Infinite Horizon Stochastic Control

1st Aditya Mishra

I. Introduction

This literature deals with the problem of balancing a pendulum at a vertical state. Since its a continuous time problem, the time space and the state space are discretized. Each time step yields a gaussian distribution for the transitional probabilities which makes this problem stochastic. The goal is to implement a quadratic control that guides the pendulum to the zero angle state of the vertical state using control actions and ensure that the pendulum stays at this equilibrium state for an infinite time. The problem is formulated into a Markov Decision Process (MDP) and the mostly likely action for each state is obtained by using value iteration (VI) and policy iteration (PI).

II. Problem formulation

A. State space

The state vector x consists of the angle at which the pendulum is and the angular speed is given by

$$\begin{aligned} x &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ x_1 &\in [-\pi, \pi] \\ x_2 &\in [-v_{max}, v_{max}] \\ x &\in \mathcal{X} \end{aligned} \tag{1}$$

B. Control Space

The control u in control space $\mathcal{U}$ is the action taken to bring the pendulum to equilibrium.

$$u \in [-u_{max}, u_{max}] \tag{2}$$

C. Motion model

The motion of the pendulum is governed by the following equation

$$\begin{aligned} x_{new} &= x + dx \\ dx &= f(x,u)dt + \sigma dw \\ \sigma &= \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} \end{aligned} \tag{3}$$

Here dt is the discretized time step and dw is gaussian random noise with mean as $x + f(x,u)\tau$ and variance as $\sigma\sigma^T\tau$. $f(x,u)$ is given by

$$f(x,u) = \begin{bmatrix} x_2 \\ asin(x_1) - bx_2 + u \end{bmatrix} \tag{4}$$

D. Planning horizon

Let $V(x)$ be the value function of state x. The planning horizon T for this problem will be when

$$\begin{aligned} V_T(x) &= V_{T-1}(x) \\ &\forall x \in \mathcal{X} \end{aligned} \tag{5}$$

E. Stage cost

$$l(x_t, \pi(x_t)) = 1 - \exp(k\cos x_1 - k) + \frac{r}{2}u^2 \tag{6}$$

F. Terminal cost

Terminal cost is 0 $\forall x \in \mathcal{X}$

G. Transition Probability

$$p_f(.|x_t, \pi(x_t)) \sim \mathcal{N}(x + f(x,u)\tau, \sigma\sigma^T\tau) \tag{7}$$

H. The optimization problem

The optimal value function $V^*(x)$ can be found as

$$\begin{aligned} V^*(x) &= \min_{\pi} \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t l(x_t, \pi(x_t))|x_0 = x] \\ \text{s.t. } x_{t+1} &\sim p_f(.|x_t, \pi(x_t)) \\ x_t &\in \mathcal{X} \\ \pi(x_t) &\in \mathcal{U}(x_t) \end{aligned} \tag{8}$$

Here γ is the discount factor. $p_f(.|x_t, \pi(x_t))$ is the transition probability which is a gaussian in this case with ean as $x + f(x,u)\tau$ and variance as $\sigma\sigma^T\tau$. $l(x_t, \pi(x_t))$ is the stage cost given by

$$l(x_t, \pi(x_t)) = 1 - \exp(k\cos x_1 - k) + \frac{r}{2}u^2 \tag{9}$$

I. Interpolation Problem

To find the policy of a continuous state x, the policy of the closest discretized state x_t is chosen for this state. Hence $\pi(x) = \pi(x_t)$

III. Technical approach

A. Policy Iteration

In policy iteration for the policy evaluation step I decided to obtain $V(x)$ by solving the equation using matrix inversion instead of solving the system of equationusing Gauss Seidel.

1) Policy Evaluation:

$V^* \leftarrow (I - \gamma Pr)^{-1} l(x, \pi(x))$
return V^*

Here, Pr is the transition probability matrix of nxn size, $l(x, \pi(x))$ is a vector of $nx1$ size which has all the reward values for each state.

2) Policy Improvement:

for $s \in \mathcal{X}$ **do**
  min_val $\leftarrow$ []
  for $u \in \mathcal{U}$ **do**
    min_val.append($l(s,u)$ + $\gamma \sum_{x' \in \mathcal{X}} p_f(x'|s,u) V^{\pi}(x')$)
  end for
  $\pi(s) \leftarrow \arg\min_{u \in \mathcal{U}}$min_val
end for
return π

B. Value Iteration

In Value iteration the policy improvement is the same as in that of policy iteration. The iteration stops when the values of current time step match the values of previous time step.

1) Policy Improvement:

for $s \in \mathcal{X}$ **do**
  min_val $\leftarrow$ []
  for $u \in \mathcal{U}$ **do**
    min_val.append($l(s,u)$+$\gamma \sum_{x' \in \mathcal{X}} p_f(x'|s,u) V_k(x')$)
  end for
  $\pi(s) \leftarrow \arg\min_{u \in \mathcal{U}}$min_val
end for
return π

2) Value Update:

$\mathrm{V}_{k+1}(x)$ =l(s,u)+ $\gamma \sum_{x' \in \mathcal{X}} p_f(x'|s,u) V_k(x')$
return $\mathrm{V}_{k+1}(x)$

C. Key differences between PI and VI

- The Value Update step of VI is one step of an iterative solution to the linear system of equations in the Policy Evaluation Theorem.
- PI solves the Policy Evaluation equation completely, which is equivalent to running the Value Update step of VI an in
  nite number of times.
- Complexity of VI per Iteration $\mathcal{O}(|\mathcal{X}|^2|\mathcal{U}|)$
- Complexity of PI per Iteration $\mathcal{O}(|\mathcal{X}|^2(|\mathcal{X}| + |\mathcal{U}|))$

D. Policy interpolation method

Let x_c be the continuous time state.

dist_list $\leftarrow$ []
for $x \in \mathcal{X}$ **do**
  dist_list.append($||x - x_c||_2$)
end for
$\pi(x_s) = \pi(\min_x dist\_list)$
return $\pi(x_s)$

IV. Results

A. Effect of problem parameters

Based on tuning done on the parameters for VI and PI, the following conclusions can be mad

- Increasing n_1, n_2 and n_u will decrease the convergence speed as the time complexity of value iteration is $\mathcal{O}(|\mathcal{X}|^2|\mathcal{U}|)$ and for policy iteration $\mathcal{O}(|\mathcal{X}|^2(|\mathcal{X}| + |\mathcal{U}|))$.
- Increasing n_1, n_2 and n_u will make the control more resilient to the system noise also the balancing of pendulum will be better since the size of the state space is larger. Decreasing τ will also have a similar effect.
- Decreasing u_{max} to a lower value will decrease the convergence speed since less control energy is spent at every time step. Decreasing it even further below a certain threshold may not be able to balance the pendulum at the desired equilibrium. The same thing happens with v_{max}
- Increasing γ will decrease the convergence speed. Although this will be helpful in balancing the pendulum and will make the controls more resilient to noise.
- Decreasing k and increasing r will make the system more resilient to noise.
- Decreasing a and b will make the system more resilient to noise and also help in balancing the pendulum.

B. Comparison of $V(x)$

1) Value Iteration:

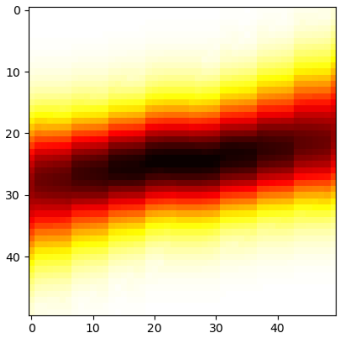

Fig. 1. Value function for value iteration for iteration no.2

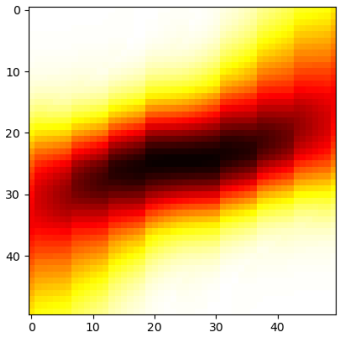

Fig. 2. Value function for value iteration for iteration no.5

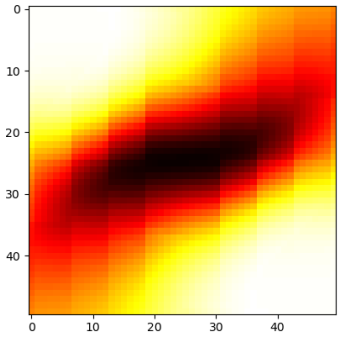

Fig. 3. Value function for value iteration for iteration no.10

2) Policy Iteration:

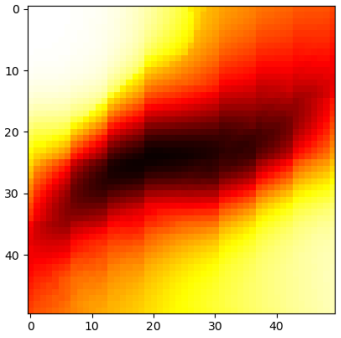

Fig. 4. Value function for policy iteration for iteration no.2

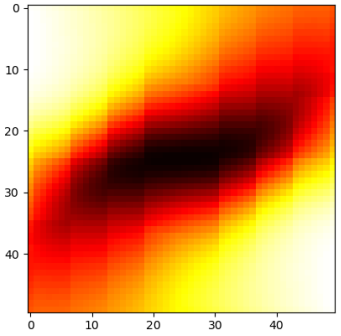

Fig. 5. Value function for policy iteration for iteration no.5

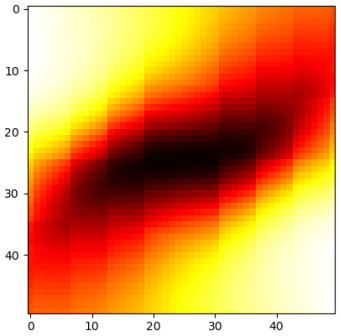

Fig. 6. Value function for policy iteration for iteration no.10

C. Plots of Optimized Policy

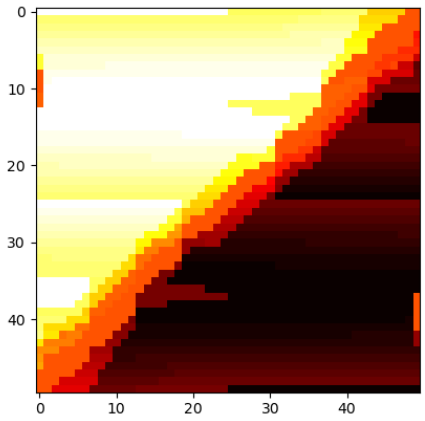

Fig. 7. Optimized policy over discretized states using Policy iteration

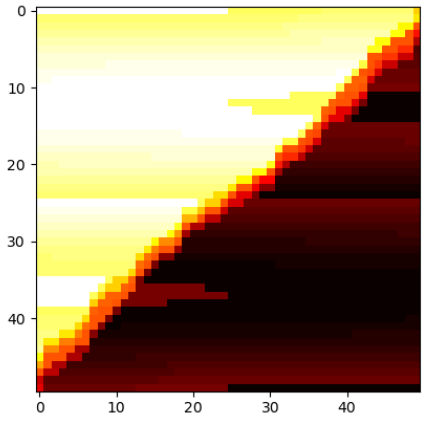

Fig. 8. Optimized policy over discretized states using Value iteration