pi_\theta}$ into policy and dynamics components.

REINFORCE, RwB, A2C, N-step A2C

Policy Gradient Theorem Recap

- Policy Gradient Theorem (version 2)
- $$\nabla_\theta J(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{(S,A)\sim\mu_{\pi_\theta}}\left\{ q_{\pi_\theta}(S,A)\,\nabla_\theta \log\left(\pi_\theta(A \mid S)\right)\right\}$$
  - Didn't derive in class but often shows up in literature so good to know (not on quiz)
  - μ_{π_θ} is the discounted state-action "visitation" distribution under policy π_θ

REINFORCE, RwB, A2C, N-step A2C

Policy Gradient Theorem Recap

- Policy Gradient Theorem (version 2)
  - $$\nabla_\theta J(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{(S,A)\sim\mu_{\pi_\theta}}\left\{ q_{\pi_\theta}(S,A)\,\nabla_\theta \log\left(\pi_\theta(A\,|\,S)\right)\right\}$$
    - $\mu_{\pi_\theta}(s,a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}(S_t = s, A_t = a)$
      - Draw a timestep $M \sim$ Geometric$(1-\gamma)$
      - Then deploy π_θ for M timesteps and observe $S_0, A_0, R_1, \ldots, S_{M-1}, A_{M-1}$
      - (S_{M-1}, A_{M-1}) will be distributed according to μ_{π_θ}

REINFORCE, RwB, A2C, N-step A2C

Policy Gradient Theorem Recap

- Policy Gradient Theorem (version 2)
  - $$\nabla_\theta J(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{(S,A)\sim\mu_{\pi_\theta}}\left\{ q_{\pi_\theta}(S,A)\,\nabla_\theta \log\left(\pi_\theta(A \mid S)\right)\right\}$$
    - $$\mu_{\pi_\theta}(s,a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}(S_t = s, A_t = a)$$
      - See Sham Kadade's RL Theory Notes for derivation (Theorem 9.4)

REINFORCE, RwB, A2C, N-step A2C

Obtaining Estimators from the Policy Gradient Theorem

- Policy Gradient Theorem gives us a starting point to build an estimators:
  - E.g. using version 1:
    - $$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \mathbb{P}_{\pi_\theta}} \left\{ G_0 \sum_{t=0}^{T} \nabla_\theta \log \left(\pi_\theta(A_t \mid S_t) \right) \right\}$$
    - $$\hat{g} = G_0 \sum_{t=0}^{T-1} \nabla_\theta \log \left(\pi_\theta(A_t \mid S_t) \right)$$ (Take a single sample from the RV in the Policy Gradient Theorem)

REINFORCE, RwB, A2C, N-step A2C

Obtaining Estimators from the Policy Gradient Theorem

- Policy Gradient Theorem gives us a starting point to build estimator:
  - E.g. using version 1:
    - $$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \mathbb{P}_{\pi_\theta}} \left\{ G_0 \sum_{t=0}^{T} \nabla_\theta \log \left(\pi_\theta(A_t \mid S_t) \right) \right\}$$
    - $$\hat{g} = \sum_{t=0}^{T-1} G_t \nabla_\theta \log \left(\pi_\theta(A_t \mid S_t) \right)$$

(Inject temporal structure; this is the REINFORCE policy gradient estimate; note that it is basically a single sample estimator of the Expectation from the PG theorem, so expect very high variance!)

REINFORCE, RwB, A2C, N-step A2C

REINFORCE

1: **procedure** REINFORCE
2: *Start with policy network* π_θ
3: **repeat for** E **training episodes:**
4: *Generate an episode* $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$ *following* $\pi_\theta(\cdot)$
5: **for** $t = 0, 1, \ldots, T$:
6: $G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$
7: $\hat{g} = \sum_{t=0}^{T-1} G_t \nabla_\theta \ln \pi_\theta(A_t|S_t)$
8: $\theta \leftarrow \theta + \alpha \hat{g}$
9: **end procedure**

REINFORCE, RwB, A2C, N-step A2C

RwB

- Subtracting a state dependent baseline doesn't affect bias

$$\hat{g} = \sum_{t=0}^{T-1} \left(G_t - b(S_t)\right) \nabla_\theta \log\left(\pi_\theta(A_t \mid S_t)\right)$$

  - Convenient to choose $b(s) = \hat{V}_\phi(s)$
  - Baseline can be learned using Monte Carlo methods, or semi-gradient TD methods; both are known as RwB
  - Everything else exactly the same as REINFORCE

REINFORCE, RwB, A2C, N-step A2C

(N-step) A2C

- Take a look at the following version of the true Policy Gradient (with a baseline):
- $$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \mathbb{P}_{\pi_\theta}} \left\{ \sum_{t=0}^{T-1} \left(G_t - b(S_t) \right) \nabla_\theta \log \left(\pi_\theta(A_t \mid S_t) \right) \right\}$$
- The true Policy Gradient involves taking an expectation over several terms. In Vanilla REINFORCE with Baseline, we really were just using a single sample from the random variable under the expectation (in blue) as our estimator
- The main idea with Actor-Critic methods is to do the same thing, but using a critic to approximately perform *part* of this expectation by relying on previous data to reduce the variability coming from G_t terms i.e. use an estimate of $\mathbb{E}\{G_t \mid S_t, A_t\}$ instead of G_t

REINFORCE, RwB, A2C, N-step A2C

(N-step) A2C

- Take a look at the following version of the true Policy Gradient (with a baseline):
- $$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \mathbb{P}_{\pi_\theta}} \left\{ \sum_{t=0}^{T-1} \left(G_t - b(S_t) \right) \nabla_\theta \log \left(\pi_\theta(A_t \,|\, S_t) \right) \right\}$$
- If we use $\hat{V}_\phi(s)$ to construct estimates of $\mathbb{E}\{G_t \,|\, S_t, A_t\}$ by bootstrapping then we call $\hat{V}_\phi$ a *critic* (and π_θ an *actor*)
  - For example:
    $$\mathbb{E}\{G_t \,|\, S_t, A_t\} \approx R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \gamma^{T-t} R_T \quad \text{(high variance but unbiased)}$$
    $$\mathbb{E}\{G_t \,|\, S_t, A_t\} \approx R_{t+1} + \gamma \hat{V}_\phi(S_{t+1}) \quad \text{(lower variance but biased)}$$
  - Bootstrapping for 1 time-step = Advantage Actor-Critic (A2C)
  - Bootstrapping for N time-steps = N-step A2C

REINFORCE, RwB, A2C, N-step A2C

(N-step) A2C

- Take a look at the following version of the true Policy Gradient (with a baseline):
- $$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \mathbb{P}_{\pi_\theta}} \left\{ \sum_{t=0}^{T-1} \left(G_t - b(S_t) \right) \nabla_\theta \log \left(\pi_\theta(A_t \mid S_t) \right) \right\}$$

- **Why do we say $R_{t+1} + \gamma \hat{V}_\phi(S_{t+1})$ has lower variance?**
  - Because it is a lower variance estimate of $\mathbb{E}\{G_t \mid S_t, A_t\}$, and $\hat{V}_\phi(s)$ has averaged over much of the randomness that previously existed in G_t
- **Why do we say using $R_{t+1} + \gamma \hat{V}_\phi(S_{t+1})$ introduces bias?**
  - Imagine using a really inexpressive network for $\hat{V}_\phi(s)$. Then our estimates for $\mathbb{E}\{G_t \mid S_t, A_t\}$ will be very poor and we can never hope to estimate the policy gradient well!

REINFORCE, RwB, A2C, N-step A2C

(N-step) A2C

- Take a look at the following version of the true Policy Gradient (with a baseline):
  - $$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \mathbb{P}_{\pi_\theta}} \left\{ \sum_{t=0}^{T-1} \left(G_t - b(S_t) \right) \nabla_\theta \log \left(\pi_\theta(A_t \,|\, S_t) \right) \right\}$$
- $$\hat{g} = \sum_{t=0}^{T-1} \left(R_{t+1} + \gamma \hat{V}_\phi(S_{t+1}) - \hat{V}_\phi(S_t) \right) \nabla_\theta \log \left(\pi_\theta(A_t \,|\, S_t) \right)$$
  - Note that the critic network is used in two places: baseline & to bootstrap an estimate for $\mathbb{E}\{G_t \,|\, S_t, A_t\}$