Natural Actor Critic

Steering policy gradients in a different direction

John Martin Jr.

March 30, 2016

Introduction

Motivation

- RL is applied to continuous problems through function approximation

Introduction

Motivation

- RL is applied to continuous problems through function approximation
- Policy-gradient methods provide strong convergence guarantees

Introduction

Motivation

- RL is applied to continuous problems through function approximation
- Policy-gradient methods provide strong convergence guarantees
- Standard gradient methods can be inefficient

Introduction

Motivation

- RL is applied to continuous problems through function approximation
- Policy-gradient methods provide strong convergence guarantees
- Standard gradient methods can be inefficient

Natural gradients are better

- Peters, Vijayakumar, and Schaal. *Natural Actor-Critic*[1]

Reinforcement Learning

Given the problem description $\langle \mathcal{X}, \mathcal{U}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ in which

- $\mathcal{X}$: State set,
- $\mathcal{U}$: Set of admissible inputs,
- $\mathcal{P}$: Transition probability matrix
  $\mathcal{P}^u_{xx'} = \mathbb{P}\big[x_{k+1} = x' | x_k = x, u_k = u\big]$,
- $\mathcal{R}$: Reward function $\mathcal{R}^u_x = \mathbb{E}\big[g_k | x_k = x, u_k = u\big]$,
- γ : Discount factor in $[0, 1)$,

Reinforcement Learning

Given the problem description $\langle \mathcal{X}, \mathcal{U}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ in which

- $\mathcal{X}$: State set,
- $\mathcal{U}$: Set of admissible inputs,
- $\mathcal{P}$: Transition probability matrix
  $\mathcal{P}^u_{xx'} = \mathbb{P}\big[x_{k+1} = x' | x_k = x, u_k = u\big]$,
- $\mathcal{R}$: Reward function $\mathcal{R}^u_x = \mathbb{E}\big[g_k | x_k = x, u_k = u\big]$,
- γ : Discount factor in $[0, 1)$,

compute a policy $\pi(u_k|x_k) = \mathbb{P}\big[u_k = u | x_k = x\big]$ maximizing

$$q_\pi(x, u) = \mathbb{E}\big[J_k | x_k = x, u_k = u\big],$$

Reinforcement Learning

Given the problem description $\langle \mathcal{X}, \mathcal{U}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ in which

- $\mathcal{X}$: State set,
- $\mathcal{U}$: Set of admissible inputs,
- $\mathcal{P}$: Transition probability matrix $\mathcal{P}^u_{xx'} = \mathbb{P}\big[x_{k+1} = x' | x_k = x, u_k = u\big]$,
- $\mathcal{R}$: Reward function $\mathcal{R}^u_x = \mathbb{E}\big[g_k | x_k = x, u_k = u\big]$,
- γ : Discount factor in $[0, 1)$,

compute a policy $\pi(u_k | x_k) = \mathbb{P}\big[u_k = u | x_k = x\big]$ maximizing

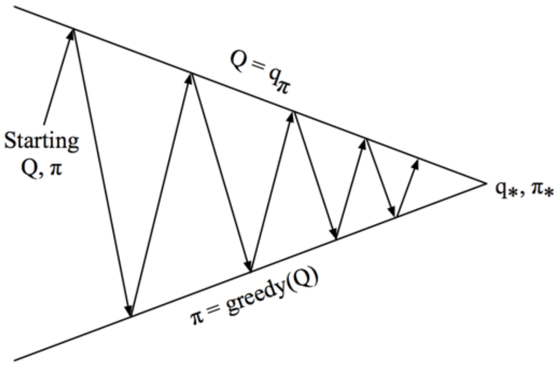
$$q_\pi(x, u) = \mathbb{E}\big[J_k | x_k = x, u_k = u\big],$$

where each value function obeys the recursive decomposition of

$$q_\pi(x, u) = g(x, u) + \gamma \int_{\mathcal{X}} \mathcal{P}^u_{xx'} \int_{\mathcal{U}} \pi(u|x) q_\pi(x, u) dx du$$

Solutions can be obtained from Generalized Policy Iteration.

Generalized Policy Iteration



Alternate between evaluation and improvement.

- Policy Evaluation: determine worth of the current policy
- Policy Improvement: select more valuable actions

Policy Gradient

Policy Optimization

- Parameterize the policy $\pi_\theta = \mathbb{P}\big[u_k | x_k, \theta\big]$
- Maximize $J(\theta) = \int_{\mathcal{X}} d_{\pi_\theta}(x) \int_{\mathcal{U}} \pi_\theta(u|x) g(x,u) dx du$
- Policy gradient is known:

$$\begin{aligned} \nabla_\theta \pi_\theta(x,u) &= \pi_\theta(x,u) \frac{\nabla_\theta \pi_\theta(x,u)}{\pi_\theta(x,u)} \\ &= \pi_\theta(x,u) \nabla_\theta \log \pi_\theta(x,u) \end{aligned}$$

Policy Gradient Theorem

For an average value objective, and for any differentiable π_θ,

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\big\{\nabla_\theta \log \pi_\theta(x,u) A_{\pi_\theta}(x,u)\big\}.$$

Here $A_{\pi_\theta}(x,u) = q_{\pi_\theta}(x,u) - b_{\pi_\theta}(x)$ is the *advantage function*.

Actor-Critic Algorithms

- Maintain two sets of parameters w, θ
  - Critic: Updates action-value parameters w
  - Actor: Updates policy parameters θ according to critic
- Follow an approximate policy gradient
- Steepest parameter gradient may be inconsistent with the true gradient

Actor-Critic Algorithms

$\xrightarrow{\theta}$

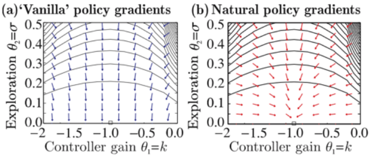
- Maintain two sets of parameters w, θ
  - Critic: Updates action-value parameters w
  - Actor: Updates policy parameters θ according to critic
- Follow an approximate policy gradient
- Steepest parameter gradient may be inconsistent with the true gradient

Actor-Critic Algorithms

$$\xrightarrow{\theta} \qquad \xleftarrow[A^w_{\pi_\theta}]{}$$

- Maintain two sets of parameters w, θ
  - Critic: Updates action-value parameters w
  - Actor: Updates policy parameters θ according to critic
- Follow an approximate policy gradient
- Steepest parameter gradient may be inconsistent with the true gradient

Natural Actor-Critic



- Using compatible function approximation

$$A_{\pi_\theta} = \nabla_\theta \log \pi_\theta(x, u)^\top w$$

- Applying the policy gradient theorem reveals

$$\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\pi_\theta}\{\nabla_\theta \log \pi_\theta(x, u) A_{\pi_\theta}(x, u)\}, \\
&= \mathbb{E}_{\pi_\theta}\{\nabla_\theta \log \pi_\theta(x, u) \nabla_\theta \log \pi_\theta(x, u)^\top\} w, \\
&= G_\theta w, \\
\nabla_\theta^{\text{nat}} J(\theta) &= G_\theta^{-1} \nabla_\theta J(\theta) = w.
\end{aligned}$$

References I

[1] Jan Peters, Sethu Vijayakumar, and Stefan Schaal.
Natural actor-critic.
Proceedings of the European Conference on Machine Learning (ECML), pages 280–291, 2005.