高维 PDE 求解算法:
从非线性蒙特卡罗到机器学习

刘行

2025 年 3 月 26 日

大纲

引言与背景

- 高维偏微分方程 (PDEs) 在控制理论，金融工程，量子力学等领域有重要应用
- 传统数值方法 (如有限差分，有限元) 在高维情况下受 “维数灾难” 限制
- 本论文提出利用非线性蒙特卡罗方法和深度学习技术求解高维 PDE，从而突破维数灾难

深度 BSDE 方法 (Deep BSDE)

- 将非线性抛物型 PDE

$$\frac{\partial u}{\partial t} + \frac{1}{2}\mathrm{Tr}\left(\sigma\sigma^\top \mathrm{Hess}_x u\right) + \langle \nabla u, \mu \rangle + f\left(t, x, u, \sigma^\top \nabla u\right) = 0, \quad u(T, x) = g(x)$$

与后向随机微分方程 (BSDE) 联系起来

- 通过 Itô 引理可得:

$$u(t, X_t) - u(0, X_0) = -\int_0^t f\left(s, X_s, u(s, X_s), \sigma^\top \nabla u(s, X_s)\right) \mathrm{d}s + \int_0^t (\nabla u(s, X_s))^\top \sigma(s, X_s)\, \mathrm{d}W_s.$$

深度 BSDE 方法 (Deep BSDE)

- Pardoux 和 Peng 提出如果令 $Y_t = u(t, X_t)$, $Z_t = [\sigma(t, X_t)]^\top (\nabla_x u)(t, X_t)$, 则随机过程 $(X_t, Y_t, Z_t) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d$, $t \in [0, T]$, 满足下面的 BSDE:

$$\begin{cases} X_t = \xi + \int_0^t \mu(s, X_s)\,\mathrm{d}s + \int_0^t \sigma(s, X_s)\,\mathrm{d}W_s \\ Y_t = g(X_T) + \int_t^T f(s, X_s, Y_s, Z_s)\,\mathrm{d}s - \int_t^T (Z_s)^\top\,\mathrm{d}W_s \end{cases}$$

- 利用深度神经网络逼近未知函数: 例如用网络 ψ_0 表示 $u(0, X_0)$, 用子网络 ϕ_n 逼近 Z_t. 通过离散化时间构建网络, 将末端误差作为损失函数进行训练.

- 问题最终转化为

$$
\begin{aligned}
&\inf_{\psi_0,\{\phi_n\}_{n=0}^{N-1}} \mathbb{E}\left|g\left(X_T\right)-Y_T\right|^2, && (1)\\
&s.t. \quad X_0=\xi, \quad Y_0=\psi_0(\xi), && (2)\\
&\qquad X_{t_{n+1}}=X_{t_i}+\mu\left(t_n, X_{t_n}\right) \Delta t+\sigma\left(t_n, X_{t_n}\right) \Delta W_n, && (3)\\
&\qquad Z_{t_n}=\phi_n\left(X_{t_n}\right), && (4)\\
&\qquad Y_{t_{n+1}}=Y_{t_n}-f(t_n, X_{t_n}, Y_{t_n}, Z_{t_n}) \Delta t+\left(Z_{t_n}\right)^{\top} \Delta W_n && (5)
\end{aligned}
$$

- 取误差函数为

$$
l(\theta)=\mathbb{E}\left[\left|g\left(X_{t_N}\right)-\hat{u}\left(\left\{X_{t_n}\right\}_{0 \leq n \leq N},\left\{W_{t_n}\right\}_{0 \leq n \leq N}\right)\right|^2\right].
$$

其中 $\hat{u}\left(\left\{X_{t_n}\right\}_{0 \leq n \leq N},\left\{W_{t_n}\right\}_{0 \leq n \leq N}\right)$ 为网络的最后一个输出，为 $u\left(t_N, X_{t_N}\right)$ 的近似

高维控制问题

- 传统控制理论中，最优控制问题的解往往受到维数灾难的限制
- 我们考虑下面的最优控制问题:

$$\min_{u} g\left(x\left(T\right)\right)+\int_{0}^{T} L\left(t, x\left(t\right), u\left(t, x\left(t\right)\right)\right) \mathrm{d}t$$

  受限于动态系统

$$\dot{x}=f(t, x, u) \quad x\left(0\right)=x_0$$

- 对于开环控制，仅沿最优路径定义控制函数 $u\left(t, x\right)=u^{*}\left(t, x^{*}\left(t\right)\right)$. 其自变量仅为时间，维数灾难并不是主要问题
- 对于闭环控制，控制函数 $u\left(t, x\right)=u^{*}\left(t, x\right)$ 在全局状态空间定义，维数灾难成为主要问题

高维控制问题

- Pontryagin 极小原理给出了最优控制问题的理论基础
- 令扩展哈密顿量为 $\tilde{H} = L + \lambda^\top f$, $u^*(t, x; \lambda) = \arg\min_{u \in U} \tilde{H}(t, x, \lambda, u)$ 则由 Pontryagin 极小原理, 下面的方程组有解

$$\begin{cases} \dot{x} = \tilde{H}_\lambda \Big|_{u=u^*} & x(0) = x_0 \\ \dot{\lambda} = -\tilde{H}_x \Big|_{u=u^*} & \lambda(T) = \nabla g(x(T)) \\ \dot{v} = -L\big|_{u=u^*} & v(T) = g(x(T)) \end{cases}$$

其中, x 为状态变量, λ 为伴随变量, $\tilde{H}$ 为 Hamilton 函数

高维控制问题

- 记

$$V(t,x)=\inf_{u\in U}\{g(y(T))+\int_t^T L(\tau,y,u)\,\mathrm{d}\tau\}$$

其中 $\dot{y}(\tau)=f(\tau,y,u)\,,y(t)=x,$

$$H^*(t,x,\lambda)=\tilde{H}(t,x,\lambda,u^*(t,x;\lambda))$$

则 V 满足 HJB 方程 $V_t+H^*(t,x,V_x)=0$, 且有终端条件 $V(T,x)=g(x)$ 通过数值方法求解 HJB 方程, 可以得到最优控制函数 u^*

高维控制问题

- 想通过机器学习方法求解高维控制问题，需要推广到所有初值条件．于是考虑如下问题

$$\min_{u} \mathbb{E}_{x_0 \sim \mu} \left[g\left(x\left(T\right)\right) + \int_0^T L\left(t, x\left(t\right), u\left(t, x\left(t\right)\right)\right) \mathrm{d}t \right]$$

$$\dot{x} = f\left(t, x, u\right), \quad x\left(0\right) = x_0, \quad \mu = \frac{1}{Z} e^{-\beta V}$$

- 为解决数据生成问题，需要采用如下策略：
  1. 求解前面的两点边值问题来生成训练数据
  2. 利用生成的数据训练神经网络，以逼近值函数 V 和最优控制函数 u^*
- 然而，这个两点边值问题的求解也不简单，具体可以采用 "热启动" 策略或者自适应采样方法．

Deep Ritz Method

- 考虑如下变分问题:

$$\min_{u\in H} I(u), \quad I(u) = \int_\Omega \left(\frac{1}{2}|\nabla u(x)|^2 - f(x)\,u(x)\right) \mathrm{d}x$$

- Deep Ritz 方法主要包括:
  1. 使用 DNN 参数化试探函数
  2. 对 $I(u)$ 采取数值积分进行离散化
  3. 设计最终算法
- 处理边界条件时可能会遇到一些问题, 可以对泛函进行修正:

$$I(u) = \int_\Omega \left(\frac{1}{2}u_x^2 - fu\right) \mathrm{d}x + \beta \int_{\partial\Omega} u^2 \, \mathrm{d}x$$

通常可选 $\beta = 500$.

最小二乘法

- 考虑在区域 $\Omega \in \mathbb{R}^d$ 上求解 PDE

$$Lu = f$$

- 可等价的转化为变分问题

$$\min_{u} J(u) = \int_{\Omega} \|Lu - f\|^2 \mu(\mathrm{d}x)$$

其中 μ 是 Ω 上选取的非退化易于采样的概率分布. 于是问题与前面变得类似.

Galerkin 方法

- 考虑弱形式: 寻找 $u \in H_1$ 使得对任意 $\varphi \in H_2$ 有

$$a(u, \varphi) = \langle Lu, \varphi \rangle = \langle f, \varphi \rangle$$

- 通常进行分部积分使其仅涉及一阶导数.
- 类似 WGAN 方法, 可以将其改写为

$$\min_{u \in H_1} \max_{\|\varphi\|_{H_2} \leq 1} \left(a(u, \varphi) - \langle f, \varphi \rangle\right)^2$$

- 然而这种形式的公式很难求解

多层皮卡德方法 (MLP)

- 定理 3 主要提供了多层 Picard 近似方法 (MLP 方法) 在求解半线性热偏微分方程 (具有 Lipschitz 连续的非线性项) 时的误差估计和计算复杂度分析.
- 假设 PDE 由半线性热方程

$$\frac{\partial}{\partial t} u_d(t,x) = \Delta_x u_d(t,x) + f(u_d(t,x))$$

给出, 其中 $u_d(t,x)$ 是精确解, $f(u)$ 是 Lipschitz 连续的非线性项.
- 设定精度阈值 ε, 则 MLP 方法所需的计算成本 $\mathfrak{C}_{d,\mathfrak{N}_\varepsilon,\mathfrak{N}_\varepsilon}$ 满足:

$$\mathfrak{C}_{d,\mathfrak{N}_\varepsilon,\mathfrak{N}_\varepsilon} \leq c d^c \varepsilon^{-3},$$

其中 c 是一个与维数 d 和精度 ε 无关的常数. 这表明 MLP 方法的计算成本仅呈多项式增长, 相比于传统方法的指数增长 (即 “维数灾难”), MLP 方法在一定程度上克服了维数灾难.

多层皮卡德方法 (MLP)

- 该方法的 L^2 误差估计满足:

$$\left(\mathbb{E}\left[\left|U^{d,0}_{\mathfrak{N}_\varepsilon,\mathfrak{N}_\varepsilon}(T,0)-u_d(T,0)\right|^2\right]\right)^{\frac{1}{2}}\leq\varepsilon$$

  这意味着, MLP 方法可以在计算成本受控的情况下, 将误差控制在 ε 以内.
- 定理 3 的核心贡献
  1. 提供了 MLP 方法对 PDE 逼近的严格误差分析, 证明了该方法能够在多项式计算复杂度下达到高精度解
  2. 明确了计算复杂度的上界, 即计算成本随维度 d 线性增长, 随误差 ε^{-3} 级增长
  3. 支持 MLP 方法在高维 PDE 计算中的可行性, 理论上证明了该方法在高维问题中比传统网格方法 (如有限差分, 有限元) 更具优势
- 总的来说, 定理 3 证明了 MLP 方法是一种在计算效率和逼近精度之间具有良好平衡的数值方法, 特别适用于高维 PDE 计算

神经网络近似逼近 PDEs 数值解的数学结果

- 截至今日，尚没有完整的严格数学分析能够证明 (或证伪) 以下猜想：存在一种基于深度学习的近似方法能够在偏微分方程 (PDE) 的数值逼近中克服维度诅咒 (curse of dimensionality). 然而，已经有一些数学结果证明，DNN 具备在不引入维度诅咒的条件下近似 PDE 解的能力.
- 定理 4 表明，设 $u_d\,(t,x)$ 满足非线性热方程:

$$\frac{\partial u_d}{\partial t}=\Delta_x u_d+f(u_d),\quad u_d(0,x)\text{初值}$$

若初值可用 DNN 以多项式复杂度逼近，则

1. 存在神经网络 $\mathfrak{u}_{d,\varepsilon}$，其参数数量 $\mathcal{P}\left(\mathfrak{u}_{d,\varepsilon}\right)$ 至多以多项式速率增长于维度 d 和精度倒数 ε^{-1}
2. 在超立方体 $[0,1]^d$ 上，神经网络实现的 L^2-逼近误差不超过 ε

结论与未来展望

- 利用非线性蒙特卡罗和深度学习方法，设计出不受维数灾难影响的高效数值算法
- 理论与数值实验均证明：对于控制，金融，量子等领域的高维问题，这些方法具有显著优势
- 未来工作：进一步完善理论证明，改进算法效率以及扩展到更广泛的应用场景

结束页

谢谢！