Policy Optimization In Reinforcement Learning

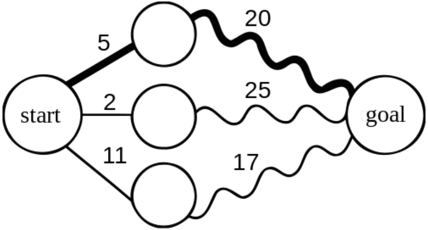
Ziniu Li
ziniuli@link.cuhk.edu.cn

The Chinese University of Hong Kong, Shenzhen, Shenzhen, China

July 22, 2021

Motivation Example

How to solve this decision-making problem? i.e., the shortest path finding.

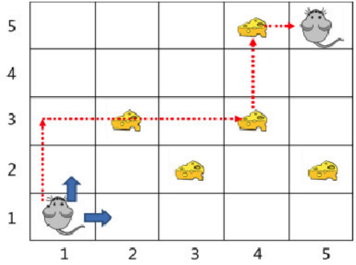


[figure from wiki]

- Just enumerate all paths and find the shortest path.

Motivation Example

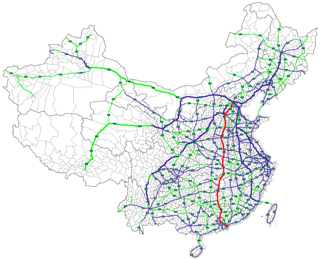
How to solve this decision-making problem? i.e., the shortest path finding + eating items.



- (Still) enumerate all paths and find the most valuable path.

Motivation Example

How to solve this decision-making problem? i.e., the shortest path from Beijing to Shenzhen with a cheap tool.



- It is intractable to enumerate all paths and picks up the best one.
- How to efficiently solve large-scale sequential decision making tasks?

Dynamic Programming and Markov Decision Process

Approach: Dynamic Programming + Markov Decision Process

R. E. Bellman (1920-1984)

Ronald A. Howard (1934-)

- Bellman Optimality Equation:

$$V(s) = \max_{a \in \mathcal{A}} Q(s, a) \tag{1a}$$

$$Q(s, a) = r(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) V(s') \tag{1b}$$

- It reduces the "multi-stage" maximization problem to "sing-stage" optimization sub-problems.

Value Iteration

How to solve Bellman Optimality Equation?

- (**Method 1**) Value Iteration.

$$V^{k+1} = \mathcal{T}V^k \quad \text{with} \quad (\mathcal{T}V)(s) = \max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V(s') \right].$$

What about the control/policy?

- Derive the greedy policy w.r.t. $\widehat{Q}$, i.e., $\pi(s) = \text{argmax}_{a \in \mathcal{A}} \widehat{Q}(s,a), \forall s \in \mathcal{S}$.

Disadvantage of Method 1

- (Issue 1a) Suboptimality: an ε-optimal $\widehat{Q}$ induces an $\varepsilon/(1-\gamma)$-optimal greedy policy π.
- (Issue 1b) It is not clear how to obtain the greedy policy under the case where action space is continuous.

Policy Iteration

Policy Iteration is an algorithm based on DP to directly solve the optimal policy.

- Firstly evaluate action-value function Q^π:

$$Q^\pi(s,a) = \mathbb{E}_{a\sim\pi(\cdot|s),s'\sim P(\cdot|s,a),s'\sim\pi(\cdot|s')}\left[r(s,a) + \gamma Q^\pi(s',a')\right]$$

- Secondly improve the policy by one-stage optimization:

$$\pi^{k+1}(a|s) \leftarrow \operatorname*{argmax}_{a\in\mathcal{A}} Q^{\pi^k}(s,a).$$

Remark

- We optimize the decision from a "discrete" view.
  - We operate with deterministic policies, which corresponds to a "path" in the shortest path problem.

Motivation For Policy Gradient Methods

Can we model the optimization as a mathematical programming problem?

$$\min_{x \in \mathbb{R}^n} f(x) \quad \Longrightarrow \quad \max_{\pi} V(\pi) := \sum_{s \sim \rho} \rho(s) V^{\pi}(s).$$

- where ρ is the initial state distribution.

What is the parameter?

- (**Approach 1**) Direct parameterization: $\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$ with $\pi(a|s)$ being the probability of selecting action a at state s.

	a_1	a_2	a_3
s_1	0.3	0.4	0.3
s_2	0.5	0	0.5

Policy Gradient Method I

$$\max_{\pi} V(\pi) \quad \text{s.t.} \quad \pi \in \Delta(\mathcal{A})^{|\mathcal{S}|} \tag{2}$$

Is (2) differentiable?

- Yes! It's gradient can be computed as
$$\frac{\partial V(\pi)}{\partial \pi(a|s)} = \frac{1}{1-\gamma} d^{\pi}(s) Q^{\pi}(s,a),$$
where $d^{\pi}(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s)$.

Is (2) smooth?

- Yes! For all policies π, π', we have
$$\left\| \nabla_{\pi} V(\pi) - V^{\pi'} \right\|_2 \leq \frac{2\gamma|\mathcal{A}|}{(1-\gamma)^3} \left\| \nabla \pi - \pi' \right\|_2 .$$

Policy Gradient Method II

Is (2) concave?

- No! There are some MDPs such that (2) is nonconcave.

Why (2) is nonconcave?

- Let us consider a simple function:

$$f(x,y) = xy, \quad \nabla f(x) = \begin{bmatrix} y \\ x \end{bmatrix}, \quad \nabla^2 f(x,y) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{3}$$

  ♠ : this function is convex/concave w.r.t x or y, but is neither convex or concave w.r.t. (x, y).

- Informally, expected return $= \sum_{\text{trajectory}} \mathbb{P}(\text{trajectory}) \times R(\text{trajectory})$.
- We see that $\mathbb{P}(\text{trajectory}) = \prod_{t=0}^{\infty} \pi(a_t|s_t) p(s_{t+1}|s_t, a_t)$ could have the structure in (3); therefore, we expect it is nonconcave for policy optimization.

Policy Gradient Method Historical Remark

Before 2020, people believe that policy gradient methods are not important because they could converge to the local solution.

- There are total 17 chapters in the book [Sutton and Barto, 2018] but only Chapter 13 is for policy gradient methods.

Since 2015, deep policy gradient methods (with neural networks) attracts more interests due to its superior performance.

Trust region policy optimization
J Schulman, S Levine, P Abbeel… - … on machine learning, 2015 - proceedings.mlr.press
In this article, we describe a method for optimizing control policies, with guaranteed monotonic improvement. By making several approximations to the theoretically-justified scheme, we develop a practical algorithm, called Trust Region Policy Optimization (TRPO) …
☆ 99 被引用次数：3687 相关文章 所有 17 个版本 »

Proximal policy optimization algorithms
J Schulman, F Wolski, P Dhariwal, A Radford… - arXiv preprint arXiv …, 2017 - arxiv.org
We propose a new family of policy gradient methods for reinforcement learning, which alternate between sampling data through interaction with the environment, and optimizing a" surrogate" objective function using stochastic gradient ascent. Whereas standard policy …
☆ 99 被引用次数：4810 相关文章 所有 7 个版本 »

Global Convergence of Policy Gradient Methods

Claim 1: the policy gradient method by direct parameterization can linearly converge to the global optimization even though it is a nonconcave optimization problem [Bhandari and Russo, 2021].

- Concavity is just a sufficient condition to derive the global convergence.

Jalaj Bhandari (PhD student of Columbia University)

Daniel Russo (Assistant Professor at Columbia University)

Claim 2: Policy gradient methods with direct parameterization can be viewed the soft policy iteration [Bhandari and Russo, 2021].

Weighted Bellman Objective and Soft Policy Iteration I

Weighted Bellman Objective: For any policy π, let us introduce weighted Bellman objective, defined as

$$\mathcal{B}(\overline{\pi}|d^\pi, Q^\pi) = \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d^\pi(s)Q^\pi(s,a)\overline{\pi}(a|s) = \langle Q^\pi, \overline{\pi}\rangle_{d^\pi\times 1}, \tag{4}$$

- Note the decision variable is $\overline{\pi}$ with π being fixed.
- Our objective is to maximize such defined weighted Bellman objective,

$$\pi^+ = \underset{\overline{\pi}\in\Delta(\mathcal{A})^{|\mathcal{S}|}}{\operatorname{argmax}} \mathcal{B}(\overline{\pi}|d^\pi, Q^\pi).$$

- At state s, the optimal solution is $\pi^+(s) = \operatorname{argmax}_{a\in\mathcal{A}} Q^\pi(a|s)$, which is identical with policy iteration.

Weighted Bellman Objective and Soft Policy Iteration II

- The gradient of Bellman objective function is

$$\frac{\partial \mathcal{B}(\overline{\pi}|d^{\pi}, Q^{\pi})}{\partial \overline{\pi}(a|s)} = d^{\pi}(s)Q^{\pi}(s,a).$$

Scaled Objective Function:

$$\ell(\pi) := (1-\gamma)V(\pi) = (1-\gamma)\sum_{s\sim\rho}\rho(s)V^{\pi}(s). \tag{5}$$

- Recall the policy gradient theorem states that

$$\frac{\partial \ell(\pi)}{\partial \pi(a|s)} = d^{\pi}(s)Q^{\pi}(s,a).$$

The gradient of scaled objective function is same as the weighted Bellman objective.

Projected Gradient Algorithms I

- The iterate of policy gradient methods on $\ell(\pi)$ can be translated to the one by maximizing the Bellman objective function.

Example 1 Frank Wolfe

1) optimize the linearized objective over the constrained set;

$$\pi^{+} = \underset{\overline{\pi} \in \Pi}{\operatorname{argmax}} \langle \nabla \ell(\pi), \overline{\pi} \rangle = \underset{\overline{\pi} \in \Pi}{\operatorname{argmax}} \langle Q^{\pi}, \overline{\pi} \rangle_{d^{\pi} \times 1};$$

2) make a convex combination update:

$$\pi' = (1 - \eta)\pi + \eta \pi^{+} \quad \text{with} \quad \eta \in [0, 1]$$

Projected Gradient Algorithms II

Example 2 Projected Gradient Ascent

We first take a gradient descent update then project the updated policy into the constrained set:

$$\pi' = \underset{\bar{\pi}\in\Pi}{\operatorname{argmax}} \left\{ \langle \nabla \ell(\pi), \bar{\pi} \rangle - \frac{1}{2\eta} \|\bar{\pi} - \pi\|_2^2 \right\}$$
$$= \underset{\bar{\pi}\in\Pi}{\operatorname{argmax}} \left\{ \langle Q^\pi, \bar{\pi} \rangle_{d^\pi \times 1} - \frac{1}{2\eta} \|\bar{\pi} - \pi\|_2^2 \right\}$$

Projected Gradient Algorithms III

Example 3 Mirror Ascent

We replace the ℓ-2 norm "regularization" to a geometry-aware "regularization":

$$\pi' = \underset{\pi \in \Pi}{\operatorname{argmax}} \left\{ \langle \nabla \ell(\pi), \overline{\pi} \rangle - \frac{1}{\eta} D_{\mathrm{KL}}(\overline{\pi} \| \pi) \right\}.$$

where $D_{\mathrm{KL}}(\overline{\pi}\|\pi) = \sum_{s \in \mathcal{S}} D_{\mathrm{KL}}\left(\pi(\cdot|s) \| \overline{\pi}(\cdot|s)\right)$, and $D_{\mathrm{KL}}(p\|q) = \sum_{x \in \mathcal{X}} p(x) \log\left(p(x)/q(x)\right)$ for two probability distributions p and q.

It is well know that the solution is the exponentiated gradient update [Bubeck, 2015, Section 6.3],

$$\pi'(a|s) = \frac{\pi(a|s) \exp\left(\eta d^{\pi}(s) Q^{\pi}(s,a)\right)}{\sum_{a \in \mathcal{A}} \pi(a|s) \exp\left(\eta d^{\pi}(s) Q^{\pi}(s,a)\right)}.$$

Summary of Part II

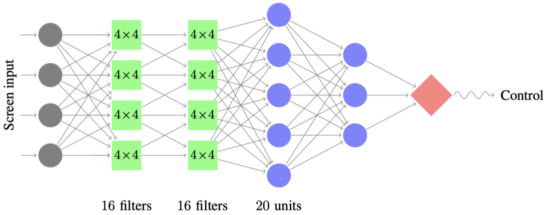
- Policy gradient methods with **direct parameterization** can be viewed an soft policy iteration (especially frank-wolfe based algorithm).
- By well-designed stepsizes (line search or constant stepsize), policy gradient methods can linearly converge to the global optimization solution.

People never use direct parameterization in practice!

- We cannot do any function approximation with direct parameterization.
- It is tiresome to implement the simplex projection.

Motivation For Softmax Parameterization

- Given the state/observation s, we learn a feature extractor to obtain the hidden state h, then we use h to predict control action a.
- Softmax parameterization: $\pi(i|s) = \exp(Wh)[i] / \sum_i \exp(Wh)[i]$.
- Importantly, we do not need consider the constraint! The policy optimization becomes an unconstrained optimization problem.



[Figure from [Schulman et al., 2015].]

Softmax Policy Optimization

Problem Formulation

- For simplicity, we assume that $\theta \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$ so that there is no function approximation error.
- Furthermore, we mainly focus on 1-state MDP problems to illustrate challenges, concepts and ideas. Under this case, $\theta \in \mathbb{R}^{|\mathcal{A}|}$ and $\pi_\theta(a) = \exp(\theta[a]) / \sum_a \exp(\theta[a])$.

Now, our problems becomes

$$\max_{\theta \in \mathbb{R}^{|\mathcal{A}|}} \pi_\theta^\top r := \sum_{a \in \mathcal{A}} \pi_\theta(a) r(a).$$

- We assume that $r(a) \in [0, 1], \forall a \in \mathcal{A}$.
- Yes, this problem seems trivial just like you are asked to find a classifier that shatters two points $(1, \text{class A})$ and $(-1, \text{class B})$. However, the idea can be extended to general MDP problems; see the full paper [Mei et al., 2020].

Softmax Policy Optimization I

$$\max_{\theta \in \mathbb{R}^{|\mathcal{A}|}} \pi_\theta^\top r := \sum_{a \in \mathcal{A}} \pi_\theta(a) r(a). \tag{6}$$

Q1: Is (6) differentiable?

- Yes. The gradient is

$$\nabla_\theta \pi_\theta^\top r = \left(\mathbf{diag}(\pi_\theta) - \pi_\theta \pi_\theta^\top\right) r$$

Q2: Is (6) smooth?

- Yes. For any $\theta, \theta' \in \mathbb{R}^{|\mathcal{A}|}$, we have

$$\left\|\nabla_\theta \pi_\theta^\top r - \nabla_\theta \pi_{\theta'}^\top r\right\| \le \frac{5}{2} \left\|\theta - \theta'\right\|_2 .$$

Q3: Is (6) concave?

- No! For some 1-state MDPs, (6) is nonconcave.

Softmax Policy Optimization II

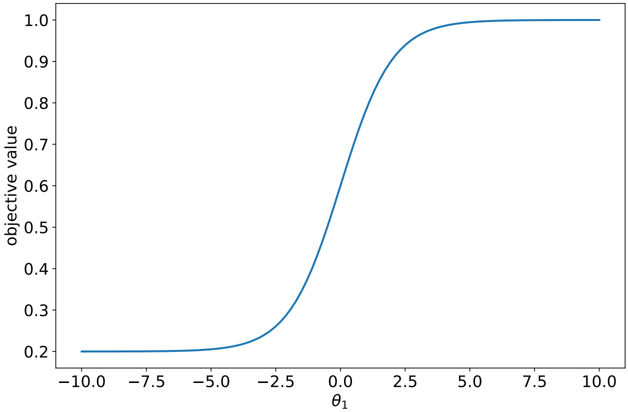
Q4: Why (6) nonconcave?

- (**Sigmoid Example**) Consider the case where $r = (1.0, 0.2)$ and the parameterization $(\theta, 0)$, i.e., the second parameter is fixed.

$$\pi_\theta^\top r = \sigma(\theta) + 0.2(1 - \sigma(\theta)) = 0.2 + 0.8\sigma(\theta),$$

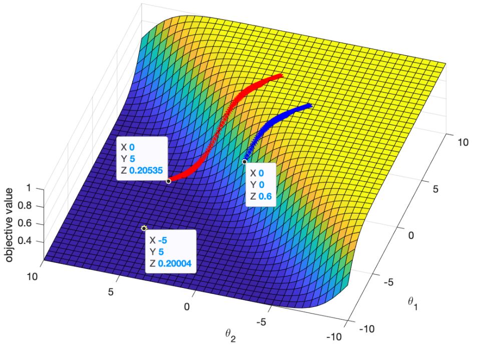
where $\sigma(\theta) = \exp(\theta)/(1 + \exp(\theta))$ is the sigmoid function.

Softmax Policy Optimization III



[Objective function $0.2 + 0.8\sigma(\theta)$, which is neither concave nor convex.]

Softmax Policy Optimization IV



[Objective function for $\pi_\theta^\top r$ with $r = (1.0, 0.2)$ and $\theta = (\theta_1, \theta_2)$. In addition, there are 3 trajectories by gradient ascent with stepsize $\eta = 2/5$, which corresponds to different initialization: $(0, 5), (0, 0), (-5, 5)$.]

Softmax Policy Optimization V

- Even though it is a nonconcave optimization problem, gradient ascent is supposed to work.

Why gradient ascent work?

- (Conjecture 1) Connection with soft policy iteration?
  - No! Gradient ascent directly optimize θ rather π_θ so that the iterate is not close to the one of policy iteration.
- (Conjecture 2) Error bound (or gradient domination) regularity?
  - Yes and no! It indeed satisfies certain Łojasiewicz condition but the parameter vanishes!

Softmax Policy Optimization VI

Lemma 1 Non-uniform Łojasiewicz condition for 1-state MDP [Mei et al., 2020]

Consider the 1-state MDP and assume $r(a) \in [0,1], \forall a \in \mathcal{A}$ has one unique optimal action. Let $\pi^* := \operatorname{argmax}_{\pi \in \Delta} \pi^\top r$. Then

$$\left\| \frac{d\pi_\theta^\top r}{d\theta} \right\|_2 \geq \pi_\theta(a^*) \cdot \underbrace{(\pi^* - \pi_\theta)^\top r}_{\text{optimality gap}},$$

where $a^* := \operatorname{argmax}_{a \in [K]} r(a)$ is the optimal action.

- Lemma 1: if we reach a stationary point θ with $\pi_\theta(a^*) > 0$, then this stationary point is a globally optimal solution.

Softmax Policy Optimization VII

Lemma 2 Pseudo-convergence rate for 1-state MDP [Mei et al., 2020]

Consider the policy gradient with the softmax parameterization, in which the stepsize $\eta = 2/5$ is employed. Then for all $t > 0$

$$(\pi^* - \pi_{\theta_t})^\top r \leq 5/(t \cdot c_t^2),$$

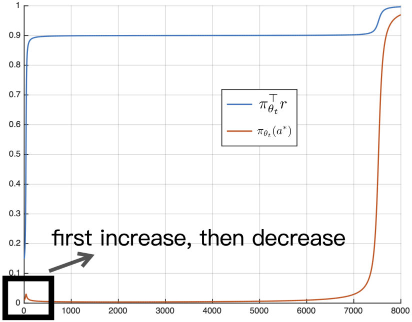
where $c_t := \min_{1 \leq \ell \leq t} \pi_{\theta_\ell}(a^*) > 0$, also

$$\sum_{t=1}^{T} (\pi^* - \pi_{\theta_t})^\top r \leq \min \left\{ \sqrt{5T}/c_T, (5 \log T)/c_T^2 + 1 \right\}.$$

- (**Observation 1**) In the previous example, there are two stationary points but gradient ascent always pickup the correction direction so that $\pi_\theta(a^*) \geq \pi_{\theta_0}(a^*) > 0$, which explains why gradient ascent works.

Does Observation 1 Hold for General Cases?

- No! In the previous example, there are only two actions so that $\pi(a)\downarrow \Longrightarrow \pi(a^*)\uparrow$.
- For general case, $\pi(a)\downarrow \not\Longrightarrow \pi(a^*)\uparrow$ due to sub-optimal actions and bad initialization.



[Illustration for the bad initialization [Mei et al., 2020]. $r = (1.0, 0.9, 0.1)$ with $\theta_1 = (0.01, 0.05, 0.94)$. For this bad initialization, the second near-optimal action dominates in the initial stage.]

Convergence Result For Softmax Parameterization

Lemma 3 [Mei et al., 2020]: For 1-state MDP with the softmax parameterization, we have $\inf_{t\geq 1} \pi_{\theta_t}(a^*) > 0$.

Proposition 1 [Mei et al., 2020]: For any initialization, there exists $t_0 > 0$ such that for any $t \geq t_0, t \mapsto \pi_{\theta_t}(a^*)$ is increasing. In particular, $t_0 = 1$ when π_{θ_1} is the uniform distribution.

Corollary 1 True convergence rate for 1-state MDP [Mei et al., 2020]

With softmax parameterization, for all $t > 0$,

$$(\pi^* - \pi_{\theta_t})^\top r \leq C/t,$$

where $1/C = [\inf_{t\geq 1} \pi_{\theta_t}(a^*)]^2 > 0$ is a constant that depends on r and θ_1, but it does not depend on the time t.

Intuition Behind Lemma 3 and Proposition 1

Goal: we want to argue that after some t_0, $\pi_{\theta_t}(a^*)$ is increasing, $\forall t \geq t_0$.

- (**Step 1**): We have to identify some "nice" region $\mathcal{R}_1 = \{\theta_t\}$ so that for any $\theta_t \in \mathcal{R}$: 1) $\theta_{t+1} \in \mathcal{R}_1$; 2) $\pi_{\theta_{t+1}}(a^*) \geq \pi_{\theta_t}(a^*)$.
  - (**Claim 1**): "gradient domination" guarantees $\mathcal{R}_1$; i.e.,

$$\mathcal{R}_1 := \left\{\theta : \frac{d\pi_\theta^\top r}{d\theta(a^*)} \geq \frac{d\pi_\theta^\top r}{d\theta(a)}, \quad \forall a \neq a^*\right\}.$$

- (**Step 2**) Show that $\mathcal{R}_1$ contains a subset $\mathcal{N}_c$ such that $\pi_\theta(a^*) > c'$:

$$\mathcal{N}_c := \left\{\theta : \pi_\theta(a^*) \geq \frac{c}{c+1}\right\},$$

  where $c = g(|\mathcal{A}|, \Delta)$ with $\Delta = r(a^*) - \max_{a \neq a^*} r(a) > 0$.

- (**Step 3**) Show that there exists a finite time t_0 so that $\theta_{t_0} \in \mathcal{N}_c$, which is based on the asymptotic convergence result that $\pi_{\theta_t}(a^*) \to 1$ when $t \to \infty$ in [Agarwal et al., 2020].

Summary of Part III

- We focus on the 1-state MDP problem:

$$\max_{\theta \in \mathbb{R}^{|\mathcal{A}|}} \pi_\theta^\top r := \sum_{a \in \mathcal{A}} \pi_\theta(a) r(a).$$

- Even though obj. is linear w.r.t. π_θ, it is nonconcave w.r.t. θ.
- Luckily, it satisfies a non-uniform Łojasiewicz condition.
- However, the bad initialization due to sub-optimal actions makes the progress slow.

References I

A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In Proceedings of the 33rd Annual Conference on Learning Theory, pages 64–66, 2020.

J. Bhandari and D. Russo. On the linear convergence of policy gradient methods for finite mdps. In Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, pages 2386–2394, 2021.

S. Bubeck. Convex optimization: Algorithms and complexity. Foundations and Trends in Machine Learning, 8(3-4):231–357, 2015.

J. Mei, C. Xiao, C. Szepesvári, and D. Schuurmans. On the global convergence rates of softmax policy gradient methods. In Proceedings of the 37th International Conference on Machine Learning, pages 6820–6829, 2020.

References II

J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz. Trust region policy optimization. In Proceedings of the 32nd International Conference on Machine Learning, pages 1889–1897, 2015.

R. S. Sutton and A. G. Barto. Reinforcement Learning: An Introduction. MIT press, 2018.