Acceleration of Reinforcement Learning

Kei Senda, Suguru Hattori
(Kyoto Univ.)

1. Background

Background

EVA by an astronaut example : constructing a space station

The space robots are desired to do dangerous tasks instead of astronauts.

Applying Reinforcement Learning (RL), the robot can complete tasks even if there exists plant uncertainty or variation!

the space robot simulator

Objective

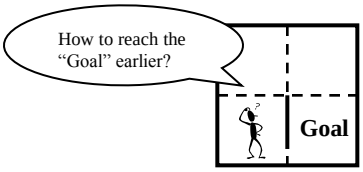
The RL requires numerous computation costs (∝computation times).

⬇

We improve the part of the RL which incurs the computation costs to decrease total ones!

2. The Reinforcement Learning

example : maze problem



your current position = current state

The positive state transition cost is incurred.

For any state, you can select one of four action…
・go UP ・go DOWN ・go LEFT ・go RIGHT

The series of the RL for optimizing a policy

First we take a Sampling.

= to observe statistically what state we move to when we are in each state and select a certain action

After that, we repeat Policy Evaluation and

= to calculate the J-factor
(J-factor : expected cost-to-go using the policy)

Policy Improvement to obtain a sequence of policies.

= to re-obtain the greedy policy judging from the currently calculated J-factor.

Finally we can obtain the optimal policy (= minimizing cost-to-go) by making the policy better!!

The computation cost is a bottleneck of Policy Evaluation but it is repeated until we obtain the optimal policy. ➡ **Make the Policy Evaluation efficient!**

3. Efficient Policy Evaluation

Total computation costs for the Policy Evaluation

=(costs for iterating a J-factor vector at once (itr. cost)
×(the number of iteration for an approximate J-factor vector to obtain the precise solution on Policy Evaluation)
×(the number of undergoing Policy Evaluation step
(= the number of undergoing Policy Improvement step))

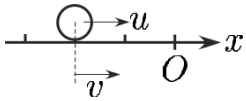
General method versus proposed method

Policy Evaluation is regarded as solving the simultaneous equation for the unknown J-factor vector.
⇒We apply the linear algebra to the Policy evaluation method.

General method
= the existing method based on the Stationery Iterative method

Proposed method ①Itr. cost is twice as many as the SIM.
= the proposed method based on the Krylov Subspace Method

Example : mass controlling and positioning



state = position and velocity(x_t, v_t)
action = input u_t
transition cost = energy function $g_t = \frac{1}{2}\left(x_t^2 + v_t^2 + u_t^2\right)$

Optimal Policy (minimize$J = \sum_t g_t$)
Starting from initial state (x_0, v_0),
the mass moves to the state $(0, 0)$
faster but more modestly!

Result

The number of iteration for each Policy Evaluation Step
(The policy (4) is optimal.)

Policy	General SIM algorim	Proposed KSM algorithm
Initial policy	4900	14
Improved policy (1)	40	11
Improved policy (2)	20	6
Improved policy (3)	10	4
Improved policy (4)	10	2
Total ②	4980	37

Convergence criterion (Relative Residual) $\leq 10^{-3}$

① and ② ➡ Totally, the KSM calculates the optimal policy about 60 times as fast as the SIM!

Consideration

The property of the RL problem
eigenvalue distribution
degenerated solution space
The property of the calculation
An error-decreasing is gradually accelerated.

➡ **The RL has the characteristic structure of accelerating!!**