Adaptive Learning for High-dimensional Hamilton-Jacobi-Bellman Equations

Dec 2022

- Put on arxiv on 2019/11/11.
- Last revise on 2021/2/8.
- The authors are: Tenavi Nakamura-Zimmerer, Qi Gong, Wei Kang.
- Accepted by SISC on 2021.

Problem Formulation

We consider fixed final time optimal control problems (OCP) of the form

$$\begin{cases} \underset{\boldsymbol{u}(\cdot)\in\mathbb{U}}{\text{minimize}} & J[\boldsymbol{u}(\cdot)] = F\left(\boldsymbol{x}\left(t_f\right)\right) + \int_0^{t_f} \mathcal{L}(t,\boldsymbol{x},\boldsymbol{u})dt \\ \text{subject to} & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t,\boldsymbol{x},\boldsymbol{u}) \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0. \end{cases}$$

Here

- $\boldsymbol{x}(t) : [0, t_f] \to \mathbb{X} \subseteq \mathbb{R}^n$ is the state;
- $\boldsymbol{u}(t,\boldsymbol{x}) : [0, t_f] \times \mathbb{X} \to \mathbb{U} \subseteq \mathbb{R}^m$ is the control;
- $J[\boldsymbol{u}(\cdot)]$ is the cost functional, $F\left(x\left(t_f\right)\right) : \mathbb{X} \to \mathbb{R}$ is the terminal cost, $\mathcal{L}(t,\boldsymbol{x},\boldsymbol{u}) : [0, t_f] \times \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ is the running cost.

Open-loop and closed-loop

- For a given initial condition $\boldsymbol{x}(0) = x_0$, many numerical methods exist to compute the optimal open-loop solution,

$$\boldsymbol{u} = \boldsymbol{u}^*\left(t; \boldsymbol{x}_0\right)$$

- Due to various sources of disturbance and real-time application requirements, for practical implementation one typically desires an optimal control in closed-loop feedback form,

$$\boldsymbol{u} = \boldsymbol{u}^*(t, \boldsymbol{x}),$$

which can be evaluated online given any $t \in [0, t_f]$ and a measurement of $\boldsymbol{x} \in \mathbb{X}$.

Value function and HJB equation

- Define the value function $V(t,\boldsymbol{x}) : [0,t_f] \times \mathbb{X} \rightarrow \mathbb{R}$ as the optimal cost-to-go starting at $(t,\boldsymbol{x})$. That is,

$$V(t,\boldsymbol{x}) := J\left[\boldsymbol{u}^*(\cdot)\right] = \begin{cases} \inf_{\boldsymbol{u}(\cdot)\in\mathbb{U}} & F\left(\boldsymbol{y}\left(t_f\right)\right) + \int_t^{t_f} \mathcal{L}(\tau,\boldsymbol{y},\boldsymbol{u})d\tau, \\ \quad \text{s.t.} & \dot{\boldsymbol{y}}(\tau) = \boldsymbol{f}(\tau,\boldsymbol{y},\boldsymbol{u}), \\ & \boldsymbol{y}(t) = \boldsymbol{x}. \end{cases}$$

- It can be shown that the value function is the unique viscosity solution of the Hamilton-Jacobi-Bellman (HJB) PDE,

$$\begin{cases} -V_t(t,\boldsymbol{x}) - \min_{\boldsymbol{u}\in\mathbb{U}} \left\{ \mathcal{L}(t,\boldsymbol{x},\boldsymbol{u}) + \left[V_{\boldsymbol{x}}(t,\boldsymbol{x})\right]^T \boldsymbol{f}(t,\boldsymbol{x},\boldsymbol{u}) \right\} = 0 \\ V\left(t_f,\boldsymbol{x}\right) = F(\boldsymbol{x}) \end{cases}$$

Costate

- Defining the Hamiltonian $\mathcal{H}(t, \boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}) := \mathcal{L}(t, \boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^T \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u})$ where $\boldsymbol{\lambda}(t) : [0, t_f] \to \mathbb{R}^n$ is the costate.
- The optimal control satisfies the Hamiltonian minimization condition,

$$\boldsymbol{u}^*(t) = \boldsymbol{u}^*(t, \boldsymbol{x}; \boldsymbol{\lambda}) = \underset{\boldsymbol{u} \in \mathbb{U}}{\arg\min} \mathcal{H}(t, \boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}).$$

- The optimal feedback control is computed by substituting

$$\lambda(t) = V_{\boldsymbol{x}}(t, \boldsymbol{x})$$

to get

$$\boldsymbol{u}^*(t, \boldsymbol{x}) = \boldsymbol{u}^*\left(t, \boldsymbol{x}; V_{\boldsymbol{x}}\right) = \underset{\boldsymbol{u} \in \mathbb{U}}{\arg\min} \mathcal{H}\left(t, \boldsymbol{x}, V_{\boldsymbol{x}}, \boldsymbol{u}\right)$$

PMP

- Pontryagin's Minimum Principle

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \mathcal{H}_{\boldsymbol{\lambda}} = \boldsymbol{f}\left(t, \boldsymbol{x}, \boldsymbol{u}^*(t, \boldsymbol{x}; \boldsymbol{\lambda})\right), & \boldsymbol{x}(0) = x_0 \\ \dot{\lambda}(t) = -\mathcal{H}_{\boldsymbol{x}}\left(t, \boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}^*(t, \boldsymbol{x}; \boldsymbol{\lambda})\right), & \boldsymbol{\lambda}\left(t_f\right) = F_{\boldsymbol{x}}\left(\boldsymbol{x}\left(t_f\right)\right) \\ \dot{v}(t) = -\mathcal{L}\left(t, \boldsymbol{x}, \boldsymbol{u}^*(t, \boldsymbol{x}; \boldsymbol{\lambda})\right), & v\left(t_f\right) = F\left(\boldsymbol{x}\left(t_f\right)\right) \end{cases} \tag{1}$$

- If we further assume that the solution is optimal, then along the characteristic $\boldsymbol{x}\left(t; \boldsymbol{x}_0\right)$ we have that

$$\begin{aligned} \boldsymbol{u}^*(t, \boldsymbol{x}) &= \boldsymbol{u}^*\left(t; \boldsymbol{x}_0\right), \\ V(t, \boldsymbol{x}) &= v\left(t; \boldsymbol{x}_0\right), \\ V_{\boldsymbol{x}}(t, \boldsymbol{x}) &= \boldsymbol{\lambda}\left(t; \boldsymbol{x}_0\right) \end{aligned}$$

Neural network approximation of the value function

1. Initial data generation;
2. Model training;
3. Adaptive data generation;
4. Model refinement and validation;
5. Feedback control.

Initial data generation

Specifically, by solving the BVP (1) from a set of randomly sampled initial conditions, we get a data set

$$\mathcal{D} = \left\{ \left(t^{(i)}, \boldsymbol{x}^{(i)} \right), V^{(i)} \right\}_{i=1}^{N_d},$$

where $\left(t^{(i)}, \boldsymbol{x}^{(i)} \right)$ are the inputs, $V^{(i)} := V\left(t^{(i)}, \boldsymbol{x}^{(i)} \right)$ are the outputs to be modeled, and $i = 1, 2, \dots, N_d$ are the indices of the data points.

Train the Network

The NN is then trained by solving the nonlinear regression problem,

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{N_d} \sum_{i=1}^{N_d} \left[V^{(i)} - V^{\mathrm{NN}} \left(t^{(i)}, \boldsymbol{x}^{(i)}; \boldsymbol{\theta} \right) \right]^2$$

Physical-informed Learning

Obtain costate data $\boldsymbol{\lambda}(t)$ is for each trajectory as a natural product of solving the BVP. Hence we have the augmented data set,

$$\mathcal{D} = \left\{ \left(t^{(i)}, \boldsymbol{x}^{(i)} \right), \left(V^{(i)}, \boldsymbol{\lambda}^{(i)} \right) \right\}_{i=1}^{N_d},$$

where $\boldsymbol{\lambda}^{(i)} := \boldsymbol{\lambda}\left(t^{(i)}; \boldsymbol{x}^{(i)} \right)$.

Physical-informed Learning

We now define the physics-informed learning problem,

$$\text{minimize}_{\boldsymbol{\theta}}\ \text{loss}(\boldsymbol{\theta};\mathcal{D}) := \underset{V}{\text{loss}}(\boldsymbol{\theta};\mathcal{D}) + \mu \cdot \text{loss}_{\boldsymbol{\lambda}}(\boldsymbol{\theta};\mathcal{D})$$

Here $\mu \geq 0$ is a scalar weight, the loss with respect to data is

$$\underset{V}{\text{loss}}(\boldsymbol{\theta};\mathcal{D}) := \frac{1}{N_d}\sum_{i=1}^{N_d}\left[V^{(i)} - V^{\mathrm{NN}}\left(t^{(i)},\boldsymbol{x}^{(i)};\boldsymbol{\theta}\right)\right]^2$$

and the gradient regularization is defined as

$$\underset{\boldsymbol{\lambda}}{\text{loss}}(\boldsymbol{\theta};\mathcal{D}) := \frac{1}{N_d}\sum_{i=1}^{N_d}\left\|\boldsymbol{\lambda}^{(i)} - V_{\boldsymbol{x}}^{\mathrm{NN}}\left(t^{(i)},\boldsymbol{x}^{(i)};\boldsymbol{\theta}\right)\right\|^2$$

Adaptive

"In this paper, we concentrate samples where $\left\| V_{\boldsymbol{x}}^{\mathrm{NN}}(\cdot) \right\|$ is large. Regions of the **value function with large gradients tend to be steep or complicated**, and thus may benefit from having more data to learn from."

Well......

Adaptive

Algorithm 4.1 Adaptive sampling and model refinement

```
Generate $\mathcal{D}^1_{\text{train}}$ using time-marching
for $r = 1, 2, \dots$ do
    Solve (3.3) for $\boldsymbol{\theta}$
    if (4.8) is satisfied then
        return optimized parameters $\boldsymbol{\theta}$ and NN validation accuracy
    else
        while (4.9) is not satisfied do
            Sample candidate initial conditions $\boldsymbol{x}_0^{(i)}$, $i = 1, \dots, N_c$
            In parallel, predict $\left\| V_{\boldsymbol{x}}^{\text{NN}} \left(0, \boldsymbol{x}_0^{(i)}\right) \right\|$, $i = 1, \dots, N_c$
            Choose the initial condition(s) with largest predicted gradient norm and
            use NN warm start to solve the corresponding BVP(s) (2.11)
            Add the resulting trajectorie(s) to $\mathcal{D}^{r+1}_{\text{train}}$
        end while
    end if
end for
```

Solving the control

- Suppose that the system dynamics can be written in the form

$$\dot{x} = \boldsymbol{f}(t, \boldsymbol{x}) + \boldsymbol{g}(t, \boldsymbol{x})\boldsymbol{u},$$

  where $\boldsymbol{f}(t, \boldsymbol{x}) : [0, t_f] \times \mathbb{X} \to \mathbb{R}^n, \boldsymbol{g}(t, \boldsymbol{x}) : [0, t_f] \times \mathbb{X} \to \mathbb{R}^{n \times m}$, and the control is unconstrained.
- Further, suppose that the running cost is of the form

$$\mathcal{L}(t, \boldsymbol{x}, \boldsymbol{u}) = h(t, \boldsymbol{x}) + \boldsymbol{u}^T \boldsymbol{W} \boldsymbol{u},$$

  for some convex function $h(t, \boldsymbol{x}) : [0, t_f] \times \mathbb{X} \to \mathbb{R}$ and some positive definite weight matrix $\boldsymbol{W} \in \mathbb{R}^{m \times m}$.

Solving the control

- Then the Hamiltonian is

$$\mathcal{H}(t, \boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}) = h(t, \boldsymbol{x}) + \boldsymbol{u}^T \boldsymbol{W} \boldsymbol{u} + \boldsymbol{\lambda}^T \boldsymbol{f}(t, \boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(t, \boldsymbol{x}) \boldsymbol{u}.$$

- Now we apply PMP, which for unconstrained control requires

$$\mathbf{0}_{m \times 1} = \mathcal{H}_{\boldsymbol{u}}(t, \boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}^*) = 2 \boldsymbol{W} \boldsymbol{u}^* + \boldsymbol{g}^T(t, \boldsymbol{x}) \boldsymbol{\lambda}.$$

- Letting $\boldsymbol{\lambda} = V_{\boldsymbol{x}}(t, \boldsymbol{x})$ and solving for $\boldsymbol{u}^*$ yields the optimal feedback control law in explicit form:

$$\boldsymbol{u}^*(t, \boldsymbol{x}; V_{\boldsymbol{x}}) = -\frac{1}{2} \boldsymbol{W}^{-1} \boldsymbol{g}^T(t, \boldsymbol{x}) V_{\boldsymbol{x}}(t, \boldsymbol{x})$$

Validation

- The relative mean absolute error (RMAE)

$$\mathrm{RMAE}\left(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{val}}\right) := \frac{\sum_{i=1}^{N_d} \left| V^{(i)} - V^{\mathrm{NN}}\left(t^{(i)}, \boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right)\right|}{\sum_{i=1}^{N_d} \left| V^{(i)} \right|}$$

- Relative mean L^2 error

$$\mathrm{RM}\, L^2\left(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{val}}\right) := \frac{\sum_{i=1}^{N_d} \left\| \boldsymbol{\lambda}^{(i)} - V_{\boldsymbol{x}}^{\mathrm{NN}}\left(t^{(i)}, \boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right)\right\|_2}{\sum_{i=1}^{N_d} \left\| \boldsymbol{\lambda}^{(i)} \right\|_2}$$