Data-driven parameterizations in numerical models using data assimilation and machine learning.

Julien Brajard[1,2], Alberto Carrassi[3,4], Marc Bocquet[5], Laurent Bertino[1]

May 4, 2020

[1]NERSC, [2]Sorbonne Université, [3]University of Reading, [4]Unversity of Utrecht, [5]École des Ponts ParisTech

Objective of this work

The objective is to produce a hybrid (physical/data-driven) model

$$\mathbf{x}(t+\delta t) = \mathcal{M}^{\varphi}[\mathbf{x}(t)] + \mathcal{M}^{\mathrm{UN}}[\mathbf{x}(t)],$$

where:

- $\mathbf{x}(t)$ is the state of the dynamical system
- $\mathcal{M}^{\varphi}$ is the physical model (assumed to be known a priori)
- $\mathcal{M}^{\mathrm{UN}}$ is the unresolved component of the dynamics to be inferred from data
- δt is the integration time step

$\mathcal{M}^{\mathrm{UN}}$ is approximated by a data-driven model represented under the form of a neural network whose parameters are $\boldsymbol{\theta}$: $\mathcal{M}_{\boldsymbol{\theta}}[\mathbf{x}(t)]$

What is known:

- Observations of the system $\mathbf{y}_k = H(\mathbf{x}_k) + \boldsymbol{\epsilon}_k^{\mathrm{obs}}$
- The observation operator H and observation noise statistics
- The physical model $\mathcal{M}^{\varphi}$

What is to be determined (unknown):

- The state of the system $\mathbf{x}_k = \mathbf{x}(t_k)$ $(0 \leq k \leq K)$;
- The neural networks and its associated parameters $\mathcal{M}_{\boldsymbol{\theta}}$.

Proposed approach

Observation Setup

Observations $\mathbf{y}_k$ are assumed to be made at each Δt time step such as $\Delta t = N_c \delta t$ (N_c is a positive integer and δt is the integration time step).

Simplified description of the algorithm:

1. Estimating the state $\mathbf{x}^{\mathrm{a}}_{1:K}$. At each time t_k, we calculate a forecast $\mathbf{x}^{\mathrm{f}}$:

$$\mathbf{x}^{\mathrm{f}}_{k+1} = \mathbf{x}^{\mathrm{f}}(t_k + \Delta t) = (\mathcal{M}^{\varphi})^{N_c}(\mathbf{x}^{\mathrm{a}}_k)$$

   An observation $\mathbf{y}_{k+1}$ is assimilated to produced an analysis state $\mathbf{x}^{\mathrm{a}}_{k+1}$

2. Determining an estimation of the unknown part of the model. We assume that:
   - $\mathbf{x}(t + \Delta t) \approx (\mathcal{M}^{\varphi})^{N_c}(\mathbf{x}(t)) + N_c \cdot \mathcal{M}^{\mathrm{UN}}[\mathbf{x}(t)]$
   - $\mathbf{x}(t) \approx \mathbf{x}^{\mathrm{a}}(t)$

   We consider that $\mathcal{M}^{\mathrm{UN}}(\mathbf{x}_k) \approx \mathbf{z}_{k+1} = 1/N_c \cdot \left(\mathbf{x}^{\mathrm{a}}_{k+1} - \mathbf{x}^{\mathrm{f}}_{k+1}\right)$

3. Training a neural network $\mathcal{M}_{\boldsymbol{\theta}}$ by minimizing the loss $L(\boldsymbol{\theta}) = \sum_{k=0}^{K-1} ||\mathcal{M}_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{a}}_k) - \mathbf{z}_{k+1}||^2$

4. Using the hybrid model $\mathcal{M}^{\varphi} + \mathcal{M}_{\boldsymbol{\theta}}$ to produce new simulations (e.g. to make forecasts).

Numerical experiment setup

We illustrate the algorithm using the Lorenz 2-scale model

- The "unknown" model (to be represented by a neural network)
- The physical model (assumed to be known)

$$\frac{\mathrm{d}x_n}{\mathrm{d}t} = \psi_n^+(\mathbf{x}) + F - h\frac{c}{b}\sum_{m=0}^{9} u_{m+10n},$$

$$\frac{\mathrm{d}u_m}{\mathrm{d}t} = \frac{c}{b}\psi_m^-(b\mathbf{u}) + h\frac{c}{b}x_{m/10},$$

$$\psi_n^\pm(\mathbf{x}) = x_{n\mp1}(x_{n\pm1} - x_{n\mp2}) - u_n,$$

$n = 0, \cdots, N_x - 1$ $(N_x = 36)$, $m = 0, \cdots, N_u - 1$ $(N_u = 360)$, $(c, b, h, F) = (10, 10, 1, 10)$.

Data generation

The full model (+) is integrated using RK4 scheme with an integration time step $\delta t = 0.005$ to generate the true field $\mathbf{x}_{0:K}$. The observations are produced at each $\Delta t = 3 \cdot \delta t$ time steps by perturbing the true field with a centered gaussian of standard deviation $\sigma_{\mathrm{obs}} = 1$.

Algorithm setup

Data assimilation

We use a square-root ensemble Kalman smoother with a ensemble of size 50, a multiplicative inflation of 1.08 an additive noise of 0.06 at each time step δt and a lag of 12 time steps.
https://github.com/nansencenter/DAPPER

Neural network

The neural network is composed of 3 convolutional layers. Hyperparameters (size of each layer, batchsize, optimizer, regularization, ...) are determined via Bayesian optimisation (*hypertopt* package).

For an upper bound hybrid model, we train a additional neural net with "true data" ($\mathbf{x}_k^{\mathrm{a}} = \mathbf{x}_k$).
https://keras.io/

Preliminary results

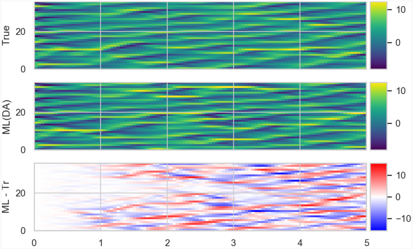

Figure 1: Trajectory of the true model and the hybrid model with noisy observations.

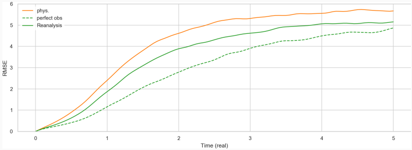

Figure 2: RMSE of the forecast of the physical model (orange) and the hybrid model (green) with perfect (dashed line) and noisy (plained line) observations.

Conclusion and discussion

- We introduced an algorithm to learn the unknown part of a numerical model from data and to combine it with a known physical part (whose adjoint may not be known). The algorithm was illustrated with the Lorenz 2-scale model.
- This algorithm relies on data assimilation and machine learning techniques. DA is instrumental to handle partial and noisy data that then inform the ML process.
- The proposed learning algorithm used algorithms that have individually proved their efficiency for high-dimensional systems.
- The hybrid model produced can be expensive to compute because of, e.g., different computing requirement (CPU multiprocessors vs GPU).
- Our approach is able to handle two main issues arising in realistic applications: (i) one has not generally access to observations of the model tendencies, and, (ii) observations are available at a lower frequency than the model computational time-step.
- The approach relies on the fact that the dynamics is well sampled in time by the observations, so only the larger time scale variability (relatively to Δt) are properly learnt by the neural network.

Reference

Bocquet, M., Brajard, J., Carrassi, A., and Bertino, L. (2019).
Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models.
Nonlinear Processes in Geophysics, 26(3):143–162.

Bocquet, M., Brajard, J., Carrassi, A., and Bertino, L. (2020).
Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization.
Foundations of Data Science, 2(1):55.

Brajard, J., Carassi, A., Bocquet, M., and Bertino, L. (2020).
Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the Lorenz 96 model.
arXiv preprint arXiv:2001.01520.

julien.brajard@nersc.no