Policy Gradient [2]

DOROZHKO Anton

Novosibirsk State University

May 31, 2019

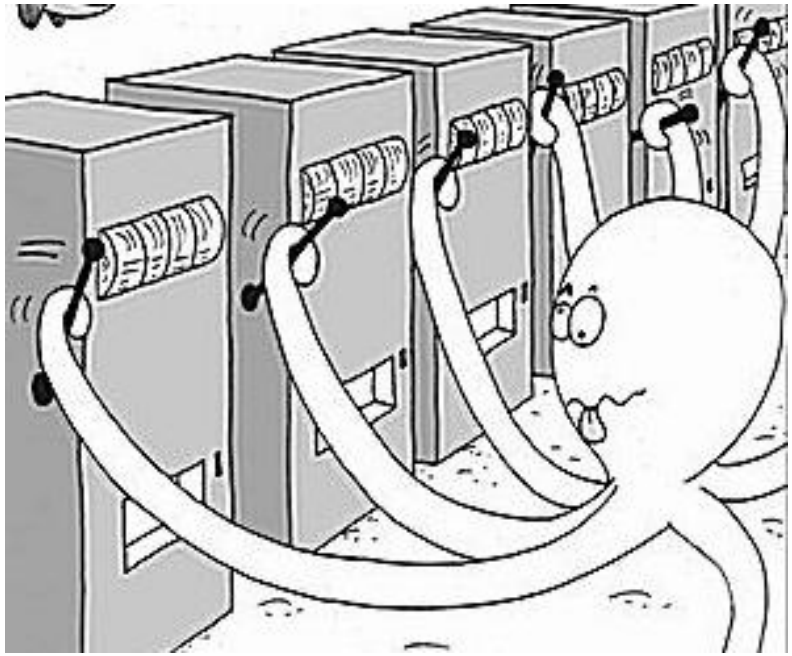
[2] SpinningUP RL research : Part 3

Outline

1. Exploration
2. Policy Gradient

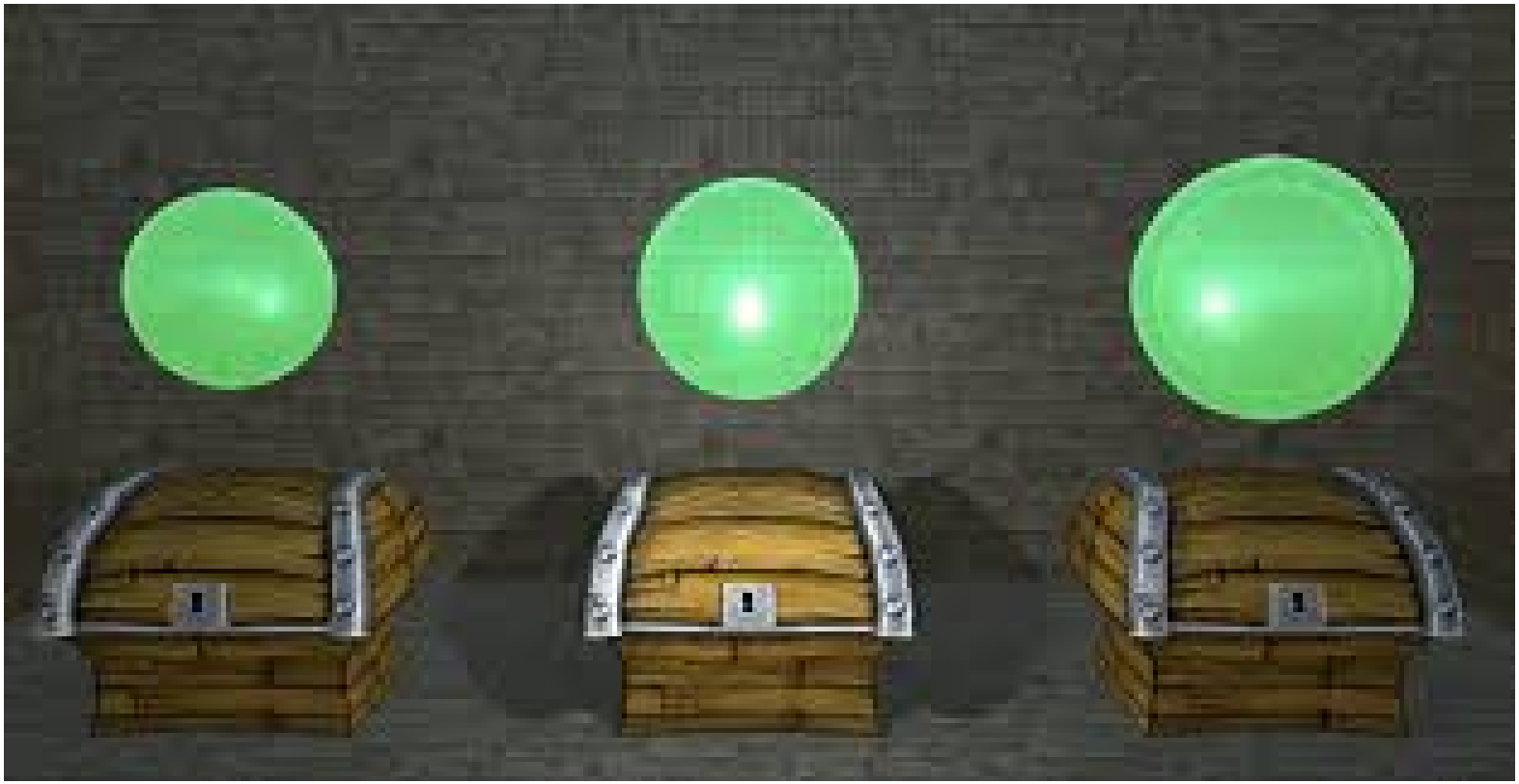
Novosibirsk State University
*THE REAL SCIENCE

Multi-Armed Bandit



Novosibirsk State University
*THE REAL SCIENCE

Multi-Armed Bandit



Novosibirsk State University
*THE REAL SCIENCE

Regret minimization

Bad idea: by the sound of the name
Good idea: by $$$ it brought/lost you

Regret of policy π(a|s):
Consider an optimal policy, π*(a|s)

Regret = sum over training time [optimal – yours]

$$\eta = \sum_t \underset{s,a\sim\pi^*}{E} r(s,a) - \underset{s,a\sim\pi_t}{E} r(s,a)$$

Finite horizon: t < max_t Infinite horizon: t → inf[8]

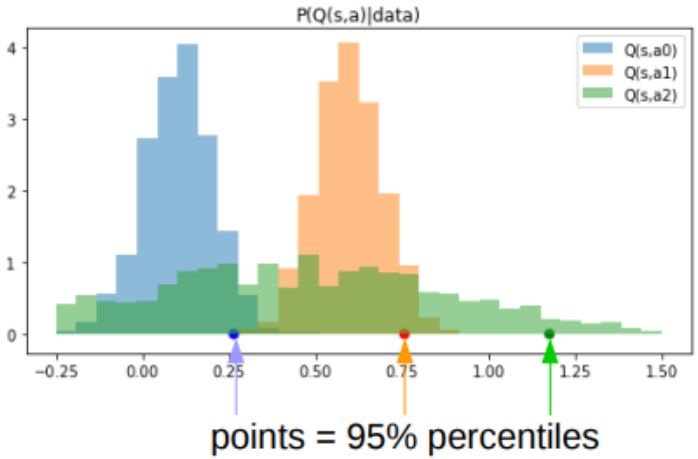
vosibirsk
State University
*THE REAL SCIENCE

- ϵ-greedy regret grows linearly
- UCB and Thompson sampling grows with $log(T)$

Novosibirsk State University
*THE REAL SCIENCE

Optimism in face of uncertainty

- Policy:
  - Compute 95% upper confidence bound *for each a*
  - Take action with highest confidence bound
  - Adjust: change 95% to more/less



Novosibirsk State University
*THE REAL SCIENCE

Policy Optimization

- Stochastic, parametrized policy π_θ
- maximize expected return

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[G_t(\tau)]$$

- SGD with **policy gradient**

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\pi_\theta)|_{\theta_k}$$

Novosibirsk State University
*THE REAL SCIENCE

How to compute policy gradient

- Derive the analitycal gradient
- Compute sample estimate

Probability of a Trajectory

Trajectory

$$\tau = (s_0, a_0, \ldots, s_{T+1})$$

Probability of a Trajectory

$$P(\tau|\theta) = \rho_0(s_0) \prod_{t=0}^{T} P(s_{t+1}|s_t, a_t)\pi_\theta(a_t, s_t)$$

Take log

$$logP(\tau|\theta) =$$

The Log-Derivative Trick

$$\nabla_\theta logP(\tau|theta) = \frac{\nabla_\theta P(\tau|\theta)}{P(\tau|\theta)}$$

Rearrange

$$\nabla_\theta P(\tau|\theta) = P(\tau|\theta)\nabla_\theta logP(\tau|\theta)$$

Novosibirsk State University
*THE REAL SCIENCE

Gradients of Environment Functions

- $\rho_0(s_0)$, $P(s_{t+1}|s_t, a_t)$ and $G_t(\tau)$
- no dependence on θ
- Gradients w.r.t θ are **zero**

Novosibirsk State University
*THE REAL SCIENCE

Grad-Log-Prob of Trajectory

$$\nabla_\theta \log P(\tau|\theta) = \cancel{\nabla_\theta \log \rho_0(s_0)} + \sum_{t=0}^{T} \left(\cancel{\nabla_\theta \log P(s_{t+1}|s_t, a_t)} + \nabla_\theta \log \pi_\theta(a_t|s_t) \right)$$

$$= \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t).$$

Novosibirsk State University
*THE REAL SCIENCE

Basic Policy Gradient

Derivation for Basic Policy Gradient

$$
\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \nabla_\theta \mathop{\mathrm{E}}_{\tau\sim\pi_\theta}[R(\tau)] \\
&= \nabla_\theta \int_\tau P(\tau|\theta)R(\tau) && \text{Expand expectation} \\
&= \int_\tau \nabla_\theta P(\tau|\theta)R(\tau) && \text{Bring gradient under integral} \\
&= \int_\tau P(\tau|\theta)\nabla_\theta \log P(\tau|\theta)R(\tau) && \text{Log-derivative trick} \\
&= \mathop{\mathrm{E}}_{\tau\sim\pi_\theta}[\nabla_\theta \log P(\tau|\theta)R(\tau)] && \text{Return to expectation form} \\
\therefore \nabla_\theta J(\pi_\theta) &= \mathop{\mathrm{E}}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)R(\tau)\right] && \text{Expression for grad-log-prob}
\end{aligned}
$$

Novosibirsk State University
*THE REAL SCIENCE

Basic Policy Gradient in words

- Define policy as parametrized function
- Write the gradient of the loss function
- Play games collect trajectories (1 or more)
- Compute cumulative discounted rewards for each t
- Compute gradients
- Update parameters of your policy
- **Comment: throw out trajectories**

Novosibirsk State University
*THE REAL SCIENCE

Basic Policy Gradient

Algorithm 1 Vanilla Policy Gradient Algorithm

1: Input: initial policy parameters θ_0, initial value function parameters ϕ_0
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
4: Compute rewards-to-go $\hat{R}_t$.
5: Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.
6: Estimate policy gradient as

$$\hat{g}_k = \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)|_{\theta_k} \hat{A}_t.$$

7: Compute policy update, either using standard gradient ascent,

$$\theta_{k+1} = \theta_k + \alpha_k \hat{g}_k,$$

or via another gradient ascent algorithm like Adam.
8: Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_\phi \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

typically via some gradient descent algorithm.
9: **end for**

It is good if you can ...

1. Define Markov Decision Process
2. Describe some task in terms of MDP
3. Understand value and action-value functions
4. Be able to apply Policy Iteration and Value Iteration
5. Understand model-based vs model-free
6. Be able to apply MC, TD(0) algorithms to Q-function
7. Be able to apply Q-learning, SARSA, Expected value SARSA
8. Describe the "Exploitation / Exploration" dilemma
9. Understand ϵ-Greedy and "Optimism in the face of uncertainty" (UCB) ideas
10. Understand the idea of Policy Optimization (Policy Gradient)