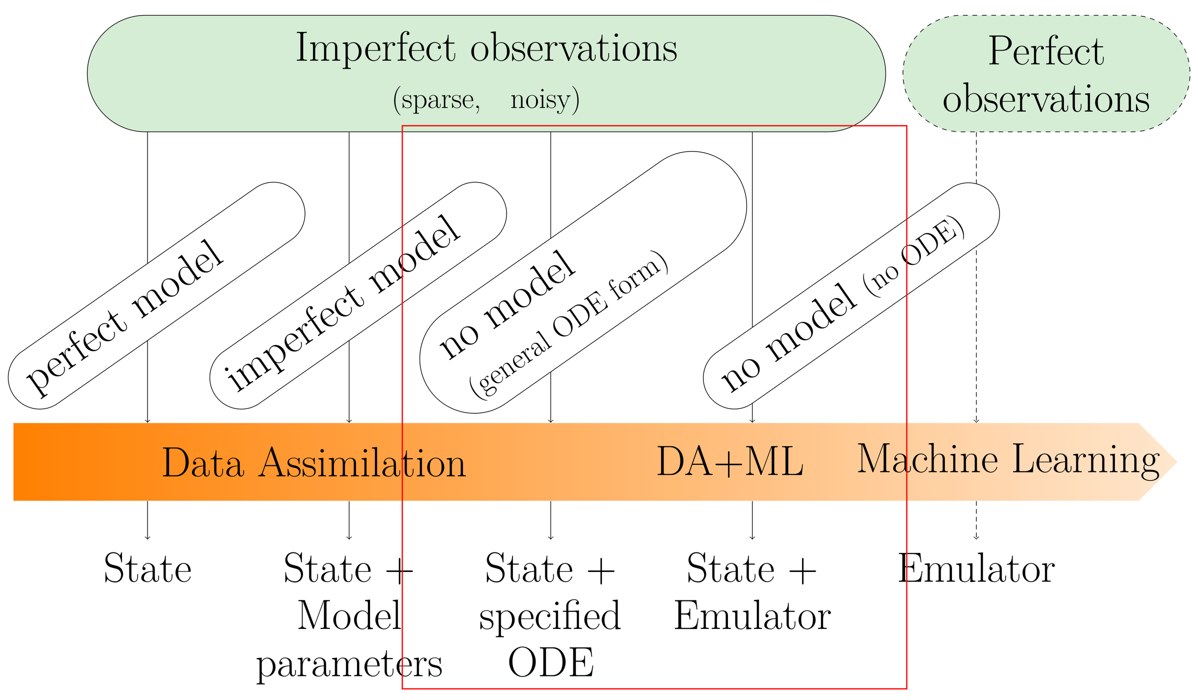
Combining Data Assimilation and Machine Learning to emulate a numerical model from noisy and sparse observations

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

[1]NERSC, Bergen, Norway [2]LOCEAN-IPSL (Sorbonne Université), Paris, France [3]Geophysical Institute, University of Bergen, Norway [4]CEREA, Champs-sur-Marne, France

Introduction



Objective

Producing an accurate and reliable emulator of a numerical model given sparse and noisy observations

Problem

Multidimensional time series $\mathbf{y}_k^{\rm obs}$ $(1 \leq k \leq K)$ observed from an underlying dynamical process:

$$\mathbf{y}_k^{\rm obs} = \mathcal{H}_k(\mathbf{x}_k) + \epsilon_k^{\rm obs}$$

- $\mathcal{H}_k$ is the known observation operator: $\mathbb{R}^m \to \mathbb{R}^p$
- $\epsilon_k^{\rm obs}$ is a noise

Underlying dynamical model:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \boldsymbol{\phi}(\mathbf{x}),$$

where $\boldsymbol{\phi}$ is unknown.
Resolvent:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \int_{t_k}^{t_{k+1}} \boldsymbol{\phi}(\mathbf{x})\,\mathrm{d}t,$$

Two parts:

1. Inferring the ODE using DA: [Bocquet at al., 2019]:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}), \qquad \boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{r}(\mathbf{x}),$$

where $\mathbf{r}(\mathbf{x})$ is specified a priori.

2. Merge DA and ML to emulate the resolvent [Brajard et at al., 2019]

$$\mathbf{x}_{k+1} = \mathcal{G}_{\mathbf{W}}(\mathbf{x}_k) + \epsilon_k^{\rm m},$$

where $\mathcal{G}_{\mathbf{W}}$ is typically a neural network parametrized by $\mathbf{W}$

Connection between data assimilation and machine learning

Data assimilation	machine learning
Dynamical system	Residual deep neural network
Parametrized forecasting model	Layer of a neural network
Optimization	Training
Adjoint modelling	Backpropagation
Locality assumption	Convolutional layers

Numerical illustration: The Lorenz 96 model

- Size of the state $m = 40$
- Integration scheme: 4th order RK (RK4)
- Integration time step: $\delta t_r = \Delta t = 0.05$
- integration length : $K = 50$

Conclusion

- Bayesian data assimilation for state and model estimation:
  - equivalent to a machine learning approach,
  - makes use of locality and homogeneity to reduce the dimension of the model parameters.
- Mixed data assimilation / machine learning approach:
  - emulate the resolvent of the model,
  - training of the neural nets are performed on state estimated from data assimilation.

Part 1: Inferring the ODE using DA

Aim: Estimating $\mathbf{A}$ in the ODE representation of the surrogate dynamics:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}), \qquad \boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{r}(\mathbf{x}),$$

where

- $\mathbf{A}$ is a matrix of coefficients of size $N_x \times N_p$
- $\mathbf{r}(\mathbf{x})$ is a vector of nonlinear regressors of size N_p. For instance, for one-dimensional spatial systems and up to bilinear order:

$$\mathbf{r}(\mathbf{x}) = \left[1, \{x_n\}_{0 \leq n < N_x}, \{x_n x_m\}_{0 \leq n \leq m < N_x}\right].$$

A priori, $N_p = \binom{N_x+1}{2} = \frac{1}{2}(N_x+1)(N_x+2)$ such regressors.

$\longrightarrow$ Intractable in high-dimension!: typically $N_x = \mathcal{O}(10^6)$

Additional assumptions:

- Physical locality of the physics: all multivariate monomials in the ODEs have variables x_n that belong to a stencil, i.e. a local arrangement of grid points around a given node.
  In 1D and with a stencil of size $2L+1$, the size of the dense $\mathbf{A}$ is

$$N_x \times N_{\rm a} \quad \text{where} \quad N_{\rm a} = \sum_{l=L+1}^{2L+2} l = \frac{3}{2}(L+1)(L+2).$$

- Moreover, we can additionally assume translational invariance. In that case $\mathbf{A}$ becomes a vector of size $N_{\rm a}$.

Bayesian analysis of the problem:

Bayesian view on state and model estimation:

$$p(\mathbf{A}, \mathbf{x}_{0:K}|\mathbf{y}_{0:K}) = \frac{p(\mathbf{y}_{0:K}|\mathbf{x}_{0:K},\mathbf{A})p(\mathbf{x}_{0:K}|\mathbf{A})p(\mathbf{A})}{p(\mathbf{y}_{0:K})}.$$

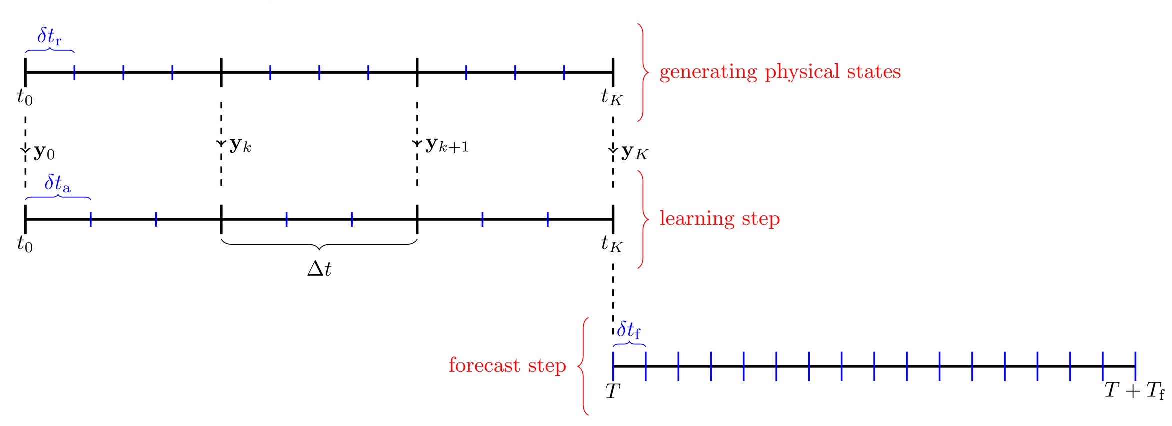
Data assimilation cost function assuming Gaussian error statistics and Markovian dynamics:

$$\mathcal{J}(\mathbf{A}, \mathbf{x}_{0:K}) = \frac{1}{2}\sum_{k=0}^{K}\|\mathbf{y}_k - \mathbf{H}_k(\mathbf{x}_k)\|^2_{\mathbf{R}_k^{-1}} + \frac{1}{2}\sum_{k=1}^{K}\|\mathbf{x}_k - \mathbf{F}_{\mathbf{A}}(\mathbf{x}_{k-1})\|^2_{\mathbf{Q}_k^{-1}} - \ln p(\mathbf{x}_0, \mathbf{A}),$$

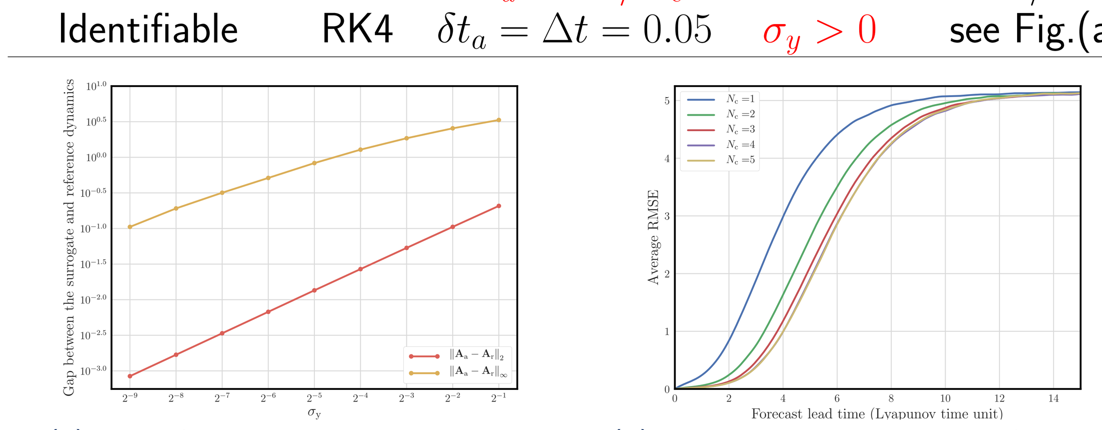
where $\mathbf{F}_{\mathbf{A}}$ is the resolvant of the model between t_k and $t_k + \Delta_t$.
Typical machine learning cost function with $\mathbf{H}_k = \mathbf{I}_k$ in the limit $\mathbf{R}_k \longrightarrow \mathbf{0}$:

$$\mathcal{J}(\mathbf{A}) \approx \frac{1}{2}\sum_{k=1}^{K}\|\mathbf{y}_k - \mathbf{F}_{\mathbf{A}}(\mathbf{y}_{k-1})\|^2_{\mathbf{Q}_k^{-1}} - \ln p(\mathbf{y}_0, \mathbf{A}).$$

Numerical experiment:



Model	scheme	time step	Obs. noise	$\|\mathbf{A}_a - \mathbf{A}_r\|_\infty$
Identifiable	RK4	$\delta t_a = \Delta t = 0.05$	0	$\sim 10^{-13}$
Non identifiable	RK2	$\delta t_a = \Delta t / N_c$	0	N/A
Identifiable	RK4	$\delta t_a = \Delta t = 0.05$	$\sigma_y > 0$	see Fig.(a)



(a) Identification with noisy observations

(b) Forecast with non identifiable model

References

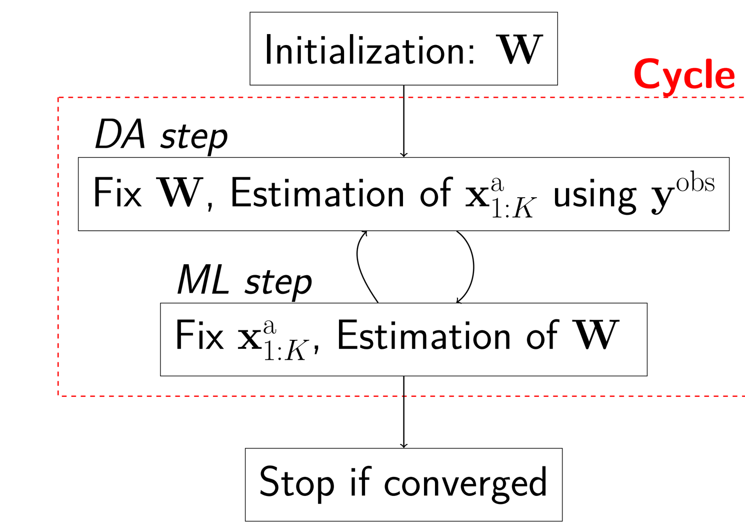
[1] Marc Bocquet, Julien Brajard, Alberto Carrassi, and Laurent Bertino.
Data assimilation as a deep learning tool to infer ODE representations of dynamical models.
Nonlinear Processes in Geophysics Discussions, pages 1–29, 2019.
URL: https://doi.org/10.5194/npg-2019-7, doi:10.5194/npg-2019-7.

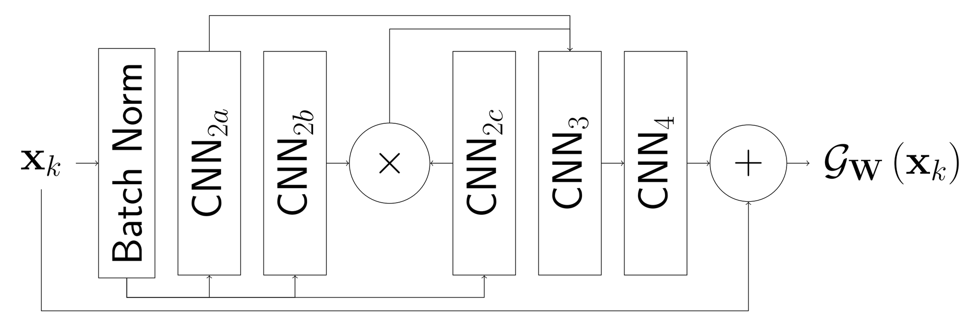
[2] J. Brajard, A. Carrassi, M. Bocquet, and L. Bertino.
Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the lorenz 96 model.
Geoscientific Model Development Discussions, 2019:1–21, 2019.
URL: https://www.geosci-model-dev-discuss.net/gmd-2019-136/, doi:10.5194/gmd-2019-136.

Part 2: Mixing data assimilation and machine learning

Aim: Estimating the weights $\mathbf{W}$ of a neural network representing the resolvent of the model:

- $\mathbf{x}_{k+1} = \mathcal{G}_{\mathbf{W}}(\mathbf{x}_k) + \epsilon_k^{\rm m} = \mathbf{x}_k + \int_{t_k}^{t_{k+1}} \phi(\mathbf{x})\,\mathrm{d}t$

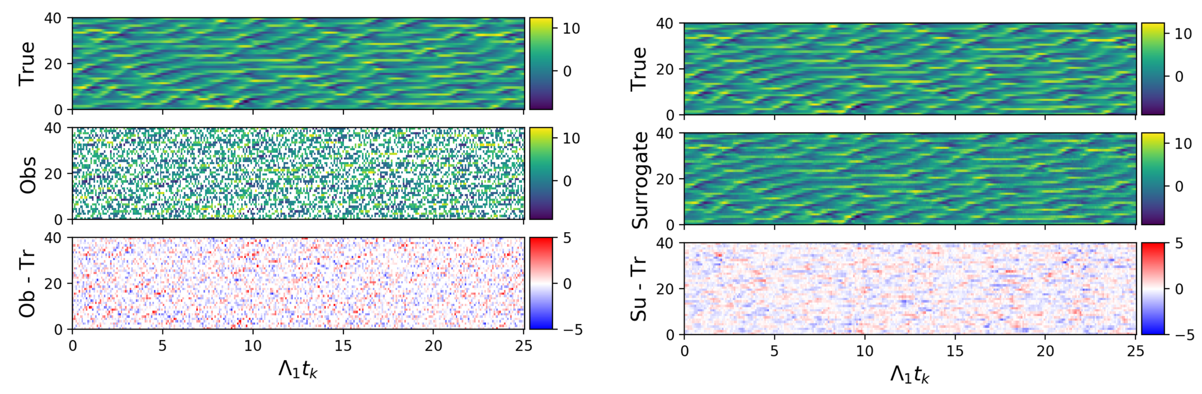




Residual bi-linear convolutive neural network (9391 weights), compared with $N_{\rm a} = 18$ in case of ODE inference.

Layer	number of unit	filter size	number of weights
1 (batchnorm)			2
2 (bilinear)	24×3	5	144×3
3 (convolutive)	37	5	8917
4 (linear)	1	1	38

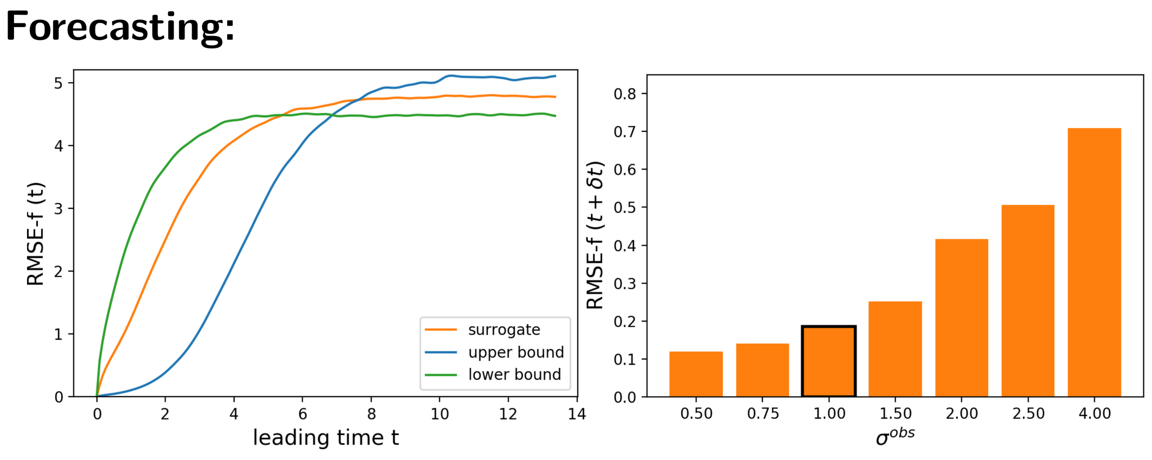
Interpolation:



RMSE (obs)= 1 RMSE-a= 0.8

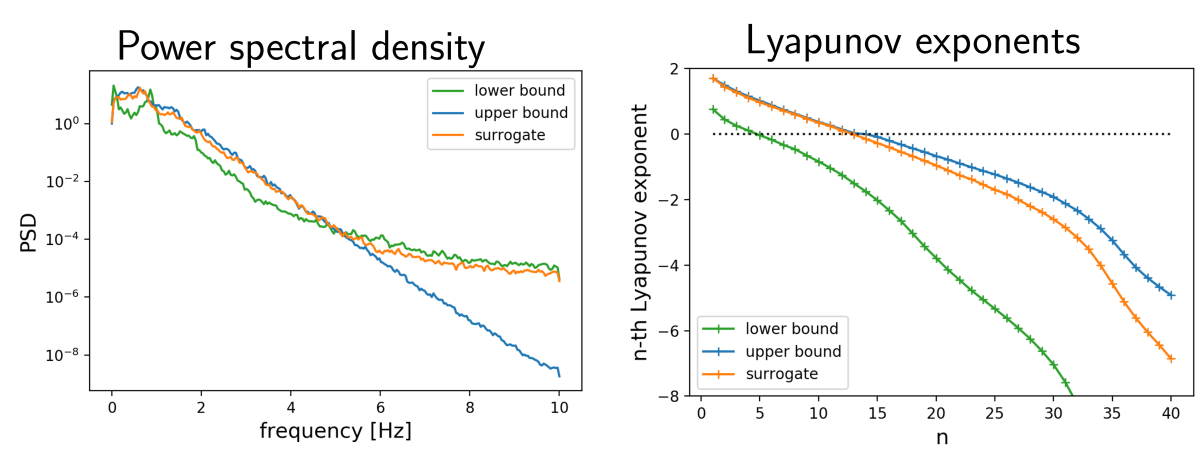
	Method	RMSE-a
Lower bound	Quadratic interpolation	2.32
	DA with surrogate model	0.80
Upper bound	DA with true model	0.34

Forecasting:



- Lower bound: Neural Net trained with observation interpolated using quadratic interpolation (no data assimilation).
- Upper bound: Neural Net trained with "perfect" observations (complete, no noise).

Long term dynamics reconstruction:



- Lower bound: Neural Net trained with observation interpolated using quadratic interpolation (no data assimilation).
- Upper bound: True model

*Contact email : julien.brajard@locean-ipsl.upmc.fr

The Research Council of Norway
project REDDA (#250711)