Report of Digital Shadow of the Mediterrean Copernicus Analysis and Forecast System

Guglielmo Padula

January 15, 2025

Abstract

This is an internal report which describes the Digital Shadow of the Mediterrean Copernicus Analysis and Forecast System, which is part of the projects developed under the Research Topic 4 of the Spoke 9 of the INEST project, as a joint work between SISSA and OGS.

1 Brief Problem Description

The problem consists in creating a Digital Shadow of the Mediterrean Copernicus Analysis and Forecast System (physics part, biogeochemistry part). A digital shadow is a surrogate model that:

- is as fast as possible
- automatically learns new data from the system is training to imitate, in the fastest way possible. So, it needs a data assimilation mechanism.
- is also able to provide uncertainty on its outputs

A digital shadows can be useful to:

- compute short term forecasts in a fast way in case of emergencies
- compute long term forecasts (Mediterrean Copernicus Analysis and Forecast System predicts only 10 days in advance).

To do this, we create a novel methodology by combining three different techiniques: a quadratic manifold dimensionality reduction technique, Gaussian Process Regression, Stochastic Differential Equations.

2 Theoretical Description

We assume that each variable of the system follows

$$X(t) = Z(t)W_1 + (Z(t) \odot Z(t))W_2 + \mu$$

where $X(t)$ has dimension D. D is the size of the basin in which we are simulating the digital shadow. $Z(t)$ are p independent curves in $\mathbb{R}$, where $p < D$, and $Z(t)$ has zero mean

$$\frac{1}{T}\int_0^T Z(t)dt = 0$$

and unit variance

$$\frac{1}{T}\int_0^T Z(t)^2 dt = 1$$

and furthermore has ergodic properties.

2.1 First step

Let $X_{train} \in \mathbb{R}^{n \times D}$ realization of variable of interested $X(t)$ evaluated in D different spatial points and at N equispaced timesteps. Due to the fact that $Z(t)$ has zero mean we can compute an approximation of μ by applying monte carlo integration

$$\mu = \frac{1}{N} \sum_{i=0}^{n-1} X_{train}[i]$$

and we subtract it from our data, to obtain

$$\tilde{X}_{train} = X_{train} - \mu.$$

Lets impose $W_2 = 0$ for a moment, then we have that the temporal covariance matrix of X_{train} can be computed as $W_1 W_1^T$ and that can be approximated as $\frac{\tilde{X}_{train}^T \tilde{X}_{train}}{n}$, which however is computationally intractable for large D. A possible solution compute the Singular Value Decomposition of $\tilde{X}_{train}$

$$\tilde{X}_{train} = U \Sigma V^T$$

where $\Sigma \in \mathbb{R}^{N \times N}$ is a diagonal matrix with nonnegative diagonal elements $\sigma_0, \ldots \sigma_N$ in decreasing order, $U \in \mathbb{R}^{N \times N}$ such that $U^T U = I$, $V \in \mathbb{R}^{N \times D}$ such that $V^T V = I_N$. Then we find the index

$$p = \operatorname*{arg\,min}_{i=0 \ldots N-1} ||\sigma_i - \frac{1}{10}\sigma_0||$$

and we extract the matrix

$$U_p^T = [U_0^T, U_1^T \ldots, U_p^T]$$

and the matrix

$$V_p^T = [V_0^T, V_1^T \ldots, V_p^T]$$

and Σ_p which is the matrix with diagonal elements $\sigma_0, \ldots \sigma_p$ in decreasing order.
So $\Sigma_p \in \mathbb{R}^{p \times p}$, $U_p \in \mathbb{R}^{N \times p}$, $V_p \in \mathbb{R}^{D \times p}$. We now define

$$W_1 = \frac{1}{\sqrt{n-1}} \Sigma_p V_p^T \in \mathbb{R} \times \text{ı}$$

,

$$W_1^{inv} = \sqrt{n-1} V_p \Sigma_p^{-1}$$

,

$$Z = \sqrt{n-1} U_p.$$

At this point (as we have lost some information due to the truncation) we compute

$$W_2 = ((Z_{train} \odot Z_{train})^T (Z_{train} \odot Z_{train}) + \gamma I)^{-1} (Z_{train} \odot Z_{train})^T \tilde{X}_{train} (I - W_p^{inv} W_p)$$

With these definitions, it holds approximately that

$$X_{train} = \mu + W_1 Z + W_2 (Z_{train} \odot Z_{train})$$

which is quadratic in Z_{train}.

2.2 Second step

We have yet to model $Z(t)$. To do it, we model is as $\dot{Z}(t) = f(Z(t))$ where f is an unknown function. Do find f first we approximate $\dot{Z}_{train}$ by fixing a time step δt which is an integer multiple of one, and we use the second order centred difference scheme when possible, and the first order backward or forward scheme when we can't. We approximate f using a Gaussian Process. we define the training matrix as

$$K_{train,train}[i, j] = K(Z_{train} Z_{train})$$

then we have that

$$\dot{Z}(t) \sim Normal(K(Z(t), Z_{train})K_{train}^{-1}Z_{train} -$$

$$K(Z(t), Z_{train})K_{train}^{-1}K(Z(t), Z_{train})^T,$$

where $\hat{Y}$ is the one-step hindcast with respect to $\hat{X}$.
This expression if formal, but it is similar to the SDE

$$dZ(t) = K(Z(t), Z_{train})K_{train}^{-1}Z_{train}, K_{test,test}dt -$$

$$K(Z(t), Z_{train})K_{train}^{-1}K(Z(t), Z_{train})^T dW$$

which in fact, thanks to the universal approximation properties of gaussian properties, converges to the true solution in the infinite data limit. We do a linearization of this SDE and a zero order approximation to obtain the following equations for the mean $a(t)$ and the variance $b(t)$ of $z(t)$:

$$\dot{a}(t) = K(a(t), Z_{train})K_{train})^{-1}\dot{Z}_{train}$$

$$\dot{b}(t) = K(a(t), Z_{train})K_{train}^{-1}(K_{test,train})^T$$

We solve this ode, using the time step δt and by adopting a forward Euler scheme for the first time step and a two step Adam-bashford for the later time steps.

2.3 Forth step

I will now put all the things together. For what regards forecasting, from the previous points we have that:

$$X(t) = Z(t)W_1 + (Z(t) \odot Z(t))W_2 + \mu$$

with

$$Z(t) \sim N(a(t), b(t))$$

By using the properties of normal random variables, we can approximate everything with a normal random variable with

$$E[X(t)] = a(t)^T W_1 + (a(t) \odot a(t) + b(t))^T W_2 + \mu$$

$$Var[X(t)] = (b(t))^T (W_1 \odot W_1) + (4b(t) \odot a(t) \odot a(t) + a(t) \odot a(t) \odot a(t) \odot a(t))^T (W_2 \odot W_2)$$

At this point we truncate the Normal Distribution if required.

2.4 Validation

For the validation of the mean, I adopt the approach described in the previous report. Validation of the uncertainty is more tricky, because we cannot use directly the state variables, as they are highly correlated. Let me define

$$Z_N = W_1^{inv}(X_{test} - \mu)$$

where N is a time in which we can compute a forecast. Let $V(N)$ such that

$$V(N)[i] = \frac{Z(N)[i] - a(N)[i]}{\sqrt{b[i]}} \quad \forall i = 0...p-1$$

It can be proven that

$$V(N) \sim Normal(0, 1).$$

This can be tested in the following way. I specify a p-value: 0.95. The associated confidence interval is $[-1.96, 1.96]$ Then the quantity

$$H(N) = \frac{1}{p} \sum_{i=0}^{p-1} 1_{(-1.96<Z(N)<1.96)}$$

is distributed as

$$H(N) \sim Binomial(0.95, p)$$

So if the model is correct, the quantity

$$H_N = \frac{1}{p} \sum_{i=0}^{p-1} 1_{(-1.96<Z_N<1.96)}$$

should be contained in α specified confidence interval. I chose $\alpha = 0.99$.

3 Practical consideration and an example

To parallelize the computations, we split the Mediterrean sea in different basins.

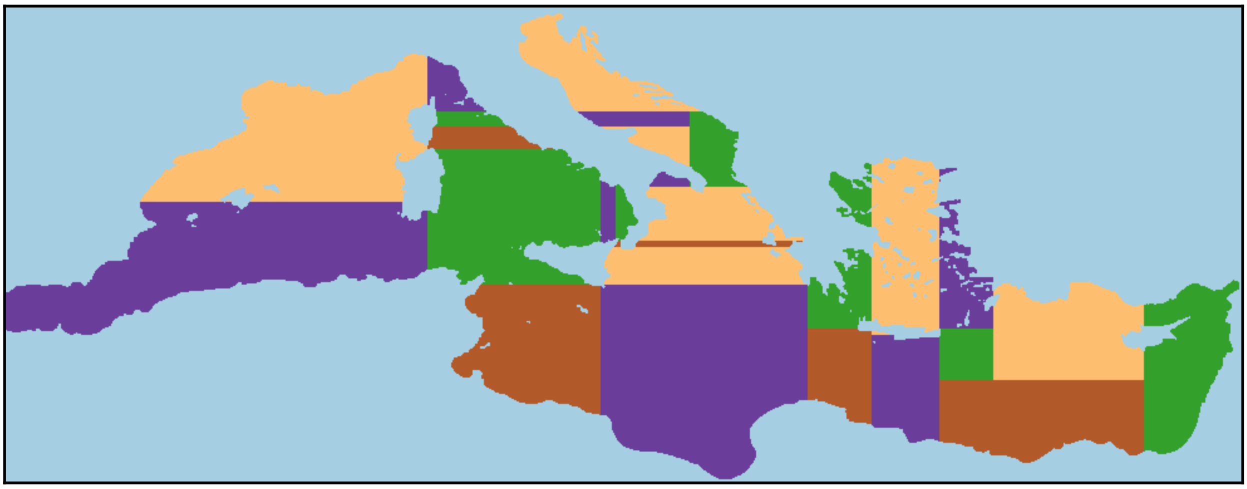
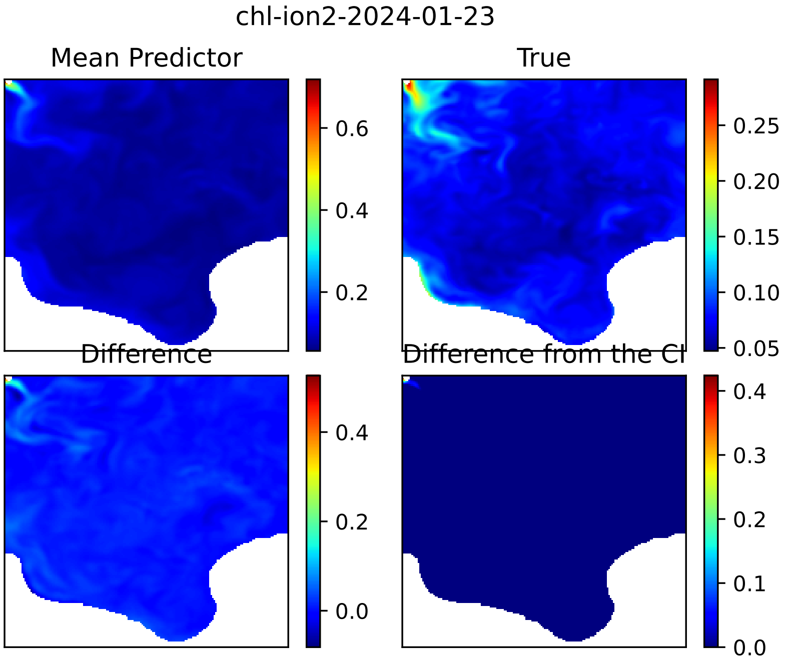
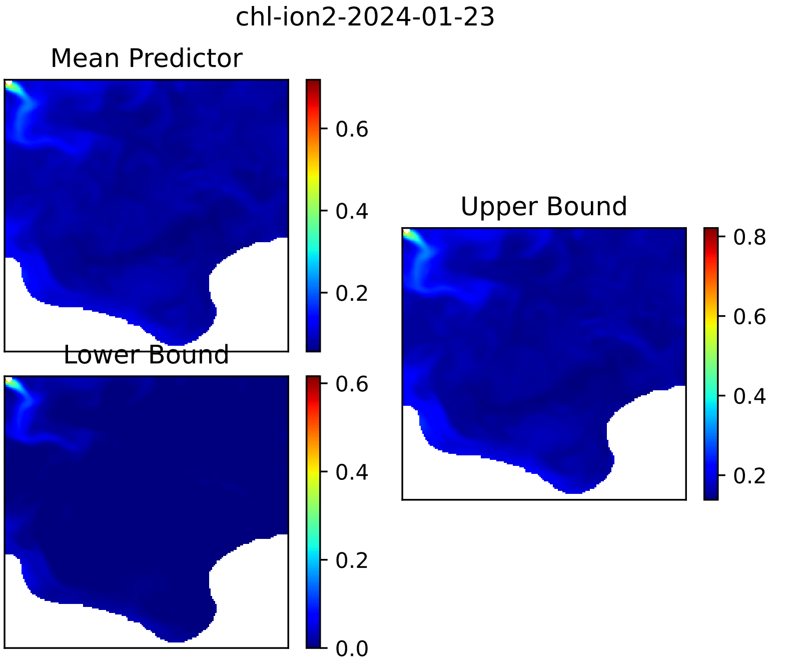


Figure 1: All the rectangular cuboids for which a different model is learned.

These are the outputs for ION2, chrolophyll, 120 data used for training, for forecasting 10 days starting from the 2024-01-13. (Plots are shown for the last day only).

Bias (Winter, Coast)	RMSD (Winter, Coast)	Bias-UQ (Winter, Coast)	RMSD-UQ (Winter, Coast)	Bias (Winter, Opensea)	RMSD (Winter, Opensea)	Bias-UQ (Winter, Opensea)	RMSD-UQ (Winter, Opensea)
-2.58E-03	8.45E-04	5.77E-04	1.71E-04	-6.39E-03	2.30E-04	-8.53E-06	1.59E-07
-3.01E-03	8.03E-04	4.58E-04	1.25E-04	-6.53E-03	2.33E-04	-9.38E-06	1.80E-07
-4.43E-03	6.93E-04	2.03E-04	5.09E-05	-5.37E-03	3.15E-04	3.10E-05	2.26E-06
-3.91E-03	6.43E-04	1.86E-04	6.56E-06	5.08E-03	1.41E-03	4.23E-04	3.15E-05
7.23E-04	1.07E-03	3.41E-05	8.57E-06	-4.55E-03	2.00E-03	-4.98E-04	4.51E-05


chl-ion2-2024-01-23
Mean Predictor
True
Difference
Difference from the CI
0.2
0.4
0.6
0.05
0.10
0.15
0.20
0.25
0.0
0.2
0.4
0.0
0.1
0.2
0.3
0.4


chl-ion2-2024-01-23
Mean Predictor
Upper Bound
Lower Bound
0.2
0.4
0.6
0.2
0.4
0.6
0.8
0.0
0.2
0.4
0.6