Machine Learning of Dynamic Processes with Applications to Time Series Forecasting

Lyudmila Grigoryeva

University of St. Gallen, Switzerland

Emergent Algorithmic Intelligence Winter School 2023
JGU Research Center for Algorithmic Emergent Intelligence
Mainz (Nierstein), 2023

Reservoir computing as a machine learning paradigm.
Time series forecasting with reservoir computing

Outline for section 1

Setup(s)

- Machine learning as an input/output problem:
  - **Input z** contains available information for the solution of the problem (historical data, explanatory factors, features of the individuals that need to be classified, observations of a dynamical system).
  - **Output y** contains the target solution of the problem (ground truth forecast data, explained variables, classification results).
- Distinctive feature: (mostly) **agnostic** setup.
- Simultaneous interest for stochastic (processes) and deterministic (non-autonomous dynamical) systems.
- We distinguish between static, dynamic, discrete-time, and continuous-time setups and between deterministic and stochastic situations.

Setup(s)

	Static		Dynamic (discrete time)	
	Deterministic	**Stochastic**	**Deterministic**	**Stochastic**
Ingredients	$\mathbf{z} \in \mathbb{R}^d$ $\mathbf{y} \in \mathbb{R}^m$	$\mathbf{z} \in L^p(\Omega, \mathbb{R}^d)$ $\mathbf{y} \in L^p(\Omega, \mathbb{R}^m)$	$\mathbf{z} \in (\mathbb{R}^d)^{\mathbb{Z}_-}$ $\mathbf{y} \in (\mathbb{R}^m)^{\mathbb{Z}_-}$	$\mathbf{z} \in L^p\left(\Omega, (\mathbb{R}^d)^{\mathbb{Z}_-}\right)$ $\mathbf{y} \in L^p\left(\Omega, (\mathbb{R}^m)^{\mathbb{Z}_-}\right)$
Problem to be solved	$\mathbf{y} = f(\mathbf{z})$ f measurable	$\mathsf{E}[\mathbf{y} \mid \mathbf{z}]$	$\mathbf{y}(\cdot) = F(\mathbf{z}(\cdot))$	$\mathsf{E}[\mathbf{y}(\cdot) \mid \mathbf{z}(\cdot)]$
Examples and Applications	• observables or diagnostics variables in complex physical or noiseless engineering systems • calibration of financial models • translators • transcription	• image classification • speech recognition • factor analysis • anomaly detection	• integration or path continuation of (chaotic) differential equations • molecular dynamics • structural mechanics • vibration analysis • space mission design • autopilot systems • robotics • memory tasks • games	• physiological time series classification • financial bubble detection • time series forecasting • volatility filtering • system identification (blackboxing) • filters (transducers) and equalizers • imputation of missing values • source separators

Mathematical formulation of reservoir computing

A **reservoir computer (RC)** is a state-space system:

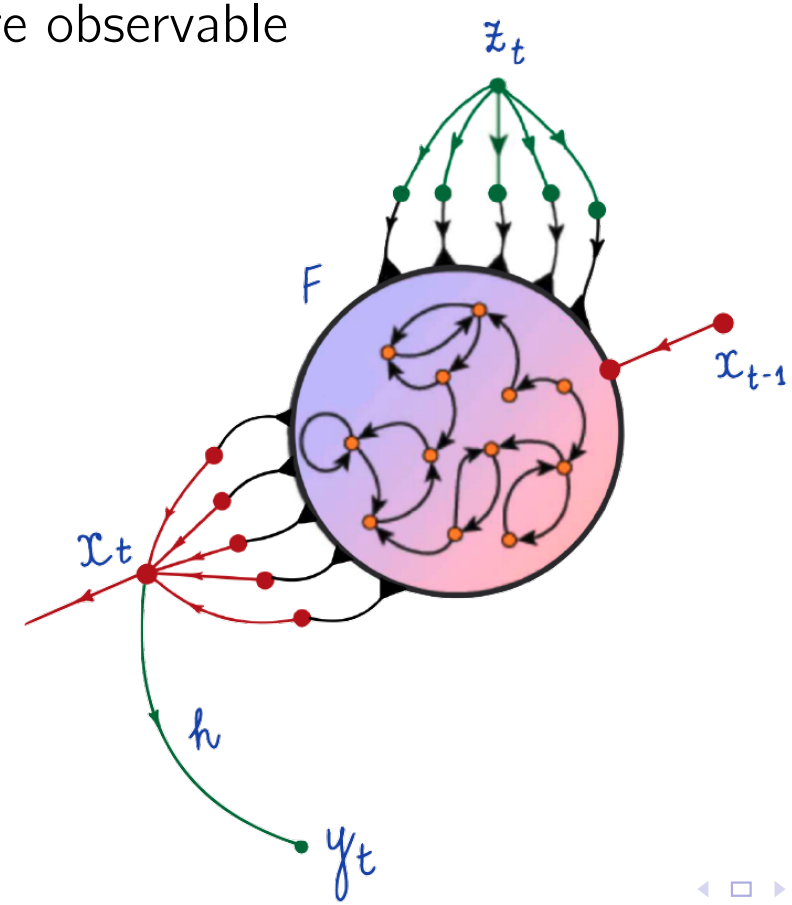
$$\begin{cases} \mathbf{x}_t = F(\mathbf{x}_{t-1}, \mathbf{z}_t), & (1) \\ \mathbf{y}_t = h(\mathbf{x}_t), & (2) \end{cases}$$

determined by a **reservoir** map $F : \mathbb{R}^N \times \mathbb{R}^d \longrightarrow \mathbb{R}^N$ and a **readout** map $h : \mathbb{R}^N \to \mathbb{R}^m$ that transform (or filter) an infinite discrete-time **input** $\mathbf{z} = (\ldots, \mathbf{z}_{-1}, \mathbf{z}_0, \mathbf{z}_1, \ldots) \in (\mathbb{R}^d)^{\mathbb{Z}}$ into an **output** signal $\mathbf{y} = (\ldots, \mathbf{y}_{-1}, \mathbf{y}_0, \mathbf{y}_1, \ldots) \in (\mathbb{R}^m)^{\mathbb{Z}}$. $\mathbf{x}_t \in \mathbb{R}^N$, $t \in \mathbb{Z}$ are the **reservoir states**.

Defining features of RC architecture

Architecture of reservoir system (F, h)

- (Components of) reservoir map $F : \mathbb{R}^N \times \mathbb{R}^d \longrightarrow \mathbb{R}^N$ randomly generated, often sparse, and fixed for the whole modeling process
- Static readout $h : \mathbb{R}^N \to \mathbb{R}^m$ is subject to choice depending on the task
- States $\mathbf{x}_t \in \mathbb{R}^N$ are observable



Perspectives of Reservoir Computing

- (Nonlinear) state-space systems ← as models to represent/approximate
  - input-output systems
    - realization problem in systems and control theory
    - internal representation through latent states in filtering theory
  - dynamical systems
    - synchronization theory of dynamical systems
  - stochastic processes
- Computational tools ← as recursive algorithm
  - computationally convenient for treating dynamic processes and sequential data
  - multiple inputs/outputs of different nature can be treated in parallel
  - well-adapted to realtime/online computing
  - admits energy efficient and high-speed hardware implementations with a variety of physical systems, substrates, and devices
- Learning paradigm ← the center of discussion

Outline for section 2

Dynamic Versus Static Learning Tasks

Learning in an agnostic environment: **reconstruct/learn an unknown target out of partial information**.

- In the case of **I/O systems** one learns
  - I/O system out of finite observations of input and output processes (learn filter)
  - stochastic process out of a finite number of observations of its realization(s) (learn Bernoulli shift)
  - conditional moments of a stochastic process out of a finite number of observations of its realization (learn optimal predictor)
- In the context of **dynamical systems** the goal is to learn
  - dynamical system out of its low-dimensional observations
  - attractor of chaotic dynamical system out of finite observations

Applications to dynamical systems learning (attractors and path continuation)

Outline for section 3

Forecasting of a Mackey-Glass chaotic time series

- We take one solution of the TDDE:

$$\frac{dz}{dt} = \frac{0.2z(t-\tau)}{1+z(t-\tau)^{10}-0.1z(t)} \quad \text{delay} \quad \tau = 17.$$

- We forecast a chaotic path by learning not the forecasting functional but by learning the dynamical system.

Forecasting of a Mackey-Glass chaotic time series

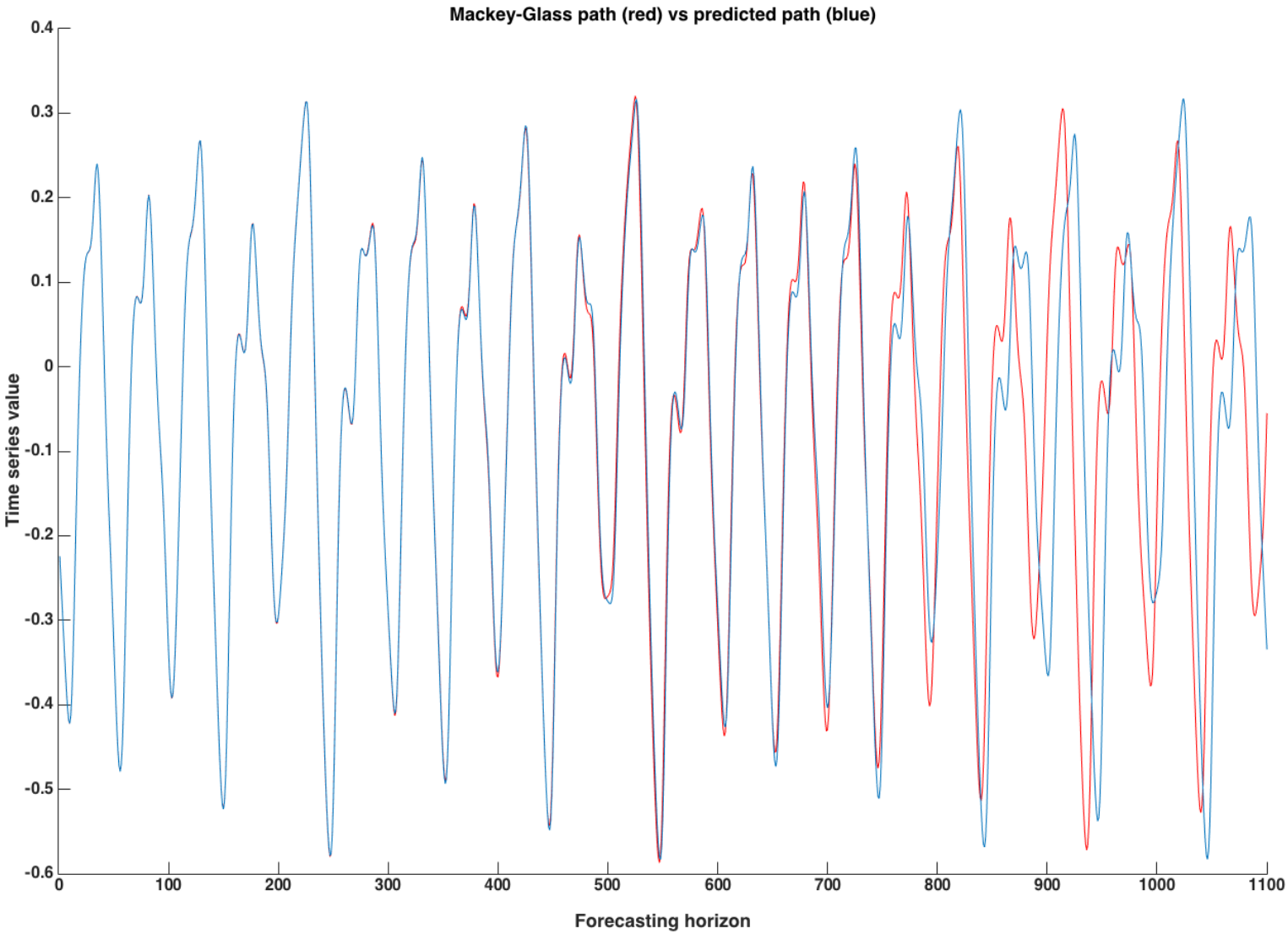
A simple reservoir computer called **Echo State Network** (ESN):

$$\begin{cases} \mathbf{x}_t = \boldsymbol{\sigma}\left(A\mathbf{x}_{t-1} + \mathbf{C}z_t + \boldsymbol{\zeta}\right), \\ y_t = \mathbf{W}^\top \mathbf{x}_t. \end{cases}$$

Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication

Herbert Jaeger* and Harald Haas

We present a method for learning nonlinear systems, echo state networks (ESNs). ESNs employ artificial recurrent neural networks in a way that has recently been proposed independently as a learning mechanism in biological brains. The learning method is computationally efficient and easy to use. On a benchmark task of predicting a chaotic time series, accuracy is improved by a factor of 2400 over previous techniques. The potential for engineering applications is illustrated by equalizing a communication channel, where the signal error rate is improved by two orders of magnitude.


Mackey-Glass path (red) vs predicted path (blue)
Time series value
Forecasting horizon
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
0
100
200
300
400
500
600
700
800
900
1000
1100

Learning a chaotic PDE

The Kuramoto-Sivashinsky model for flame propagation

$$y_t = -yy_x - y_{xx} - y_{xxxx} + \mu \cos\left(\frac{2\pi x}{\lambda}\right)$$

- Prediction using an ESN-based learning of this system by Edward Ott's group in [PLH+17, PHG+18]
- It works well up to eight Lyapunov times

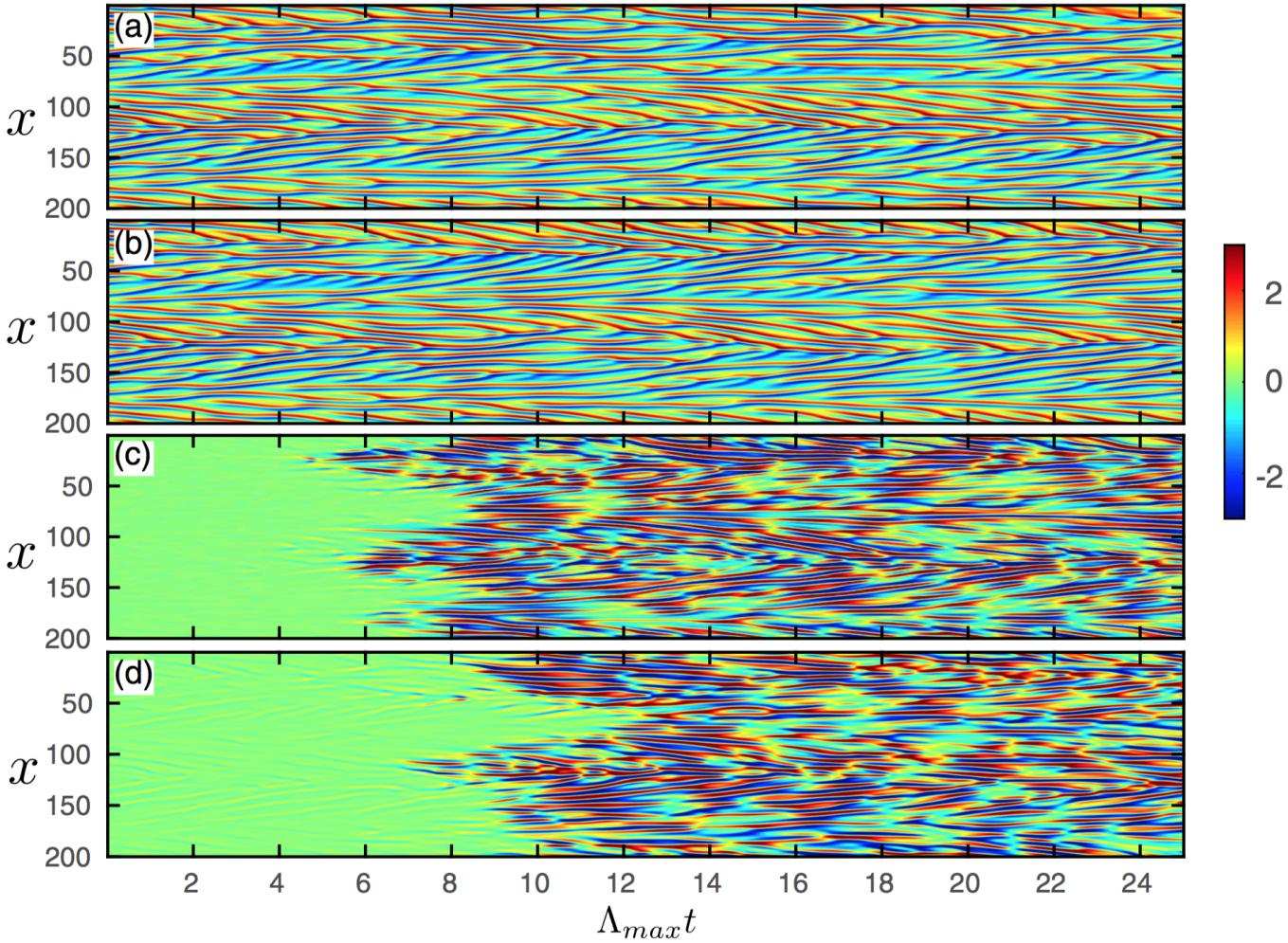


Figure: (a) is the actual solution, (b) is the solution produced by the ESN proxy (c) and (d) are the errors obtained by substraction using two different initializations

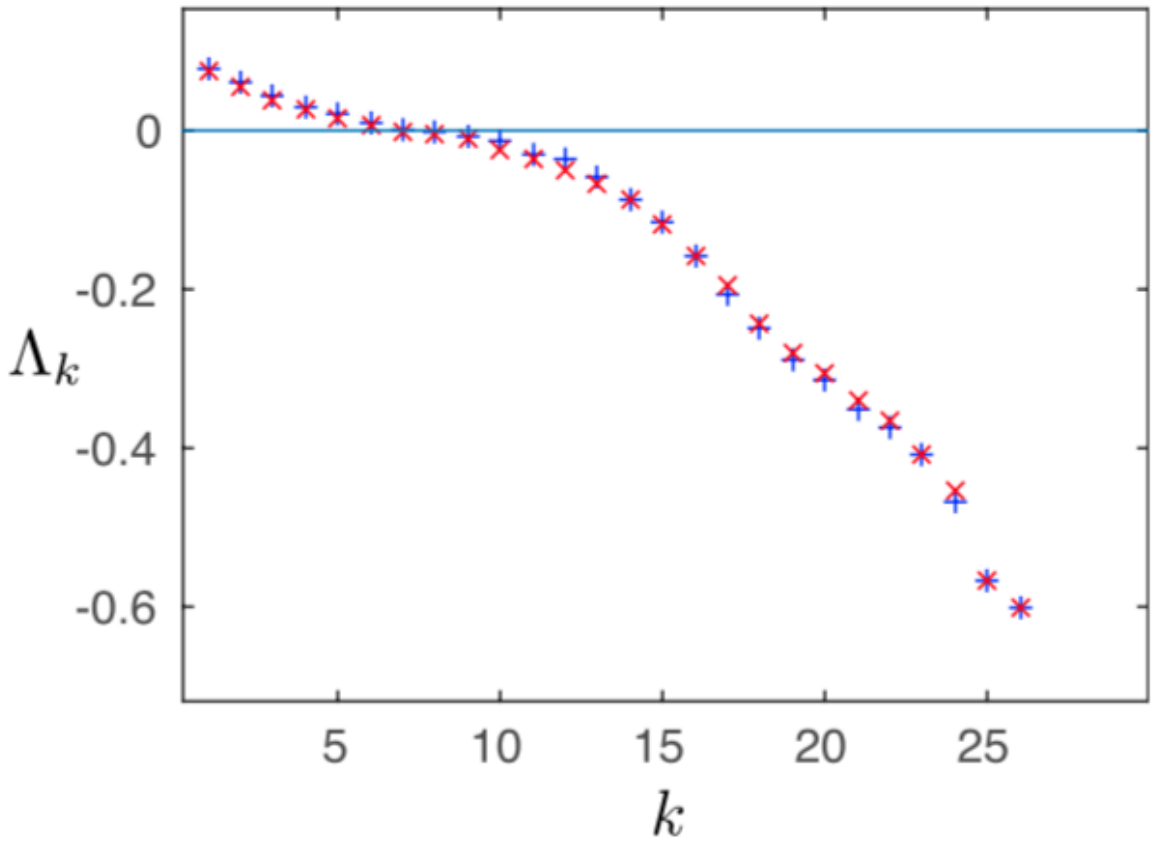


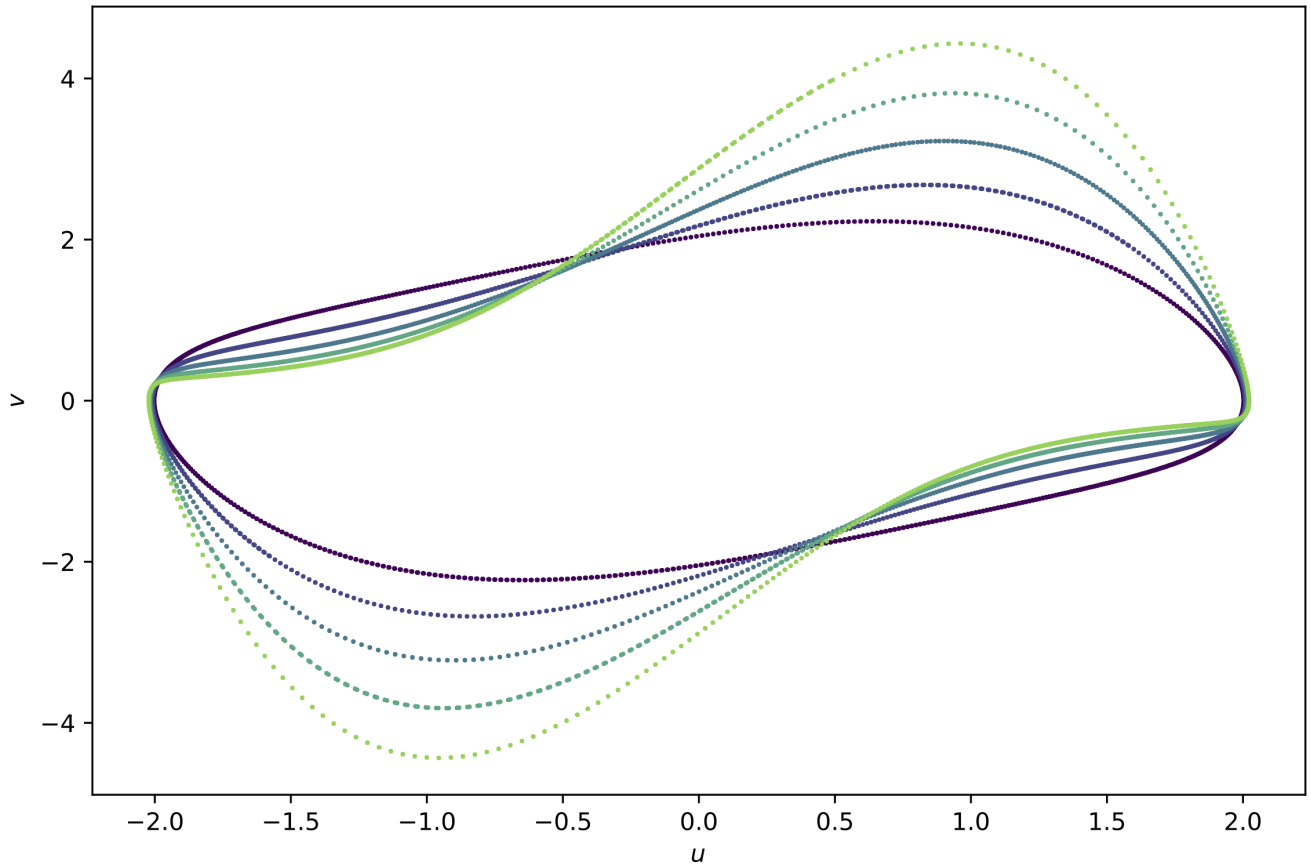
FIG. 8. Estimating the Lyapunov exponents of the inhomogeneous ($\mu > 0$) KS equation. First 26 Lyapunov exponents of the trained reservoir dynamical system running in the autonomous prediction mode (blue "+" markers) and the modified (i.e., $\mu > 0$) KS system (red "×" markers). The parameters of Eq. (7) are $L = 60$, $\mu = 0.1$, and $\lambda = 15$.

Taken from [PLH+17]

Van der Pol Attractor

$$\dot{u} = v$$

$$\dot{v} = \mu(1 - u^2)v - u$$



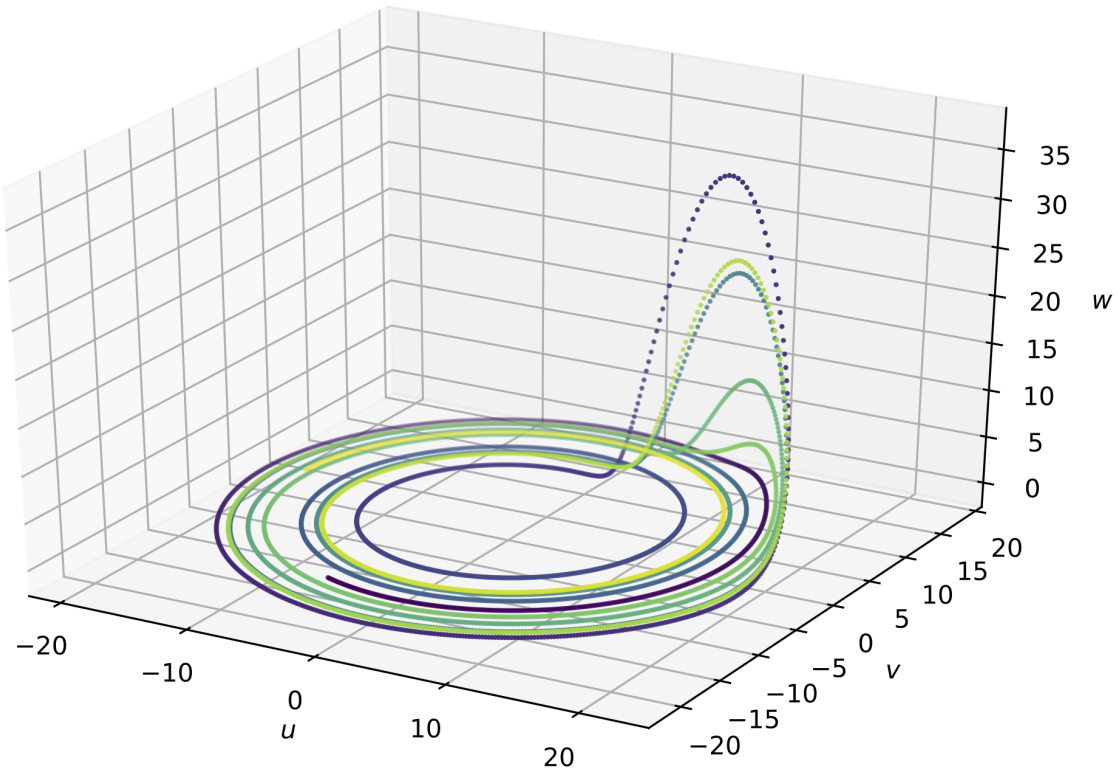
$$\mu = 0.5, 1, 1.5, 2, 2.5$$

Rössler Attractor

$$\dot{u} = -v - w$$

$$\dot{v} = u + av$$

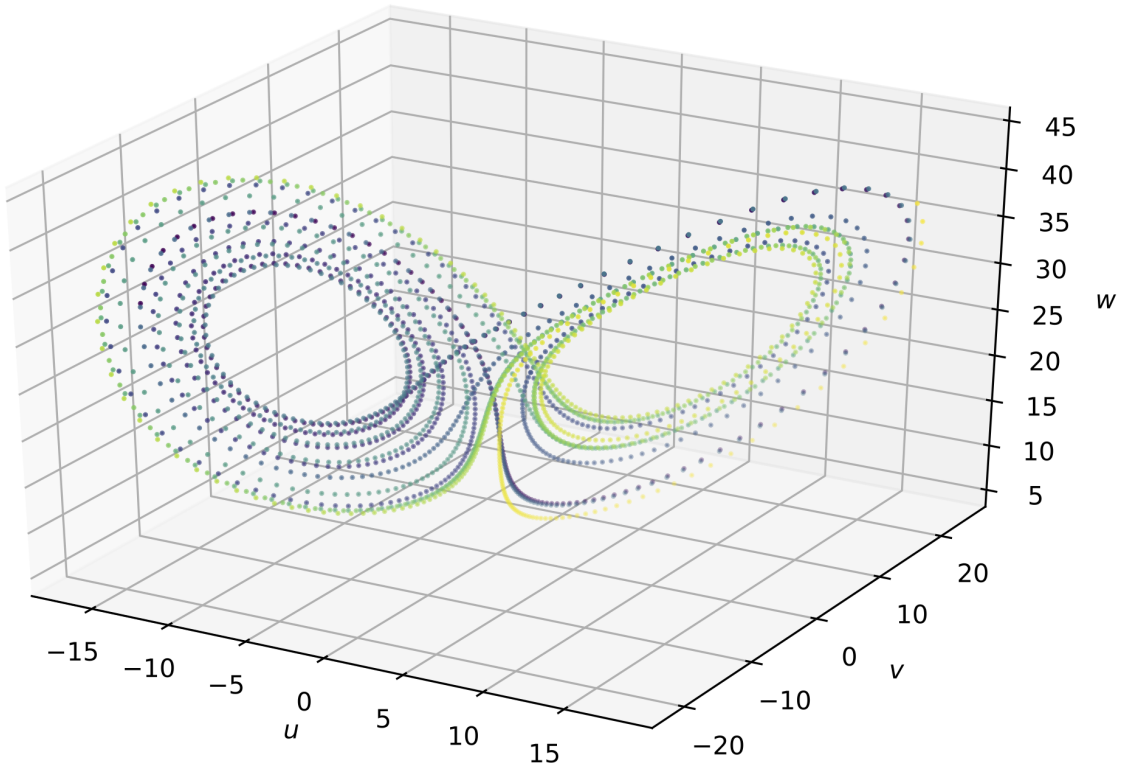
$$\dot{w} = b + w(u - c)$$



$$a = \frac{1}{10}, b = \frac{1}{10}, c = 14$$

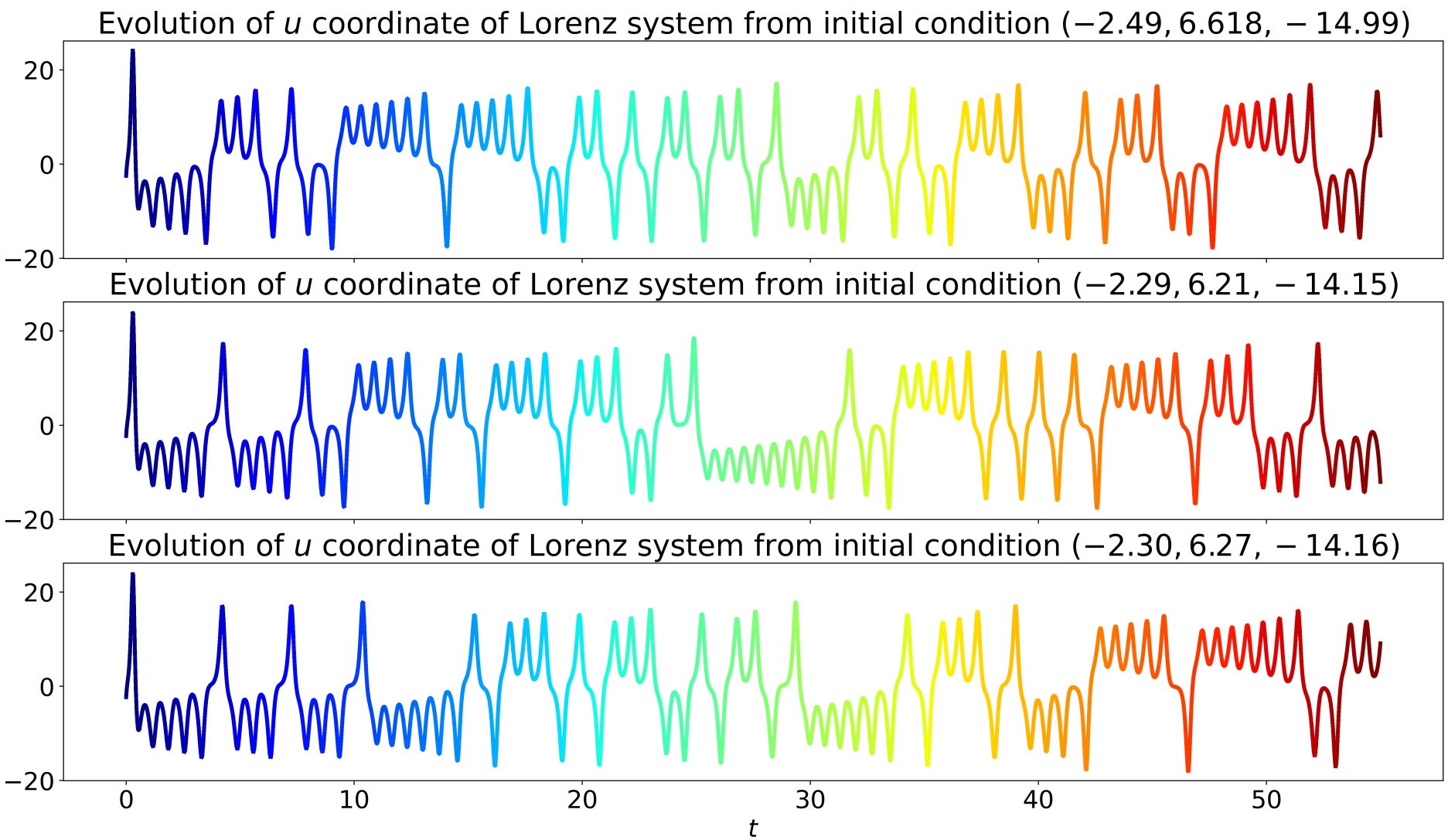
Lorenz Attractor

$$\dot{u} = \sigma(v - u)$$
$$\dot{v} = u(\rho - w) - v$$
$$\dot{w} = uv - \beta w$$



$$\sigma = 10, \rho = 28, \beta = \frac{8}{3}$$

Lorenz Attractor



Are (chaotic) dynamical systems learnable?

Let M be a topological space and $\phi \in C(M, M)$ a continuous discrete-time dynamical system with flow $F : M \times \mathbb{Z}_+ \longrightarrow M$

$$F_t(m) := \phi^t(m), \quad m \in M, \quad t \in \mathbb{Z}_+.$$

The time evolution is given by

$$m_0, m_1 = \phi(m_0), m_2 = \phi(m_1) = \phi^2(m_0), \ldots$$

Are dynamical systems learnable?

Consider a one-dimensional observation of the dynamical system trajectories given by the observation map $\omega \in C^2(M, \mathbb{R})$:

$$u_0 := \omega(m_0), u_1 := \omega(m_1), u_2 := \omega(m_2), \ldots$$

Question: Can we learn/reconstruct the (potentially high-dimensional) dynamics induced by ϕ out of one-dimensional observations?

Takens' Embedding Theorem

Theorem

M compact of dimension q and $\phi \in \mathrm{Diff}^2(M)$ with finitely many periodic points with period less or equal than $2q$. Suppose that the linearized Poincaré maps at periodic points with period smaller than $2q$ have distinct eigenvalues.

Then, for a generic observation function $\omega \in C^2(M, \mathbb{R})$, the $(2q+1)$-delay observation map

$$\begin{array}{rccc} \Phi_{(\phi,\omega)} : & M & \longrightarrow & \mathbb{R}^{2q+1} \\ & m & \longmapsto & \left(\omega(m), \omega(\phi(m)), \omega(\phi^2(m)), \ldots, \omega(\phi^{2q}(m))\right) \end{array}$$

is a C^1 embedding.

Take m_0, get

$\Phi_{(\phi,\omega)}(m_0) = (\omega(m_0), \omega(m_1), \omega(m_2), \ldots, \omega(m_{2q}) = (u_0, u_1, u_2, \ldots, u_{2q})$

This theorem yields learnability!

The dynamics induced by ϕ on M and that induced by the map $F := \Phi_{(\phi,\omega)} \circ \phi \circ \Phi^{-1}_{(\phi,\omega)}$ on $\Phi_{(\phi,\omega)}(M) \subset \mathbb{R}^{2q+1}$ are topologically conjugate. Moreover, if we denote by

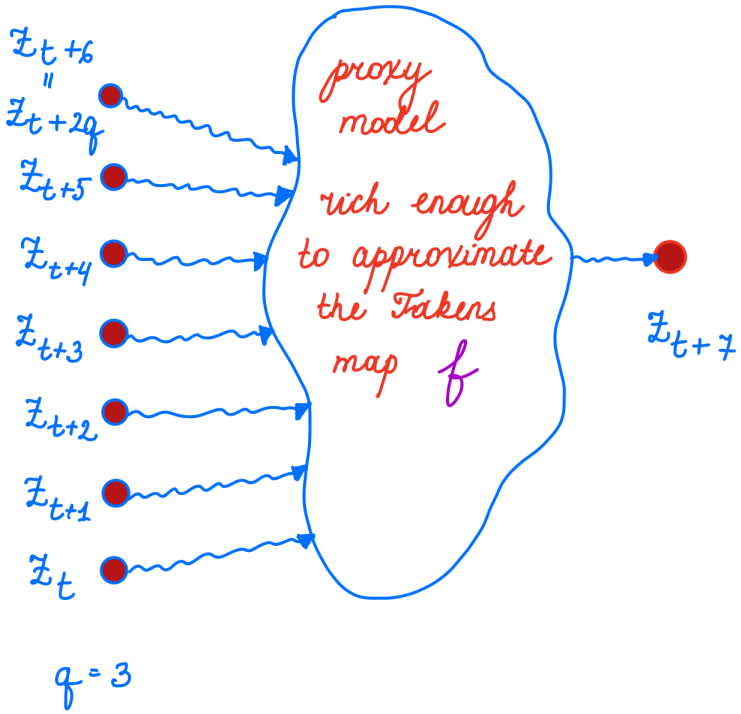
$$\mathbf{z}_t := (z_t, z_{t+1}, \ldots, z_{t+2q}) := \left(\omega(m_t), \omega(\phi(m_t)), \ldots, \omega(\phi^{2q}(m_t))\right)$$

then, the dynamical system $\mathbf{z}_{t+1} = F(\mathbf{z}_t)$ is determined by the map $f : \mathbb{R}^{2q+1} \longrightarrow \mathbb{R}$ such that

$$F\left(z_t, z_{t+1}, \ldots, z_{t+2q}\right) = \left(z_{t+1}, z_{t+2}, \ldots, f\left(z_t, z_{t+1}, \ldots, z_{t+2q}\right)\right)$$

Input-output systems

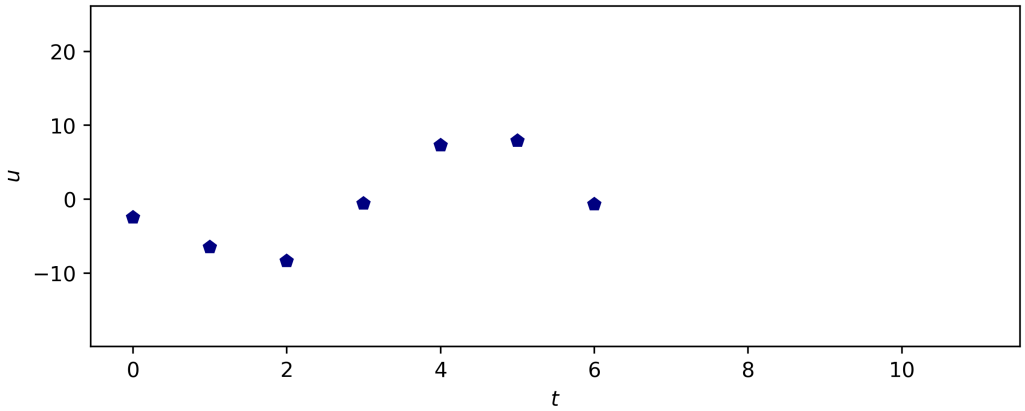
For any set of t values use given data $(z_t, z_{t+1}, \ldots, z_{t+2q})$ as inputs and target outputs z_{t+2q+1} to learn $f : \mathbb{R}^{2q+1} \longrightarrow \mathbb{R}$



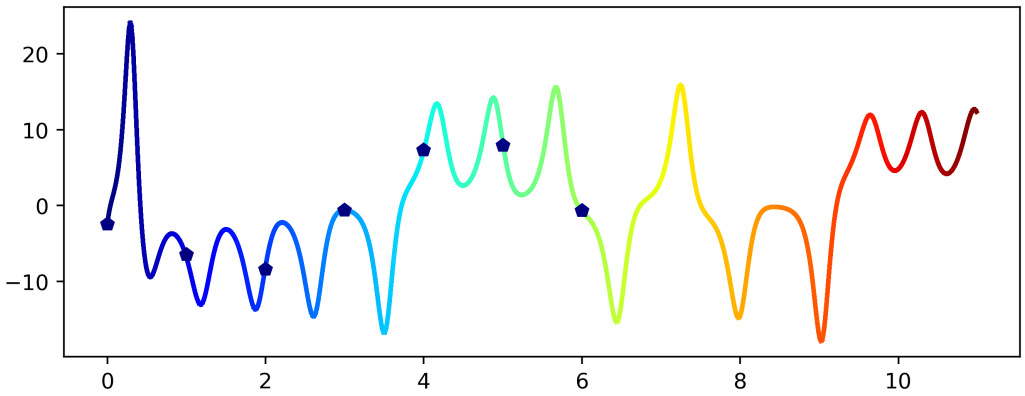
$$F\left(z_t, z_{t+1}, \ldots, z_{t+2q}\right) = \left(z_{t+1}, z_{t+2}, \ldots, f\left(z_t, z_{t+1}, \ldots, z_{t+2q}\right)\right)$$

The dynamical system ϕ can be learnt up to C^1 diffeomorphisms out of one-dimensional observations by learning the function $f : \mathbb{R}^{2q+1} \longrightarrow \mathbb{R}$

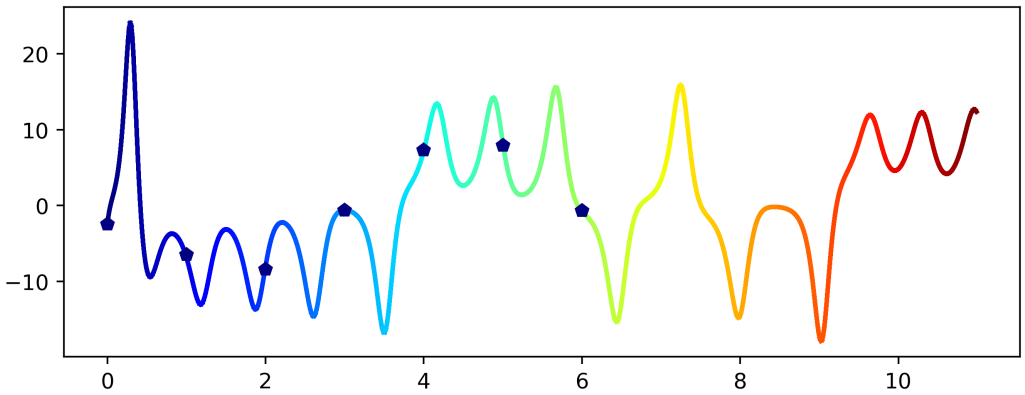
Takens map

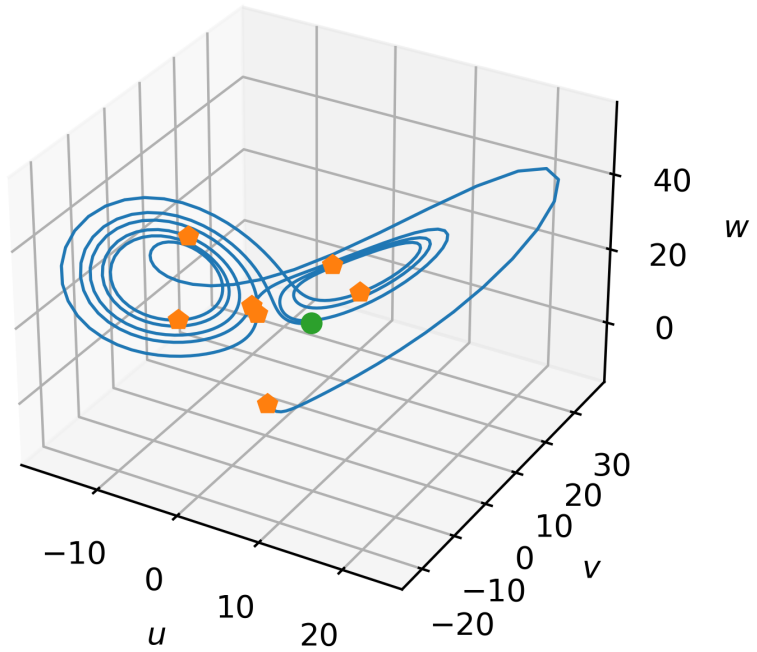


Takens map



Takens map





Are we done?

- The delay embedding idea has given rise to an entire embedology literature with applications to forecasting and time series analysis. See [KS03, HM15].
- C^1 conjugacy implies that the two systems have the same C^1 invariants (for example, eigenvalues of linearizations, Lyapunov exponents, dimensions of attractors, ...) but close points in phase space for one may be distant for the other. Problematic for forecasting. Recent advances in geometry preserving embeddings [EYWR18].

Outline for section 4

Statistical learning problem

- Input $\mathbf{Z} = (\mathbf{Z}_t)_{t\in\mathbb{Z}_-}$, $\mathbf{Z}_t \in D_d \subset \mathbb{R}^d$, target $\mathbf{Y} = (\mathbf{Y}_t)_{t\in\mathbb{Z}_-}$, $\mathbf{Y}_t \in \mathbb{R}^m$
- $\mathcal{F} := \{H\colon (D_d)^{\mathbb{Z}_-} \to \mathbb{R}^m \mid H \text{ measurable}\}$
- *Loss* $L\colon \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^+$
- *Statistical risk* or *generalization error* associated to $H \in \mathcal{F}$

$$R(H) := \mathbb{E}[L(H(\mathbf{Z}), \mathbf{Y}_0)].$$

- Ultimate goal of the learning - find the *Bayes functional* $H^*_{\mathcal{F}} \in \mathcal{F}$ such that

$$R(H^*_{\mathcal{F}}) = \inf_{H\in\mathcal{F}} R(H)$$

with $R^*_{\mathcal{F}} := R(H^*_{\mathcal{F}})$ the *Bayes risk* (for deterministic setup and under realizability assumption $R^*_{\mathcal{F}} = 0$ a.s.)

- Inductive bias: pick *hypothesis class* $\mathcal{H}$ of admissible functionals $\mathcal{H} \subset \mathcal{F}$
- More feasible goal - find the *best-in-class* or *oracle functional* $H^*_{\mathcal{H}} \in \mathcal{H}$ which solves

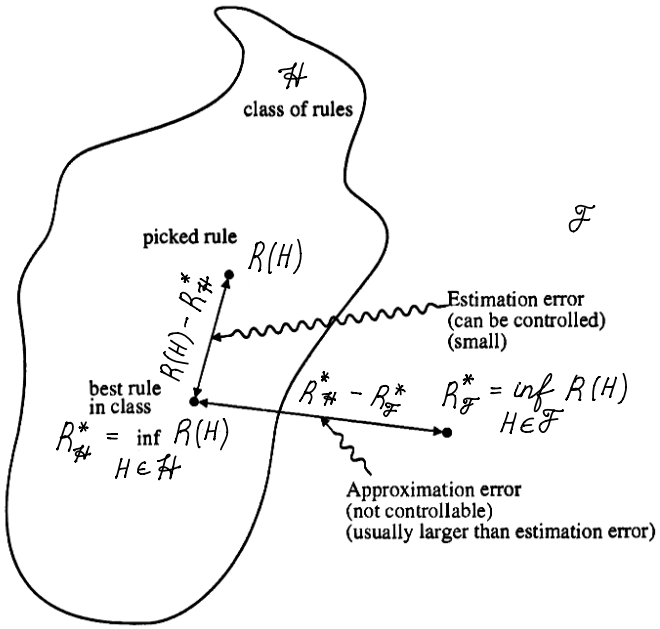
$$R(H^*_{\mathcal{H}}) = \inf_{H\in\mathcal{H}} R(H)$$

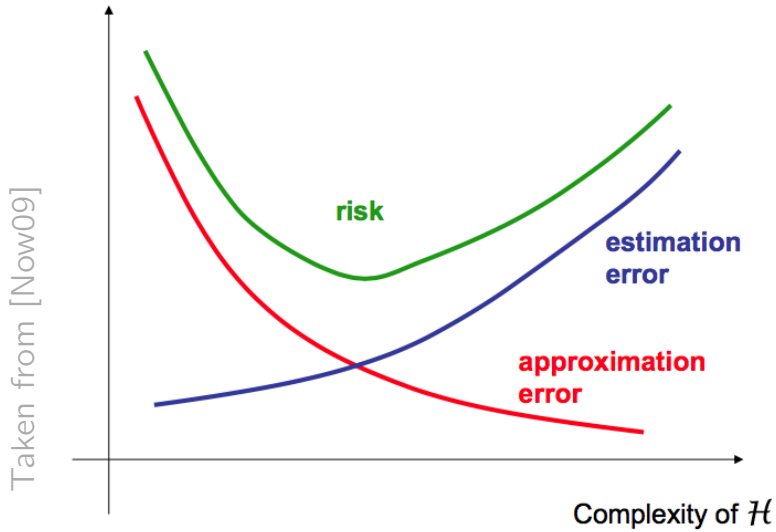
with $R^*_{\mathcal{H}} := R(H^*_{\mathcal{H}})$ the *Bayes in-class risk*

Excess risk decomposition

(Statistical) learning task consists in picking a hypothesis class and designing a *learning rule* which chooses $H \in \mathcal{H}$ with the smallest possible associated excess error

$$R(H) - R^*_{\mathcal{F}} = \underbrace{(R(H) - R^*_{\mathcal{H}})}_{\text{estimation error}} + \underbrace{(R^*_{\mathcal{H}} - R^*_{\mathcal{F}})}_{\text{approximation error}}$$





Learning rules: Empirical risk minimization (ERM), Structural risk minimization (SRM), Gibbs sampling.

Provable Universality

- Universal approximation properties for input-output systems
  - in the category of fading memory (FMP) filters: for uniformly bounded inputs defined on negative infinite times[3], for stochastic almost surely uniformly bounded inputs [1]
  - with respect to L^p-type criteria for stochastic discrete-time semi-infinite inputs[3]
  - for FMP and ESP filters for unbounded inputs[4]
- Explicit approximation error bounds available[5]

1 Grigoryeva, L. and Ortega, J.-P. 2018. Universal discrete-time reservoir computers with stochastic inputs and linear readouts using non-homogeneous state-affine systems. *Journal of Machine Learning Research*, 19(24), 1-40.

2 Grigoryeva, L. and Ortega, J.-P. 2018. Echo state networks are universal. *Neural Networks*, 108, 495-508.

3 Gonon, L. and Ortega, J.-P. 2019. Reservoir computing universality with stochastic inputs. *IEEE Transactions on Neural Networks and Learning Systems*, 31(1), 100-112.

4 Gonon, L., and Ortega, J.-P. 2020. Fading memory echo state networks are universal. *Preprint*.

5 Gonon, L., Grigoryeva, L., and Ortega, J.-P. 2020. Approximation bounds for random neural networks and reservoir systems. *Preprint*.

Ingredients and properties

We focus on state-space systems that determine an **input/output** system which happens at presence of the **echo state property (ESP)**.

- **ESP** at the level of the system: for any $\mathbf{z} \in (D_d)^{\mathbb{Z}_-}$ there exists a unique $\mathbf{y} \in (\mathbb{R}^m)^{\mathbb{Z}_-}$ such that (1)-(2) hold. Define **state-space filter** $U_h^F : (D_d)^{\mathbb{Z}_-} \longrightarrow (\mathbb{R}^m)^{\mathbb{Z}_-}$ as:
$$U_h^F(\mathbf{z}) := \mathbf{y},$$
where $\mathbf{z} \in (D_d)^{\mathbb{Z}_-}$ and $\mathbf{y} \in (\mathbb{R}^m)^{\mathbb{Z}_-}$ are linked by (1) via the ESP.
- **ESP** at the level of the state equation. Define a **state filter** $U^F : (D_d)^{\mathbb{Z}_-} \longrightarrow (D_N)^{\mathbb{Z}_-}$ for which
$$U_h^F := h \circ U^F.$$

Filters are causal and TI (see [GO18]).

U_h^F determines a **state-space functional** $H_h^F : (D_d)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}^m$ as $H_h^F(\mathbf{z}) := U_h^F(\mathbf{z})_0$, for all $\mathbf{z} \in (D_d)^{\mathbb{Z}_-}$.

- **Fading Memory Property**: CTI filter $U : V_d \subset \ell_-^w(\mathbb{R}^d) \to V_m \subset \ell_-^w(\mathbb{R}^m)$ is FMP w.r.t. w if $U : (V_d, \|.\|_w) \longrightarrow (V_m, \|.\|_w)$ is continuous, that is for any $\mathbf{z} \in V_d$, any $\epsilon > 0$ there exists $\delta(\epsilon, \mathbf{z}) > 0$ such that
$$\|\bar{\mathbf{z}} - \mathbf{z}\|_w < \delta(\epsilon, \mathbf{z}) \implies \|U(\bar{\mathbf{z}}) - U(\mathbf{z})\|_w < \epsilon, \quad \text{for any } \bar{\mathbf{z}} \in V_d.$$

Sufficient conditions for ESP

Sufficient conditions for the ESP to hold in general systems have been formulated in [GO18, GO19, GGO20].

Proposition (ESP for contracting maps with compact target)

Let $F : D_N \times D_d \longrightarrow D_N$ be a continuous state map such that D_N is a compact subset of $\mathbb{R}^N$ and F is a contraction on the first entry with constant $0 < c < 1$, that is,

$$\|F(\mathbf{x}_1, \mathbf{z}) - F(\mathbf{x}_2, \mathbf{z})\| \leq c\|\mathbf{x}_1 - \mathbf{x}_2\|,$$

for all $\mathbf{x}_1, \mathbf{x}_2 \in D_N$, $\mathbf{z} \in D_d$. Then, the associated system has the ESP for any input in $(D_d)^{\mathbb{Z}_-}$. The associated filter $U^F : (D_d)^{\mathbb{Z}_-} \longrightarrow (D_N)^{\mathbb{Z}_-}$ is continuous with respect to the product topologies in $(D_d)^{\mathbb{Z}_-}$ and $(D_N)^{\mathbb{Z}_-}$.

Example of Universal RC: Echo State Network (ESN)

Echo State Network is given by:

$$\begin{cases} \mathbf{x}_t = \boldsymbol{\sigma}\left(A\mathbf{x}_{t-1} + \gamma C\mathbf{z}_t + s\boldsymbol{\zeta}\right) \\ \mathbf{y}_t = W^\top \mathbf{x}_t + \boldsymbol{a} \end{cases}$$

The reservoir map $F : \mathbb{R}^N \times \mathbb{R}^d \to \mathbb{R}^N$ is prescribed by:

- the activation function $\boldsymbol{\sigma} : \mathbb{R}^N \longrightarrow \mathbb{R}^N$
- reservoir matrix $A \in \mathbb{M}_N$
- input mask $C \in \mathbb{M}_{N,d}$
- input scaling $\gamma \in \mathbb{R}^+$
- input shift $\boldsymbol{\zeta} \in \mathbb{R}^N$
- input shift scaling $s \in \mathbb{R}^+$

Universality theorem: ESN

Availability of universality theorems:

Theorem (Echo state networks are universal, LG, J.-P. Ortega (2018))

Let $U : I_d^{\mathbb{Z}_-} \longrightarrow (\mathbb{R}^m)^{\mathbb{Z}_-}$ *be a causal and time-invariant filter that has the fading memory property. Then, for any* $\epsilon > 0$ *there is an echo state network (ESN)*

$$\begin{cases} \mathbf{x}_t = \sigma\left(A\mathbf{x}_{t-1} + C\mathbf{z}_t + \boldsymbol{\zeta}\right), \\ \mathbf{y}_t = W\mathbf{x}_t. \end{cases}$$

whose associated filter $U_{\mathrm{ESN}} : I_d^{\mathbb{Z}_-} \longrightarrow (\mathbb{R}^m)^{\mathbb{Z}_-}$ *satisfies that*

$$\left\| U - U_{\mathrm{ESN}} \right\|_\infty < \epsilon.$$

Extensions using L^p norms [GO20, GO21]

Learning Input-Output Systems with RC

Input-output problem for stochastic processes

- Input $\mathbf{Z} = (\mathbf{Z}_t)_{t\in\mathbb{Z}_-}$, $\mathbf{Z}_t \in D_d \subset \mathbb{R}^d$
- Target $\mathbf{Y} = (\mathbf{Y}_t)_{t\in\mathbb{Z}_-}$, $\mathbf{Y}_t \in \mathbb{R}^m$
- Sample paths (realizations) $\mathbf{Z}(\omega) = (\mathbf{Z}_t(\omega))_{t\in\mathbb{Z}_-}$ and $\mathbf{Y}(\omega) = (\mathbf{Y}_t(\omega))_{t\in\mathbb{Z}_-}$, $\omega \in \Omega$

Data-driven learning ingredients:

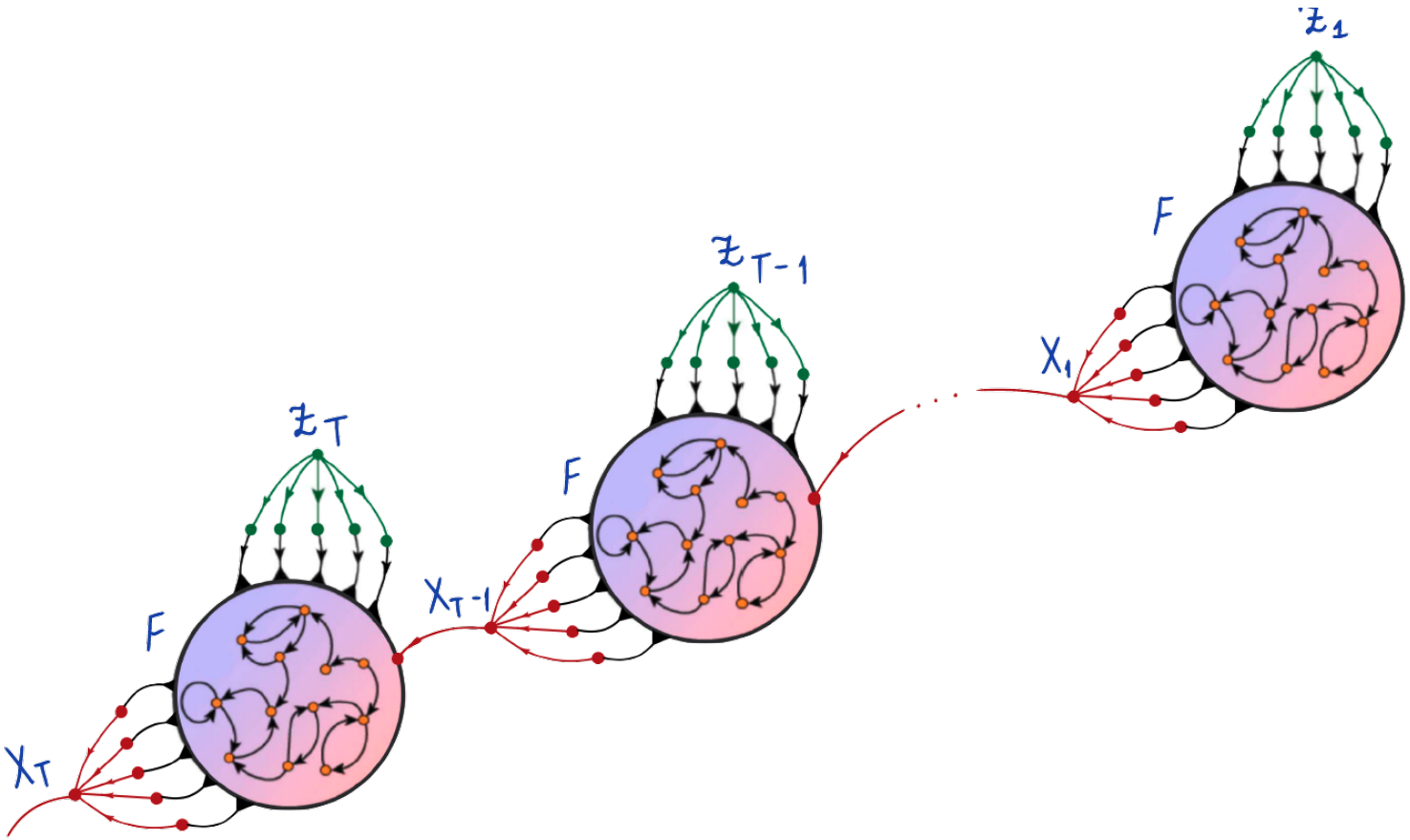
- class of candidate readout maps $\mathcal{H}_h$ and loss function $L : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^+$
- reservoir map F randomly generated and fixed
- total sample $\{(\mathbf{Z}_t, \mathbf{Y}_t)\}_{t=\{1,\ldots,T+p\}}$ with T reserved for training

Learning procedure: In the agnostic setup for any given $H_h^F \in \mathcal{H}$ $R(H_h^F)$ is not given but truncated empirical risk is $\hat{R}_n(H_h^F) = \frac{1}{T}\sum_{i=1}^{T} L(H_h^F(\mathbf{Z}_1^{T+1-i}), \mathbf{Y}_{T+1-i})$. Later on we discuss the implications. We often use the ERM procedure exclusively to choose $h \in \mathcal{H}_h$.

Learning Input-Output Systems with RC [VAL21]

1. Listening phase:

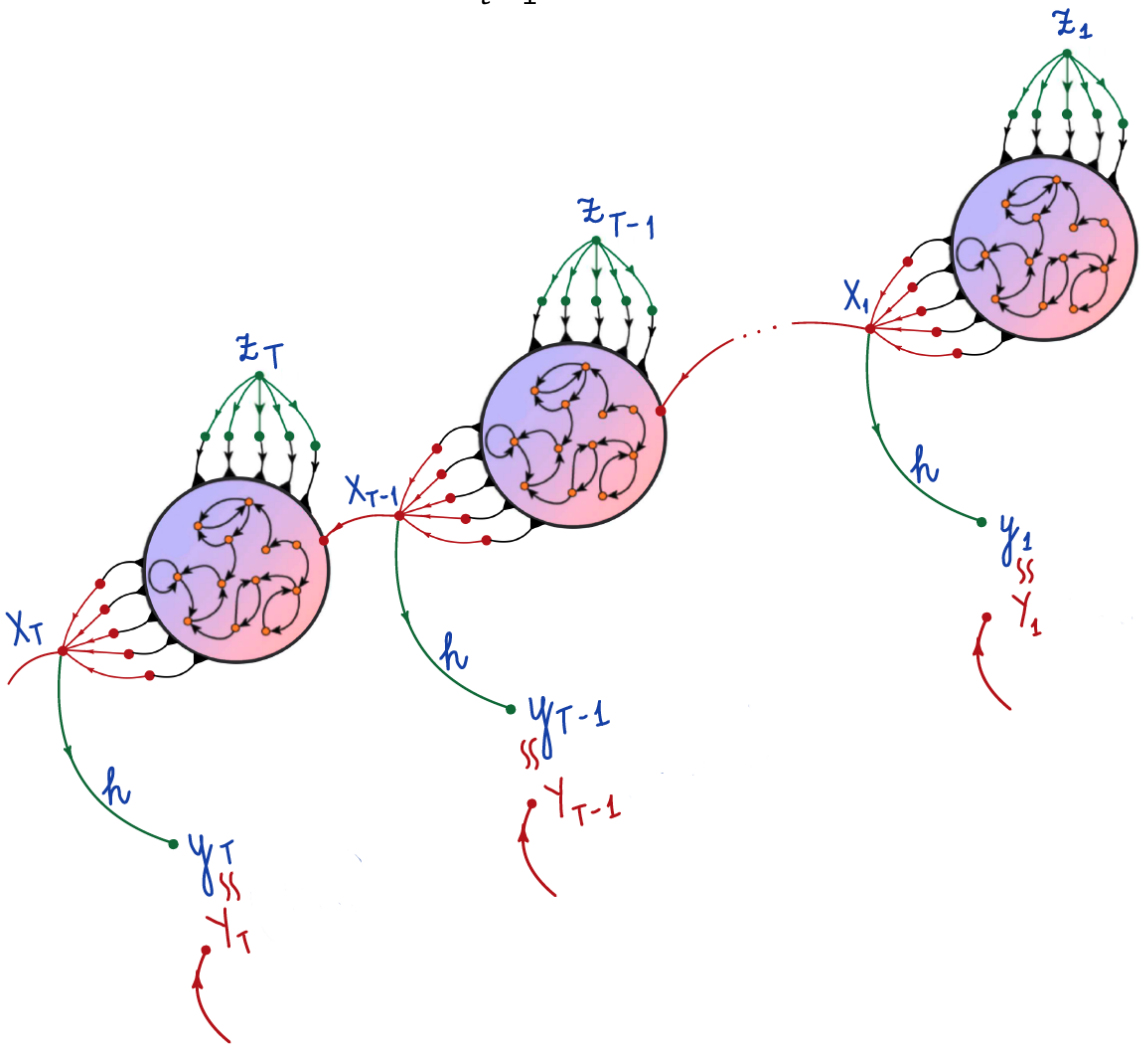
$$\mathbf{X}_t = F(\mathbf{X}_{t-1}, \mathbf{Z}_t), \quad \mathbf{X}_0 = \mathbf{0}, \quad t = 1, \ldots, T$$



Learning I/O Systems with RC

2. Fitting/training phase with (E/S)RM (general case target):

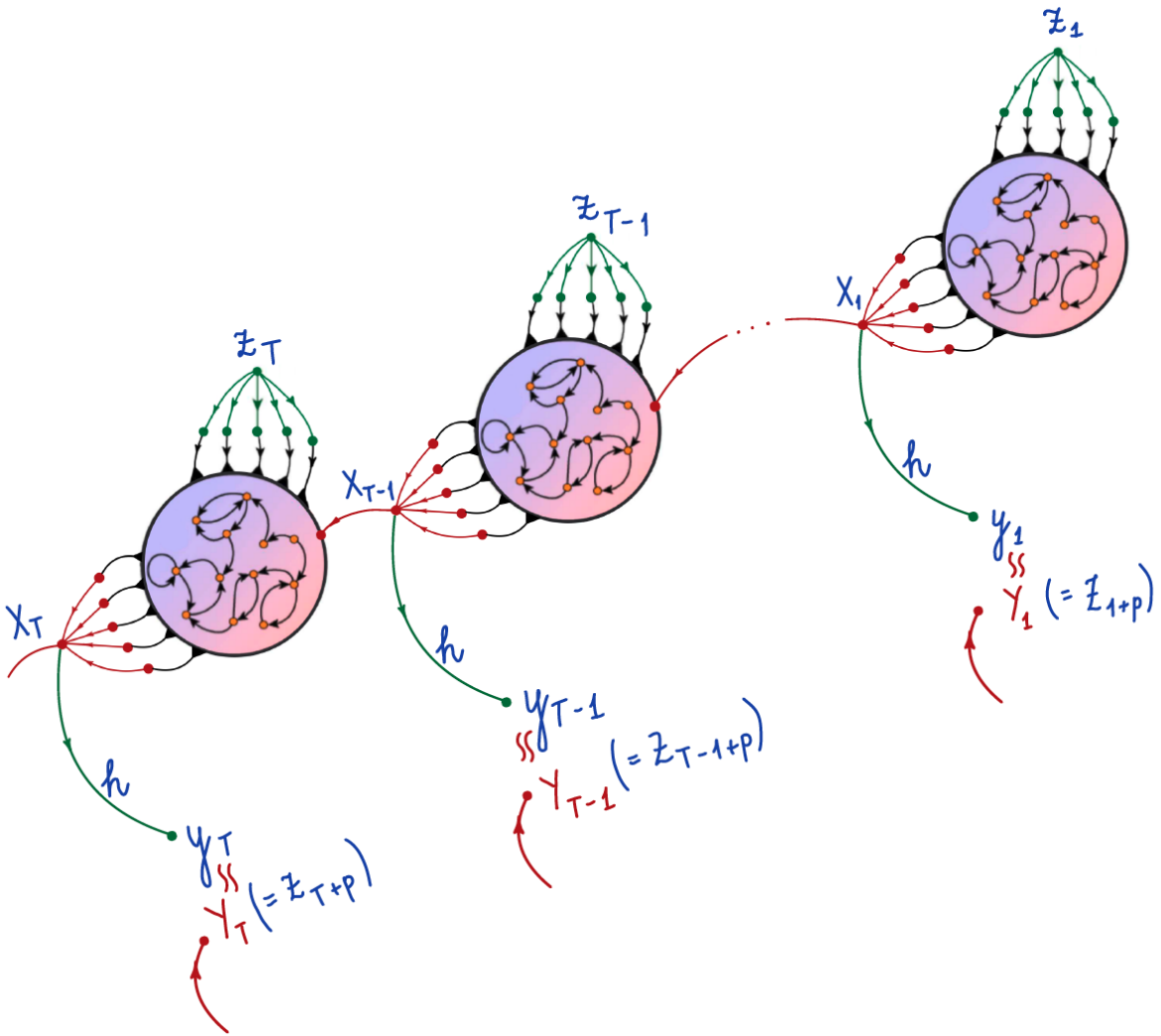
$$\widehat{h}_n \in \underset{h \in \mathcal{H}_h}{\arg\min}\{\frac{1}{T}\sum_{t=1}^{T} L(h(\boldsymbol{X}_t), \boldsymbol{Y}_t) + pen(h, T)\}$$



Learning I/O Systems with RC (Direct Forecasting)

2. Fitting/training phase with (E/S)RM (training target $\mathbf{Y}_t = \mathbf{Z}_{t+p}$):

$$\widehat{h}_n \in \underset{h \in \mathcal{H}_h}{\arg\min}\{\frac{1}{T-p}\sum_{t=1}^{T-p} L(h(\boldsymbol{X}_t), \boldsymbol{Z}_{t+p}) + pen(h, T, p)\}$$

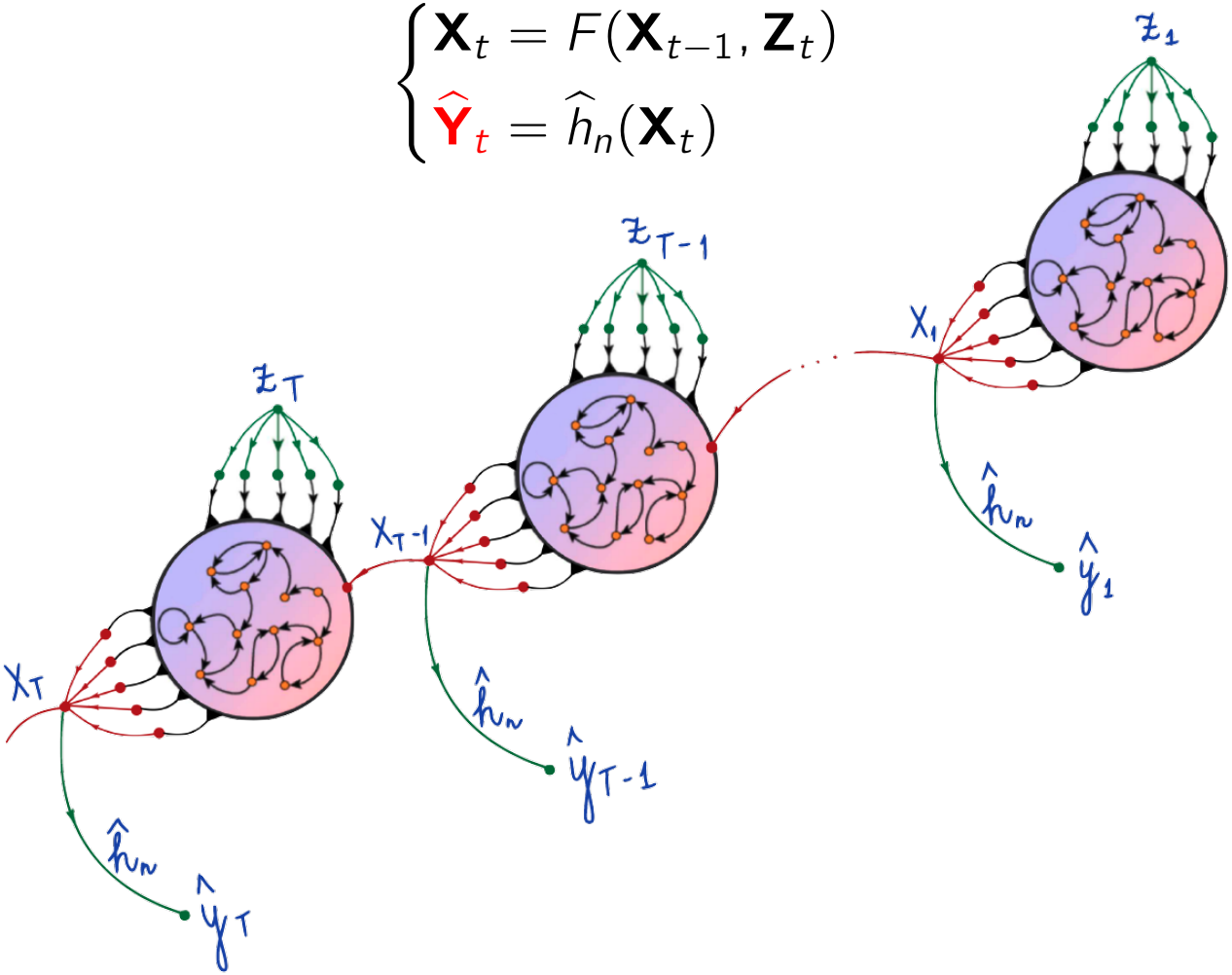


Learning Input-Output Systems with RC

3. Working phase:

   (a) input excited at all times (general case target)

$$\begin{cases} \mathbf{X}_t = F(\mathbf{X}_{t-1}, \mathbf{Z}_t) \\ \widehat{\mathbf{Y}}_t = \widehat{h}_n(\mathbf{X}_t) \end{cases}$$

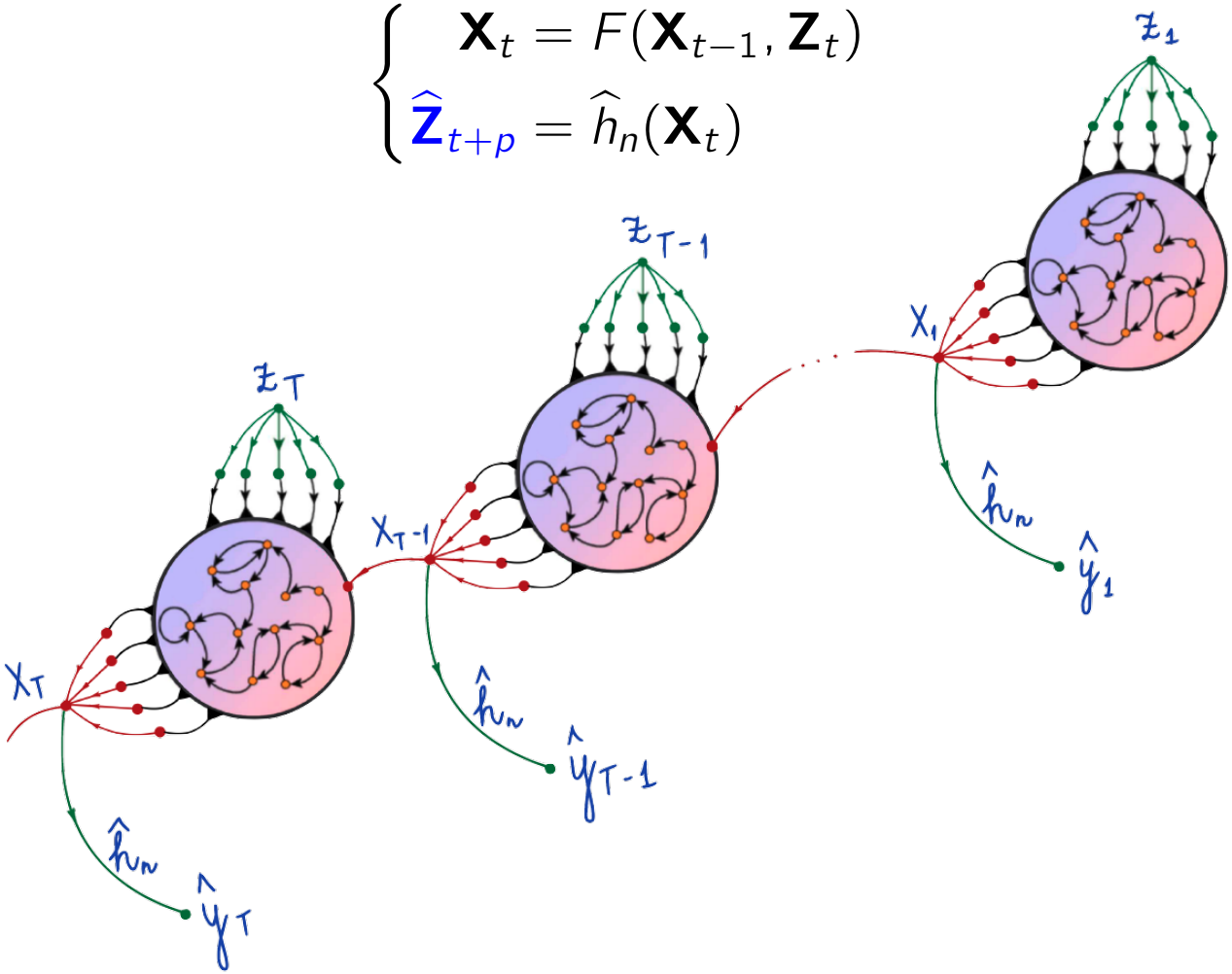


Learning I/O Systems with RC (Direct Forecasting)

3. Working phase:

   (a) input excited at all times (training target p-step forecast $\mathbf{Y}_t = \mathbf{Z}_{t+p}$)

$$\begin{cases} \mathbf{X}_t = F(\mathbf{X}_{t-1}, \mathbf{Z}_t) \\ \widehat{\mathbf{Z}}_{t+p} = \widehat{h}_n(\mathbf{X}_t) \end{cases}$$

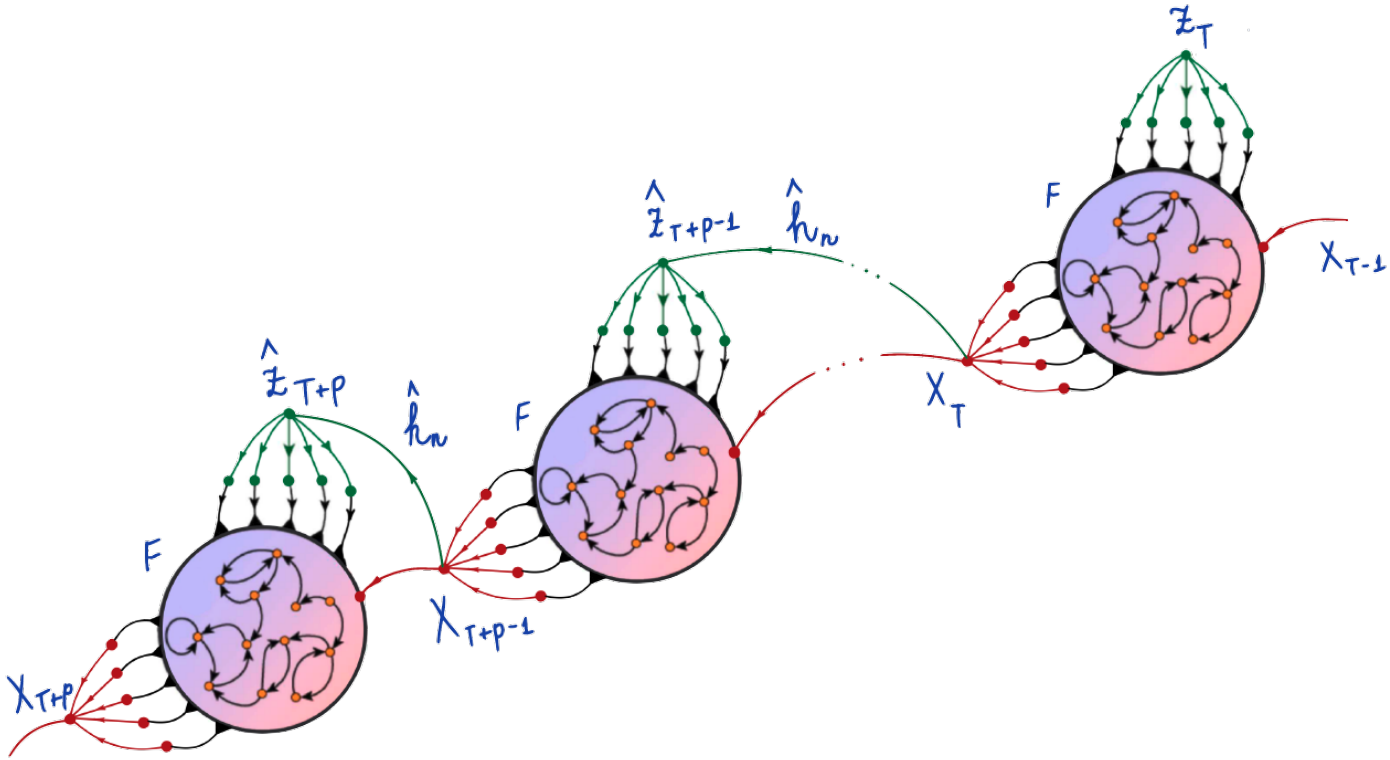


Learning I/O Systems with RC (Iterative Forecasting)

3. Working phase:

   (b) autonomous regime when the training target signal $\mathbf{Y}_t = \mathbf{Z}_{t+1}$

$$\begin{cases} \mathbf{X}_t = F(\mathbf{X}_{t-1}, \widehat{\mathbf{Z}}_t) \\ \widehat{\mathbf{Z}}_{t+1} = \widehat{h}_n(\mathbf{X}_t), \quad t = T+1, \ldots, T+p \end{cases}$$



Direct p-step forecasting

Definition

For any $\tau \in \mathbb{Z}_-$ $T_{-\tau} : (\mathbb{R}^d)^{\mathbb{Z}_-} \longrightarrow (\mathbb{R}^d)^{\mathbb{Z}_-}$ is the **time delay** operator defined by $T_{-\tau}(\mathbf{Z})_t := \mathbf{Z}_{t+\tau}$ for any $t \in \mathbb{Z}_-$.

Input $\mathbf{Z}_t$

Target $\mathbf{Y}_t = \mathbf{Z}_{t+p} = (T_{-p}(\mathbf{Z}))_t$

Goal: learn functional $H\colon (D_d)^{\mathbb{Z}_-} \longrightarrow (D_d)^{\mathbb{Z}_-}$, $D_d \subset \mathbb{R}^d$ which is given by $H(\mathbf{Z}) = p_t(T_{-p}(\mathbf{Z})) = p_{-p}(T_t(\mathbf{Z})) = (T_{t-p}(\mathbf{Z}))_0$ where for any $\tau \in \mathbb{Z}_-$ $p_\tau : (D_d)^{\mathbb{Z}_-} \longrightarrow D_d$ is the projection given by $p_\tau(\mathbf{Z}) = \mathbf{Z}_\tau$

Additional Structure: Bernoulli Shifts

Let $\mathbf{Z}$ be of a **causal Bernoulli shift** structure (see for instance [DDL$^+$07], [AW12]) that is $\mathbf{Z}_t = G(\ldots, \boldsymbol{\xi}_{t-1}, \boldsymbol{\xi}_t)$, $t \in \mathbb{Z}_-$ with $(\boldsymbol{\xi})_{t\in\mathbb{Z}_-}$ independent and identically distributed $\mathbb{R}^q$-valued random variables, $G : (\mathbb{R}^q)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}^d$ TI causal and measurable, and $\mathbb{E}[\|\mathbf{Z}_0\|_2] < \infty$ ($\mathbf{Z}$ is (strictly) stationary).

Input $\mathbf{Z}_t = \boldsymbol{\xi}_t, \quad t \in \mathbb{Z}_-$

Target $\mathbf{Y}_t = \mathbf{Z}_{t+p} = T_{-p}(\mathbf{Z})_t = G(T_{-p}(\boldsymbol{\xi}))_t$

Goal: learn functional $H\colon (D_q)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}^q$, $D_d \subset \mathbb{R}^d$ which is given by $H(\boldsymbol{\xi}) = p_t(T_{-p}(G(\boldsymbol{\xi}))) = p_t(G(T_{-p}(\boldsymbol{\xi})))$.

Direct versus iterative p-step forecasting

Direct multistep: $\mathbb{E}[\mathbf{Z}_{T+h}|\mathcal{F}_T]$ with $\mathcal{F}_T = \sigma(\mathbf{X}_T, \ldots, \mathbf{X}_0)$

- Universality properties
- Forecasting capacity bounds of ESNs [GGO2020]

Iterative multistep: $\mathbb{E}[\mathbf{Z}_{T+1}|\mathcal{F}_T]$ with $\mathcal{F}_T = \sigma(\mathbf{X}_T, \ldots, \mathbf{X}_0)$, $\mathbb{E}[\mathbf{Z}_{T+2}|\widetilde{\mathcal{F}}_{T+1}]$ with $\widetilde{\mathcal{F}}_{T+1} = \sigma(\widehat{\mathbf{X}}_{T+1}, \mathbf{X}_T, \ldots, \mathbf{X}_0)$, ...

- Available results for linear time series models

[GGO2020] Gonon, L., Grigoryeva, L., and Ortega, J.-P. 2020. Memory and forecasting capacities of nonlinear recurrent networks. *Physica D*, 414, 132721, 1-13

Stationary inputs give stationary states

Stationarity hypotheses: $\mathbf{Z} : \Omega \longrightarrow (D_d)^{\mathbb{Z}_-}$ is stationary whenever $T_{-\tau}(\mathbf{Z}) \overset{\mathrm{d}}{=} \mathbf{Z}$, for any $\tau \in \mathbb{Z}_-$.

Corollary (GGO2020)

Let $F : D_N \times D_d \longrightarrow D_N$ be a state map that satisfies the hypotheses of Proposition [ESP for contracting maps with compact target] or that, more generally, has the ESP and the associated filter $U^F : (D_d)^{\mathbb{Z}_-} \longrightarrow (D_N)^{\mathbb{Z}_-}$ is continuous with respect to the product topologies in $(D_d)^{\mathbb{Z}_-}$ and $(D_N)^{\mathbb{Z}_-}$. If the input process $\mathbf{Z}$ is stationary, then so is the state $\mathbf{X} := U^F(\mathbf{Z}) : \Omega \longrightarrow (D_N)^{\mathbb{Z}_-}$ as well as the joint processes $(T_{-\tau}(\mathbf{X}), \mathbf{Z})$ and $(\mathbf{X}, T_{-\tau}(\mathbf{Z}))$, for any $\tau \in \mathbb{Z}_-$.

Remark

Second-order stationarity and stationarity are only equivalent for Gaussian processes.

Doob decomposition implications for forecasting

Theorem (Doob decomposition)

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $(\mathcal{F}_t)_{t\in\mathbb{N}^+}$ be a filtration of $\mathcal{F}$. Let $\mathbf{Z} = (Z_t)_{t\in\mathbb{N}^+}$ be an adapted stochastic process with $\mathbb{E}[|Z_t|] < \infty$ for all $t \in \mathbb{N}^+$. Then there exists a martingale $M = (M_t)_{t\in\mathbb{N}^+}$ and an integrable and predictable process $A = (A_t)_{t\in\mathbb{N}^+}$ with $A_1 = 0$, such that $Z_t = A_t + M_t$ for all $t \in \mathbb{N}^+$. This decomposition of $\mathbf{Z}$ with respect to $\mathbb{P}$ is almost surely unique.

For all $t \in \mathbb{N}^+$

$$A_t = \sum_{j=2}^{t} \left(\mathbb{E}[Z_j|\mathcal{F}_{j-1}] - Z_{j-1}\right), A_1 = 0 \tag{3}$$

$$M_t = Z_1 + \sum_{j=2}^{t} \left(Z_j - \mathbb{E}[Z_j|\mathcal{F}_{j-1}]\right), M_1 = Z_1 \tag{4}$$

verify the decomposition.

Doob decomposition for forecasting

Training sample $\{Z_t\}_{t=\{1,\dots,T\}}$ and let $\mathcal{F}_t = \sigma(\mathbf{X}_t, \mathbf{X}_{t-1}, \dots, \mathbf{X}_0)$. Denote $\hat{Z}_t := \mathbb{E}[Z_t|\mathcal{F}_{t-1}] = \widehat{h}_n(\mathbf{X}_{t-1}), t \in \{2, \dots, T\}$ a best mean square error predictor of $\hat{Z}_t$ given states history. In the end of the fitting/training phase at T:

$$A_T = \sum_{t=2}^{T} (\hat{Z}_t - Z_{t-1})$$

and

$$M_T = Z_1 + \sum_{t=2}^{T} (Z_t - \hat{Z}_t) = Z_1 + \sum_{t=2}^{T} \hat{\epsilon}_t,$$

where $\hat{\epsilon}_t := Z_t - \hat{Z}_t$ denote in sample residuals produced by the trained ESN during fitting phase and whose empirical distribution function is $\hat{F}_T(x) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_{[\hat{\epsilon}_t - \frac{1}{T}\sum_{j=1}^{T} \hat{\epsilon}_j, \infty)}(x)$, $x \in \mathbb{R}$.

Doob decomposition for forecasting

1-step ahead

$$Z_{T+1} = A_{T+1} + M_{T+1},$$

where $\hat{Z}_{T+1}$ is the forecast provided by RC

$$A_{T+1} = \underbrace{\sum_{t=2}^{T}(\hat{Z}_t - Z_{t-1})}_{A_T} + (\hat{Z}_{T+1} - Z_T)$$

$$M_{T+1} = \underbrace{Z_1 + \sum_{t=2}^{T}(Z_t - \hat{Z}_t)}_{M_T} + (\underbrace{Z_{T+1}}_{\text{not avail.}} - \hat{Z}_{T+1})$$

Doob decomposition for forecasting

1-step ahead

$$Z_{T+1} = A_{T+1} + M_{T+1},$$

where $\hat{Z}_{T+1}$ is the forecast provided by RC

$$A_{T+1} = \underbrace{\sum_{t=2}^{T} (\hat{Z}_t - Z_{t-1})}_{A_T} + (\hat{Z}_{T+1} - Z_T)$$

$$M_{T+1} = \underbrace{Z_1 + \sum_{t=2}^{T} (Z_t - \hat{Z}_t)}_{M_T} + \underbrace{(\underbrace{Z_{T+1}}_{\text{not avail.}} - \hat{Z}_{T+1})}_{\hat{\epsilon}_{T+1}}$$

Bootstrapping residuals approach

Definition (Mallows/Wasserstein metric)

For $r \geq 1$ let $\mathcal{F}$ denote the set of distribution functions F satisfying $\int_{-\infty}^{\infty} |x|^r dF(x) < \infty$. For $F, G \in \mathcal{F}$, the Mallows (Wasserstein) metric is defined as

$$d_r(F, G) = \inf_{\mathcal{Q}} \{\mathbb{E}[|X - Y|^r]\}^{1/r},$$

where $\mathcal{Q}$ is the set of all joint distributions of (X, Y) whose marginal distributions are F and G, respectively.

Theorem (Asymptotics of Mallows metric for bootstrap, [GGO2021, in preparation)

Let F be the true distribution of innovations and $\widehat{F}_T$ be the empirical distribution of residuals. It holds that

$$d_2(F, \widehat{F}_T) \xrightarrow[T \to \infty]{p} 0.$$

See Miguel and Olave (1999), Pascual, Romo and Ruiz, Febrero et al. (1995)

One-step forecast

Let N_b be the number of bootstrap replications. Let $\hat{E}^{(j)} := (\widehat{\epsilon}_1^{(j)}, \widehat{\epsilon}_2^{(j)}, \ldots, \widehat{\epsilon}_p^{(j)})$, $j = 1, \ldots, N_b$, blocks of sampled with replacement residuals from the training phase.

- 1-step ahead, for $j = 1, \ldots, N_b$:

$$A_{T+1} = \underbrace{\sum_{t=2}^{T} (\widehat{Z}_t - Z_{t-1})}_{A_T} + (\widehat{Z}_{T+1} - Z_T)$$

$$\widetilde{M}_{T+1}^{(j)} = \underbrace{Z_1 + \sum_{t=2}^{T} (Z_t - \widehat{Z}_t)}_{M_T} + \widehat{\epsilon}_1^{(j)}$$

$$\widetilde{Z}_{T+1}^{(j)} = A_{T+1} + \widetilde{M}_{T+1}^{(j)} \tag{5}$$

Construct mean forecast as:

$$\widehat{Z}_{T+1}^{\mathrm{RC,b}} = \frac{1}{N_b} \sum_{j=1}^{N_b} \widetilde{Z}_{T+1}^{(j)} = A_T + (\widehat{Z}_{T+1} - Z_T) + M_T + \frac{1}{N_b} \sum_{j=1}^{N_b} \widehat{\epsilon}_1^{(j)} = \widehat{Z}_{T+1} + \frac{1}{N_b} \sum_{j=1}^{N_b} \widehat{\epsilon}_1^{(j)} \approx \widehat{Z}_{T+1}.$$

Multistep forecast (iterative)

- 2-steps ahead, for $j = 1, \ldots, N_b$:

$$\widetilde{A}^{(j)}_{T+2} = A_{T+1} + (\widehat{Z}^{(j)}_{T+2} - \widetilde{Z}^{(j)}_{T+1})$$
$$\widetilde{M}^{(j)}_{T+2} = \widetilde{M}^{(j)}_{T+1} + \widehat{\epsilon}^{(j)}_{2}$$
$$\widetilde{Z}^{(j)}_{T+2} = \widetilde{A}^{(j)}_{T+2} + \widetilde{M}^{(j)}_{T+2},$$

where $\widetilde{Z}^{(j)}_{T+1}$ is given in (5) and $\widehat{Z}^{(j)}_{T+2}$ is obtained as

$$\begin{cases} \mathbf{X}_{T+1} = F(\mathbf{X}_T, \widetilde{Z}^{(j)}_{T+1}) \\ \widehat{Z}^{(j)}_{T+2} = \widehat{h}_n(\mathbf{X}_{T+1}) \end{cases}$$

Construct mean forecast as:

$$\widehat{Z}^{\mathrm{RC}}_{T+2} = \frac{1}{N_b} \sum_{j=1}^{N_b} \widetilde{Z}^{(j)}_{T+2}.$$

Multistep forecast (iterative)

- p-steps ahead, for $j = 1, \ldots, N_b$:

$$\widetilde{A}^{(j)}_{T+p} = \widetilde{A}^{(j)}_{T+p-1} + (\widehat{Z}^{(j)}_{T+p} - \widetilde{Z}^{(j)}_{T+p-1})$$
$$\widetilde{M}^{(j)}_{T+p} = \widetilde{M}^{(j)}_{T+p-1} + \widehat{\epsilon}^{(j)}_{p}$$
$$\widetilde{Z}^{(j)}_{T+p} = \widetilde{A}^{(j)}_{T+p} + \widetilde{M}^{(j)}_{T+p}$$

where $\widetilde{Z}^{(j)}_{T+p-1}$ is determined in the previous step and $\widehat{Z}^{(j)}_{T+p}$ is obtained as

$$\begin{cases} \mathbf{X}_{T+p-1} = F(\mathbf{X}_{T+p-2}, \widetilde{Z}^{(j)}_{T+p-1}) \\ \widehat{Z}^{(j)}_{T+p} = \widehat{h}_n(\mathbf{X}_{T+p-1}) \end{cases}$$

Construct mean forecast as:

$$\widehat{Z}^{\mathrm{RC}}_{T+p} = \frac{1}{N_b} \sum_{j=1}^{N_b} \widetilde{Z}^{(j)}_{T+p}.$$

Bootstrapped forecast intervals

Let $F_{z,p}$ be the distribution of Z_{T+p}. Given significance α the p-prediction interval is given as

$$[F_{z,p}^{-1}(\alpha), F_{z,p}^{-1}(1-\alpha)]$$

with $F_{z,p}^{-1}(\alpha)$ and $F_{z,p}^{-1}(1-\alpha)$ are the lower and upper bounds of the $100(1-2\alpha)\%$ prediction interval.

The bootstrapping approach allows to obtain the bootstrapped forecast intervals for all forecasting horizons that is

$$[\widetilde{F}_{z,p}^{-1}(\alpha), \widetilde{F}_{z,p}^{-1}(1-\alpha)]$$

If $\widetilde{F}_{z,p}$ is a bootstrap distribution function then its Monte-Carlo approximation based on N_b resamples is given by

$$\widetilde{F}_{z,p,N_b}(k) = \frac{\sum \mathbb{1}_{\{\widetilde{Z}_{T+p}^{(j)} \leq k\}}}{N_b}.$$

Theorem (GGO2021, in preparation)

$$d_2(F_{z,p}, \widetilde{F}_{z,p}) \xrightarrow[T\to\infty]{p} 0.$$

Open problems

- Density learning with RC
- GAN for generating stochastic processes
- Filling missing data with Extreme Learning Machines
- Online and ensemble learning with RC
- Manifold learning
- Reinforcement, transfer-, and meta-learning with RC

Outline for section 5

References I

Pierre Alquier and Olivier Wintenberger.

Model selection for weakly dependent time series forecasting.
Bernoulli, 18(3):883–913, 2012.

J. Dedecker, P. Doukhan, G. Lang, J. R. León, S. Louhichi, and C. Prieur.

Weak Dependence: With Examples and Applications.
Springer Science+Business Media, 2007.

Armin Eftekhari, Han Lun Yap, Michael B. Wakin, and Christopher J. Rozell.

Stabilizing embedology: geometry-preserving delay-coordinate maps.
Physical Review E, 97(2):022222, feb 2018.

Lukas Gonon, Lyudmila Grigoryeva, and Juan-Pablo Ortega.

Risk bounds for reservoir computing.
Journal of Machine Learning Research, 21(240):1–61, 2020.

Lyudmila Grigoryeva and Juan-Pablo Ortega.

Echo state networks are universal.
Neural Networks, 108:495–508, 2018.

Lyudmila Grigoryeva and Juan-Pablo Ortega.

Differentiable reservoir computing.
Journal of Machine Learning Research, 20(179):1–62, 2019.

Lukas Gonon and Juan-Pablo Ortega.

Reservoir computing universality with stochastic inputs.
IEEE Transactions on Neural Networks and Learning Systems, 31(1):100–112, 2020.

References II

Lukas Gonon and Juan-Pablo Ortega.

Fading memory echo state networks are universal.
Neural Networks, 138:10–13, 2021.

Jeremy P. Huke and Mark R. Muldoon.

Embedding and time series analysis.
Technical report, Manchester Institute for Mathematical Sciences. The University of Manchester, 2015.

Holger Kantz and Thomas Schreiber.

Nonlinear Time Series Analysis.
Cambridge University Press, second edition, 2003.

Robert Nowak.

Statistical Learning Theory: Lecture Notes, 2009.

Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott.

Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach.
Physical Review Letters, 120(2):24102, 2018.

Jaideep Pathak, Zhixin Lu, Brian R. Hunt, Michelle Girvan, and Edward Ott.

Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data.
Chaos, 27(12), 2017.

Pietro Verzelli, Cesare Alippi, and Lorenzo Livi.

Learn to synchronize, synchronize to learn.
Chaos: An Interdisciplinary Journal of Nonlinear Science, 31:083119, 2021.