Application of stochastic approximation to Cartpole problem

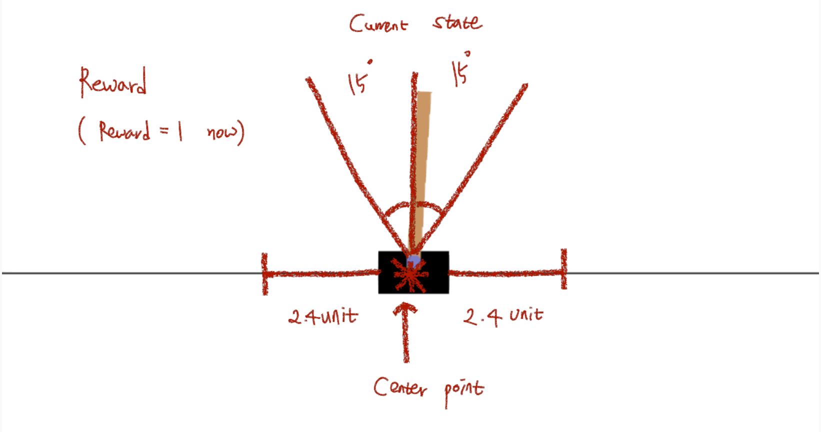
Sang Jun Moon

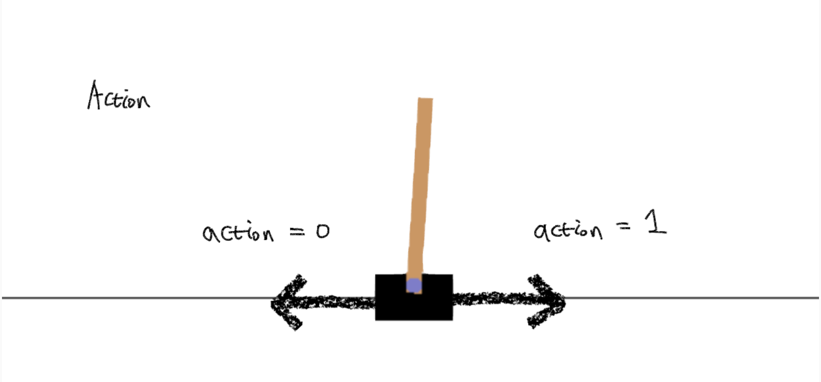
December 12, 2018

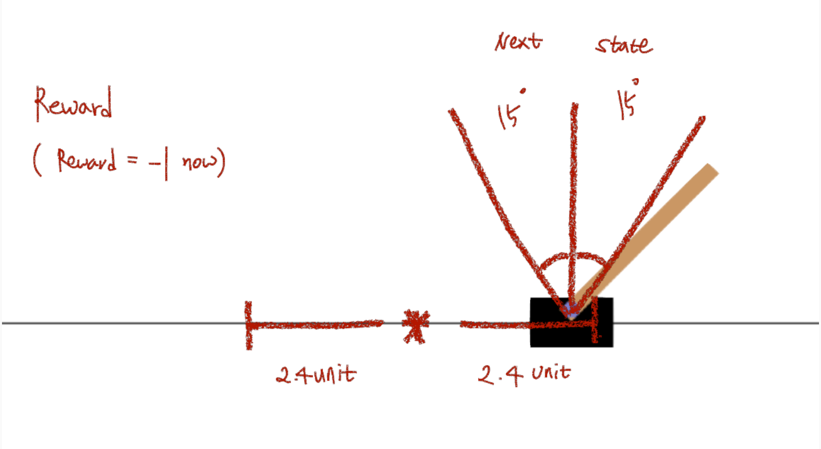
Statistics, University of Seoul

Introduction

Cartpole game




Action
action = 0
action = 1


Reward
(Reward = -1 now)
Next state
15°
15°
2.4 unit
2.4 unit

Approaches using stochastic approximation

Definition of problem

- Let the observation vector and reward value at time t be

$$\mathbf{x}_t \in \mathbb{R}^p \quad \text{and} \quad y_t \in \{-1, 1\}.$$

- It is natural that an action is determined by the current observation.
- In this presentation, the action at time t, $a_t \in \{0, 1\}$ is modeled as follows, which is designed from the idea of logistic regression:

$$a_t = I(x_{t-1}^T \boldsymbol{\beta} \geq 0).$$

  where $\boldsymbol{\beta} \in \mathbb{R}^p$ is a parameter vector.
- Note that the reward value y_t is an unknown function of $\boldsymbol{\beta}$ for given $\mathbf{x}_0$.

Approaches

- The objective function $L(\beta|\mathbf{x}_0)$ is defined as follows:

$$L(\beta|\mathbf{x}_0) = -\sum_{i=1}^{t} y_t$$

- It is impossible to obtain a gradient of the L for the β for given $\mathbf{x}_0$.
- To solve the problem, the following methods are applied:
  - Finite difference stochastic approximation (FDSA)
  - Simultaneous perturbation stochastic approximation (SPSA)
  - Second order SPSA (2SPSA) methods are applied.
- In the practical implementation, $\alpha = 0.602$, $\gamma = 0.101$ are used.
- Also, the gain sequence is:

$$a_k = \frac{1}{(k+1)^\alpha}, \quad c_k = \frac{1}{(k+1)^\gamma}$$

- In 2SPSA, the simultaneous perturbation constant $\tilde{c}_k$ is:

$$\tilde{c}_k = \frac{1}{2(k+1)^\gamma}$$

Additional processing

- The objective function is affected by the initial value of $\mathbf{x}_0$.
- It is a problem to estimate the β that maintains performance even when the initial value of $\mathbf{x}_0$ changes.
- My algorithm changes the initial point and confirm that the performance is maintained after the appropriate β candidate is determined.
- Then, it decides to use β or not.

Algorithms

Initialize $\boldsymbol{\beta}$, S, t, M, $v = 0$.

For iteration from 1 to M:

- Update step:
  - Update $\boldsymbol{\beta}$.
  - Compute $L(\boldsymbol{\beta})$ for given random $\mathbf{x}_0$.
  - Compute $v \leftarrow \min(v, L(\boldsymbol{\beta}))$.
  - If $v = L(\boldsymbol{\beta})$ then $\boldsymbol{\beta}^* \leftarrow \boldsymbol{\beta}$.
  - Else $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta}^*$.
- Stopping criteria:
  - Set $s \leftarrow 1$
  - While $L(\boldsymbol{\beta}^*) = -t$:
    - Compute $L(\boldsymbol{\beta}^*)$ for given $\mathbf{x}_{0,s}$.
    - If $S = s$ return $\boldsymbol{\beta}^*$.
    - $s \leftarrow s + 1$.

Limitations and List of related links

Limitations

- My proposed method doesn't work with the situation when success is rare.
- For example, in MountainCar-v0 problem, the success reward occurs only when the car is at a hill.
- In this case, even if parameters are updated, the value of objective function does not change easily which means the update is ended.

List of related links

Openai-gym

`https://github.com/openai/gym`

Code & Simulation results

`https://github.com/Monster-Moon/gymR`