Fisheries, Stochastic Control versus AI Solutions

Olivier Pironneau
olivier.pironneau@sorbonne-universite.fr ,
LJLL, Sorbonne Université, Paris, France.

SCML 2022 at KAUST

Abstract

Modeling and computing fish biomass is as old as computers. Using stochastic models is not so widespread, yet it is needed to comprehend fishing quotas. Observing for a few days the boats at sea and their yield allows to identify the parameters of the model; AI can even be used to port the model on a smart phone. Fishing quotas are more difficult to optimize. First they are unpleasant so they should not vary too much versus time.
The model we use has a stochastic differential equation for the biomass on which a stochastic dynamic programming or a Hamilton-Jacobi-Bellman algorithm; the stochastic control is the fishing quota. We compare the solutions obtained by dynamic programming against those obtained with a neural network which preserves the Markov property of the solution. The method is extended to a multi species model and shows that the Neural Network is usable even in high dimensions, i.e. many fish types and age categories.

In mathematical terms the problem is

$$\min_{\mathbf{u}\in\mathcal{U}} \bar{J} := \int_0^T \mathbb{E}|\mathbf{X}(t)-\mathbf{X}^d(t)|^2 \mathrm{d}t \ : \ \mathrm{d}\mathbf{X}_t = \mathbf{X}\cdot\underline{\mathbf{\Lambda}}\left[\mathbf{r}-\mathbf{u}-\underline{\boldsymbol{\kappa}}\mathbf{X}+\underline{\boldsymbol{\sigma}}\mathrm{d}\mathbf{W}_t\right], \quad \mathbf{X}(0)=\mathbf{X}^0.$$

where $\mathbf{X}^d(t)$ is the desired state, $\mathbf{r}, \underline{\boldsymbol{\kappa}}$ and $\underline{\boldsymbol{\sigma}}$ are vector and matrices parameters and $\mathbf{W}_t$ is a vector-valued Weiner process; the quotas $\mathbf{u}(t)$ drive the state $\mathbf{X}(t)$.

Without quota ($\mathbf{u}=0$) the parameters are easy to obtained from a few snapshots of $\mathbf{X}(t)$ because there are few of them.

Then we will compare traditional methods like Hamilton-Jacobi-Bellman equations, Stochastic Control with brute force solutions obtained by a neural network trained to find the control surface $\mathbf{u}(\mathbf{X},t)$.

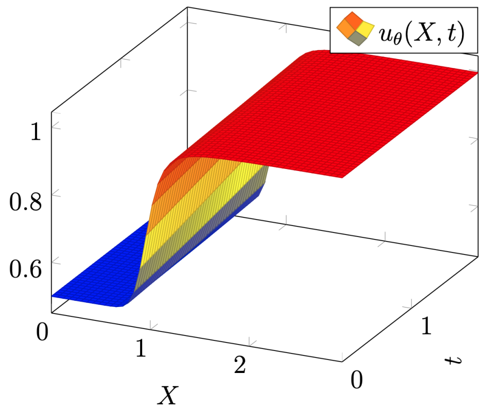

Figure 23: Dynamic feedback control,$u_\theta(X,t)$ computed by the Markovian Neural Network.

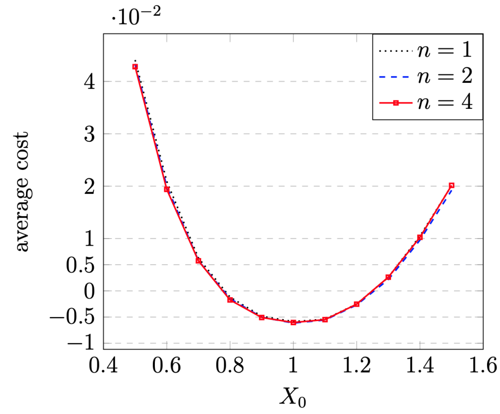

Figure 24: Single species: Values of the cost versus X_0 for 100 realizations with the Markovian neural network, for the 3 time meshes $50 \times n$.

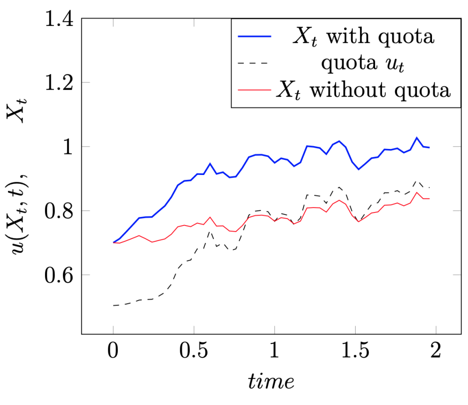

Figure 25: Simulation for a single fish species computed by the Markovian Neural Network and $X_0 = 0.7$. Performance with and without quota u_t.

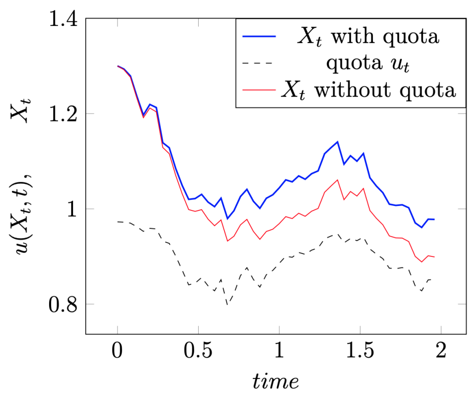

Figure 26: Simulation for a single fish species computed by the Markovian Neural Network and $X_0 = 1.3$. Performance with and without quota u_t.