Combatting distribution shift in SciML

Jiaxi Zhao

20th Mar, 2024

An abstraction of SciML workflow

Simulating the dynamics:

$$\mathbf{0} = \mathcal{L}(\mathbf{x}, \partial_t \mathbf{x}, \mathbf{u}, t), \quad \mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U}, \mathcal{L} : \mathcal{X} \times T\mathcal{X} \times \mathcal{U} \times \mathbb{R}_+ \to \mathbb{R}^c,$$
$$\mathbf{u} = \phi(\mathbf{x}, t), \quad \phi : \mathcal{X} \times \mathbb{R}_+ \to \mathbf{u}.$$

1. $\mathcal{L}$ is known, possibly non-linear.
2. ϕ is un-known.
3. A set of data pairs
$\{(\mathbf{x}_1, \mathbf{u}_1, t_1), (\mathbf{x}_2, \mathbf{u}_2, t_2), \cdots, (\mathbf{x}_N, \mathbf{u}_N, t_N)\}$.
4. Benchmark algorithm solves the ordinary least square:

$$\arg\min_{\theta} \mathbb{E} \left\| \mathbf{u} - \phi_\theta(\mathbf{x}, t) \right\|^2.$$

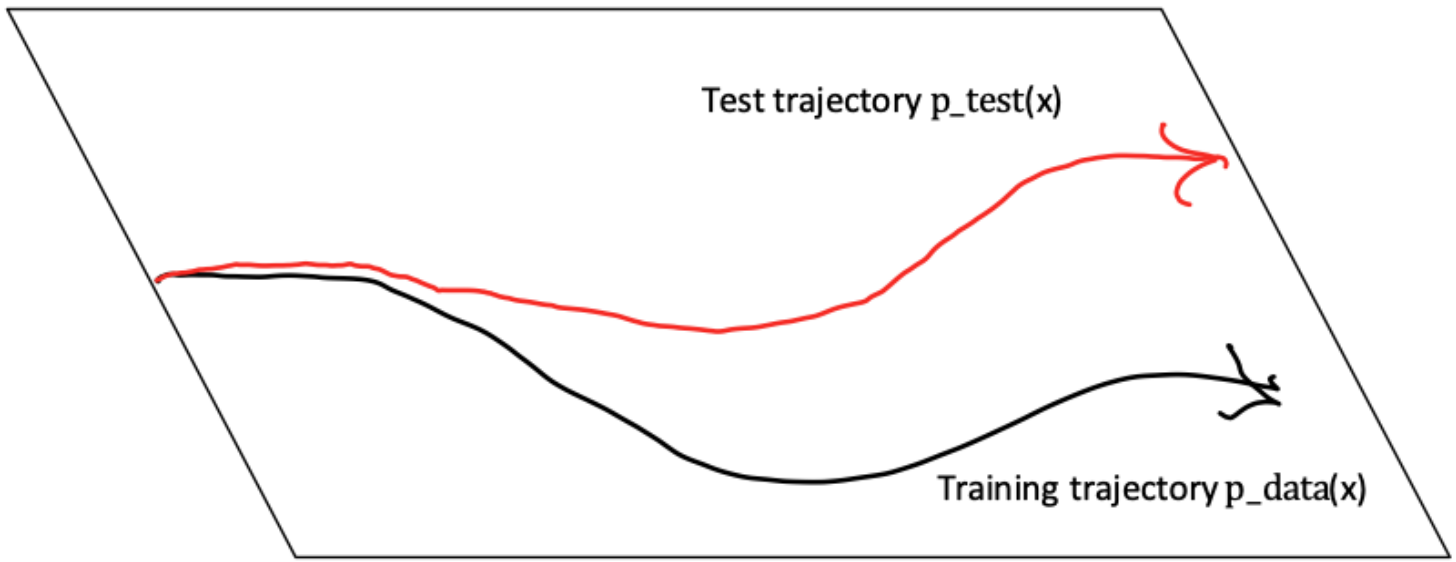
Iterative scheme is ubiquitous

There are different levels of iterative methods in scientific application:

1. High level: iterative method for forward and inverse problem
    * Steady simulation: RANS.
    * Unsteady simulation: LES, Molecular dynamics.
    * Optimization for inverse problem: inverse scattering.
2. Low level: iterative method to solve the linear system:
    * Jacobi, Gauss-Seidel, SOR etc.
    * Connjugate gradient, GMRES, BiCGSTAB etc.
    * Preconditioning, e.g. ILU, AMG etc.

Dilemma of data-driven scientific computing

In the data-driven scientific computing, **dynamics structure** can cause **distribution mismatch** between the training and testing data.



Data generation:

- Preprocessing, e.g. filterd DNS data compared with LES data
- Adaptive sampling & Active learning
- Importance reweighting

Learning algorithm:

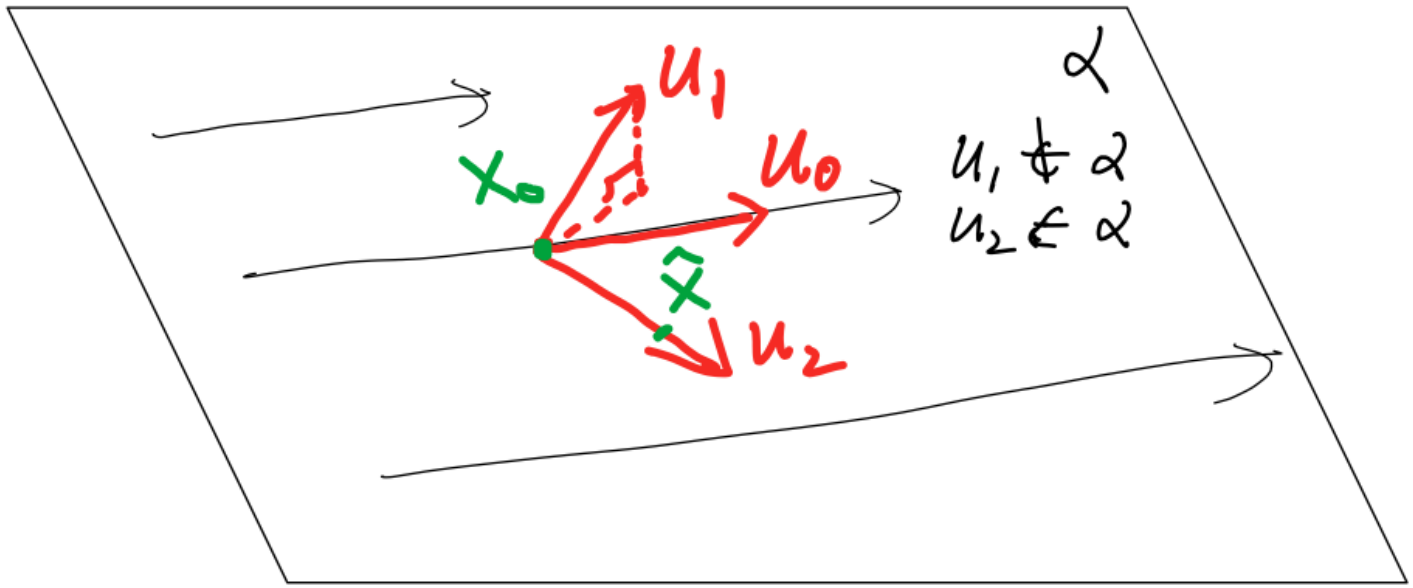
- Manifold-regularized learning
- Unrolled dynamics and reinforcement learning

Simulated algorithm:

- Subspace-aware algorithm
- Random subspace method

Manifold regularization

Regularization encodes the information of the data manifold.



The freedom of choosing simulated dynamics

For unsteady simulation problem, we can change the numerical scheme for simulation, i.e. spatial and temporal discretization. While for steady state simulation and inverse problem, we can choose or design the numerical scheme:

$$\begin{aligned} \mathbf{0} &= \mathcal{L}_\theta(\mathbf{u}), \\ \partial_t \mathbf{u} &= \mathcal{L}_\theta(\mathbf{u}), \quad \text{(RANS simulation)} \\ \partial_t \mathbf{u} &= -\left(\nabla_u \mathcal{L}_\theta(\mathbf{u})\right)^{-1} \mathcal{L}_\theta(\mathbf{u}) \quad \text{(Gauss Newton)}. \end{aligned} \tag{1}$$

The question is: *How to go beyond linear?*