Contents

1 Introduction

- Motivation: Political opinions, polls. Between which opinions do people switch the most according to a model that explains the polls evolution? Prediction of polls?

- Emulate decision making process with ABM based on assumption that people are influenced by friends and change opinion with higher chance if majority of a group has a different opinion inspired by the ABM in [Mis12]

- Mehr schreiben zu Regeln von Meinungsänderung. Mehr Quellen

- We want to find the dynamical rule that describes the evolution of the concentrations of a finite number of different opinions in a closed society. [?] used data-driven system identification methods to detect time-discrete governing equations that are Markvovian a in the case of a complete network, i.e. where every person in the society is in equally strong contact with every other person. We investigate the case of an incomplete network, especially the case in which there are two subcommunities within the society that have few links between each other. In this case, arguments used in [**?**] to identify a Markovian model do not hold up and we will need to use information from the past to describe the evolution of opinion concentrations.

- The exact reason will formally by derived in Section 2 by using the Mori-Zwanzig formalism [Zwa01, LL19, CHK02]. Inspired from problems in Physics (?), the Mori-Zwanzig formalism explains how in the case of only low-dimensional observations of a high-dimensional system being visible, the evolution of these observations of the full system can be described by replacing the missing information of the full system by past information of these visible observations. In the context of an ABM the full information would be the opinion of every agent and the observations the concentrations of each opinion among all agents, e.g. as in opinion polls. We will show how for this reason the Mori-Zwanzig formalism becomes directly applicable to the modelling of the evolution of opinion concentrationss.

- Subsequently, through MZ we will derive a non-linear autoregressive (NAR) model for the evolution of the observations (Section 2). In Section 3, we will present a method that detects coefficients of functions in these NAR models. This method will be demonstrated on an example in Section 4 and used to increase the accuracy of prediction of opinion concentrations in the case of an incomplete network in Section 5.

2 Derivation of a nonlinear autoregressive model using the Mori-Zwanzig formalism

Later on we will model the spread of opinions inside a closed society by an agent-based model that consists of a high number N of agents who change their opinions X_i within a finite set of possible opinions over discrete time steps according to a rule based on the opinions of themselves and other agents. That rule will be Markovian, or memory-free, i.e. the changes of opinions are only influenced by opinions in the current time step. This dynamic will be called the *microdynamics*. The state of the microdynamic at time t is denoted by $X^t = (X_1^t, \dots, X_N^t)^T$. The set of all possible of those states is denoted by $\mathbb{X}$.

From the individual opinions, we will observe only the concentrations of opinions, i.e. the ratios of agents with each of the M opinions. We are interested in identifying the dynamic of the evolution of the concentrations of opinions, which we call the *macrodynamic*. Identifying the dynamic of low-dimensional observations of a higher dimensional system is a typical set-up for the Mori-Zwanzig formalism. We will explain it here in a general way and show how one can derive a nonlinear autoregressive model for the macrodynamic. Later on we show how it can be applied to the specific case of the spread of opinions.

2.1 The deterministic case

We assume at first that the microdynamic is Markovian but deterministic.
We install the space of deterministic dynamics on the state space $\mathbb{X}$, the microdynamic, given by

$$\mathcal{F} := \{F : \mathbb{X} \to \mathbb{X}\}$$

Are we going to need $\mathcal{F}$ later?
Also F here is general, while below it is a fixed function

and define a particular dynamical system with the function $F \in \mathcal{F}$ as

$$X^{t+1} = F(X^t) \in \mathbb{X}. \tag{2.1}$$

Further, we define the space $\mathbb{Y}$ as the space of observations of the microdynamic and the space of functions that map values of the dynamical system (2.1) to $\mathbb{Y}$ as

$$\mathcal{G} := \{g : \mathbb{X} \to \mathbb{Y}\}.$$

Let the accessible, or *relevant*, variables be denoted by $x = \xi(X) \in \mathbb{Y}$. We suppose from here on that we have no knowledge of the state of the microdynamic at any point in time but instead have the value of the function ξ. Additionally, we define the subspace $\mathcal{H}$ of functions in $\mathcal{G}$ that depend only on these relevant variables and map to $\mathbb{Y}$ as

$$\mathcal{H} := \{h \in \mathcal{G} \mid \exists \tilde{h} \in \tilde{\mathcal{H}} : h(X) = \tilde{h}(\xi(X)) \; \forall X \in \mathbb{X}\} \text{ where } \tilde{\mathcal{H}} = \{\tilde{h} : \xi(\mathbb{X}) \to \mathbb{Y}\}.$$

Functions in $\mathcal{H}$ still depend on $X \in \mathbb{X}$ but the information of $\xi(X)$ is enough to evaluate them. We write $h(x)$ for $x \in \mathbb{Y}$ and define the notation $h(X) := h(\xi(X))$ for $X \in \mathbb{X}$. An example is

$$\mathbb{X} = \mathbb{R}^2, \xi(X) = X_1 + X_2, h(X_1, X_2) = (X_1 + X_2)^2 = \xi(X)^2. \tag{2.2}$$

In this case it is enough to know the value of $\xi(X)$ to evaluate the value of $h(X)$.
The goal is now to represent the evolution of a function from $\mathcal{G}$ with knowledge only about ξ but not of the state of the microdynamic X^t. As illustrated in the following diagram, instead of taking one step of the microdynamic and then evaluate a function $g \in \mathcal{G}$, we have only access to the observation $\xi(X)$ and want to evaluate $g(F(X))$

under the premise that $\xi(X) = x$.

$$\begin{tikzcd}
X \arrow[r, "F"] \arrow[d, "\xi"'] & F(X) \arrow[d, "\cancel{g}\ \xi"] \\
\xi(X) = x \arrow[r, "?"'] & \overset{\xi}{\cancel{g}}(F(X))
\end{tikzcd}$$

Maybe better to draw the diagram with ξ, and then extend to the case, where we want $g(F(X))$, with $g \in \mathcal{H}$?

This means, we want to install a dynamical system that is dependent only on $\xi(X^t)$. Let us consider the following simple example:

$$\mathbb{X} = \mathbb{R}^2, \xi(X) = X_1 + X_2, g(X) = X_1^2 + X_2. \tag{2.3}$$

Then if we only know $X_1 + X_2$, how do we compute $X_1^2 + X_2$? In contrast to h in the ~~upper~~ above example, $g \notin \mathcal{H}$. But the knowledge of $\xi(X)$ should not be ignored: Instead we define a function that should depend only on $\xi(X)$ but should approximate g. To this end, we define a projection operator $P : \mathcal{G} \to \mathcal{H}$ that maps a function depending on X to a function depending on $\xi(X)$. An intuitive example would be the conditional expectation

Here we already need μ!

$$(Pg)(x) = \mathbb{E}[g(X)|\xi(X) = x] = \frac{\int_{\mathbb{X}} g(X)\delta(\xi(X), x)d\mu(X)}{\int_{\mathbb{X}} \delta(\xi(X), x)d\mu(X)}, \tag{2.4}$$

correct math, we have $\frac{0}{0}$ here.

where $\delta(\xi(X), x) = 1$ if $\xi(X) = x$ and 0 otherwise.
This term represents exactly the question we should ask: What do we expect $g(X)$ to be if we know that $\xi(X) = x$? In order to evaluate this term, we assume a probability distribution μ over $\mathbb{X}$, so that when asking what $g(X)$ is, we assume that X is distributed by μ. However, the integrals in (2.4) are often infeasible if we do not know μ. We hence follow [LL19] and define P as the orthogonal projection onto a set of functions from $\mathcal{H}$. This set consists of the columns of $\varphi = [\varphi_1, \ldots, \varphi_L]$:

dim $\mathcal{H}=\infty$, while here we take an L-dim. approx.

$$(Pg)(x) := \varphi(x) < \varphi, \varphi >^{-1} < \varphi, g > \tag{2.5}$$

where $x \in \mathbb{Y}$ and the scalar product $< \cdot, \cdot >$ is defined as

Include dimensions to see, what this amounts to, eventually.

$$< f, g > := \int_{\mathbb{X}} f(X)^T g(X) d\mu(X), \tag{2.6}$$

which is the matrix-valued integral over the matrix-valued function $f^T g$. $< \varphi, \varphi >$ is a mass matrix that ensures that P is in fact an orthogonal projection.
The orthogonal projection has the property that Pg is the closest function in $span(\varphi)$

to g where closeness is measured by the scalar product, i.e. Pg minimizes

$$< g - Pg, g - Pg >= \int_{\mathcal{X}} (g - Pg)^T(X)(g - Pg)(X)d\mu(X), \tag{2.7}$$

which is the expected quadratic difference between g and Pg. If $\mathcal{H}$ is infinite-dimensional one would need an infinite number of functions to yield that $span(\varphi) = \mathcal{H}$. In this case one would have to choose a sufficiently rich set of functions so that $span(\varphi)$ covers those parts of $\mathcal{H}$ that are important.

In order to represent the evolution of a function in $\mathcal{H}$ over time, we define the complement of P as $Q = Id - P$.

With the Koopman operator [Koo31] $\mathcal{K}$ for the system (2.1) defined as the operator that maps a function $g \in \mathcal{G}$ to $g \circ F \in \mathcal{G}$, we consider the Dyson formula

$$\mathcal{K}^{t+1} = \sum_{k=0}^{t} \mathcal{K}^{t-k} P\mathcal{K}(Q\mathcal{K})^k + (Q\mathcal{K})^{k+1} \tag{2.8}$$

which can be proved by induction on t. The Dyson formula describes a way to iteratively split up the application of the Koopman operator to a function g into parts $P\mathcal{K}g$ and $Q\mathcal{K}g$. (2.8) yields, by application of both sides of the equation to ξ and evaluated at the initial value of the macrodynamic, X^0:

$$\begin{aligned}
(\mathcal{K}^{t+1}\xi)(X^0) &= \sum_{k=0}^{t} \mathcal{K}^{t-k}[P\mathcal{K}(Q\mathcal{K})^k\xi](X^0) + (Q\mathcal{K})^{t+1}\xi(X^0) \\
\Rightarrow \; \xi(X^{t+1}) &= \sum_{k=0}^{t} [P\mathcal{K}(Q\mathcal{K})^k\xi](x_{t-k}) + (Q\mathcal{K})^{t+1}\xi(X^0) \\
\text{Set } \rho^k := (Q\mathcal{K})^k\xi \Rightarrow \xi(X^{t+1}) &= \sum_{k=0}^{t} [P\mathcal{K}\rho^k](x_{t-k}) + \rho^{t+1}(X^0) \\
&= \sum_{k=0}^{t} [P(\rho^k \circ F)](\xi(x_{t-k})) + \rho^{t+1}(X^0).
\end{aligned} \tag{2.9}$$

↑ this ξ is here by mistake, I presume?

$\xi(x)$ doesn't make sense either, as $x \in \mathcal{Y}$.

We replaced X^{t-k} by x_{t-k} in the second step because the application of P to a function makes this function depend only on the relevant variables. We explicitly used the parentheses around the operator $P\mathcal{K}\rho^k$ and its equivalent formulations to indicate that P is a projection operator that works on the function $\mathcal{K}\rho^k$. This gives a new function in $\mathcal{H}$ that is applied to the argument x_{t-k}.

Since $\rho^0 = \xi$, we obtain that $P(\rho^0 \circ F) = P(\xi \circ F)$. This is usually referred to as the *optimal prediction* term since it is the best Markovian approximation of $\xi(X^{t+1})$, i.e. the best approximation that only uses $\xi(X_t)$. ✓ The sum in the last row of Equation (2.9) starting at $k = 1$ is referred to as the *memory terms*, since these terms use information from previous values of $\xi(X)$. The term $\rho^{t+1}(X^0)$ depending on the full state X^0 and not on the projection $\xi(X^0)$, is often called *noise*, because one does not have explicit access to it and can often only treat it as a stochastic influence. In total, the last row of Equation (2.9) is called the *Mori-Zwanzig equation*.

Substituting

P~~lugging in~~ the definition of P as the orthogonal projection onto basis functions as in Equation (2.5), we obtain

$$\begin{aligned}P(\rho^k \circ F)(y) &= \varphi(y) < \varphi, \varphi >^{-1} < \varphi, \rho^k \circ F > \\ &= \underbrace{\varphi(y)}_{\in\mathbb{R}^{m\times L}} \underbrace{< \varphi, \varphi >^{-1}}_{\in\mathbb{R}^{L\times L}} \underbrace{\int_{\mathbb{X}} \underbrace{\varphi(\xi(X))^T}_{\in\mathbb{R}^{L\times m}} \underbrace{\rho^k(F(X))}_{\mathbb{Y}\subset\cancel{\in}\mathbb{R}^m} d\mu(X)}_{\in\mathbb{R}^L} =: \varphi(y)h_k \in \mathbb{R}^m.\end{aligned} \tag{2.10}$$

is $\mathbb{Y} \subseteq \mathbb{R}^m$?

ah, ok

Where does this statement come from?

Since $PQ = 0$, we can assume no correlation between ρ^t and $\varphi(x)$ and replace ρ^{t+1} in the last row of Equation (2.9) by a Gaussian noise term in $\varepsilon^{t+1} \in \mathbb{R}^m$. We obtain the macrodynamic

$$x_{t+1} = \sum_{k=0}^{t} \varphi(x_{t-k})h_k + \varepsilon^{t+1}. \tag{2.11}$$

If we had access to the underling probability distribution μ over $\mathbb{X}$, we could explicitly compute the h_k. If this is not the case then at least we have derived the structure of the dynamic of x_{t+1}. By assuming a decay of the terms h_{t-k} with increasing k, we can approximate the dynamics by starting the sum in (2.11) with $k = t - p$ instead of $k = 0$ in order to obtain a feasible number of memory terms. Regarding the selection of an appropriate value for the *memory depth* p there are various methods such as Information Criteria [KK08, ADP14] or the L-curve method [HO93]. We have thus derived a nonlinear autoregressive model (NAR) over x.

h_k, right?

Note that this is not the classical form of NAR models, since the coefficients h_k are vector-valued and the basis functions φ are matrix-valued, opposed to having matrix-valued coefficients and vector-valued basis functions (Figure 1). It turns out that the classical NAR form is in fact a special case of the formulation we have derived here because by choosing scalar-valued functions $\tilde{\varphi}_1, \dots, \tilde{\varphi}_L$ and defining

$$\varphi(x) = \begin{pmatrix} \tilde{\varphi}_1(x) & \dots & \tilde{\varphi}_L(x) & 0 & \dots & & & & 0 \\ 0 & \dots & 0 & \tilde{\varphi}_1(x) & \dots & \tilde{\varphi}_L(x) & 0 & \dots & 0 \\ \vdots & & & & & & \ddots & & \\ 0 & \dots & & & & & 0 & \tilde{\varphi}_1(x) \dots & \tilde{\varphi}_L(x) \end{pmatrix} \in \mathbb{R}^{m\times mL} \tag{2.12}$$

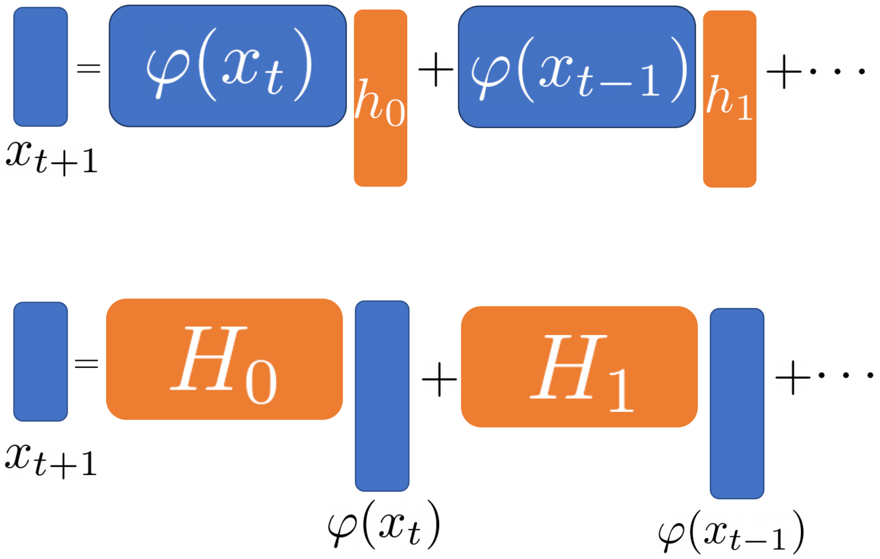


Figure 1: Top: Form of the nonlinear autoregressive model (2.11). Bottom: Form of classical nonlinear autoregressive models (2.14).

we find that

$$\varphi(x)\begin{pmatrix} h_1 \\ \vdots \\ h_{mL} \end{pmatrix} = \begin{pmatrix} h_1 & \dots & h_L \\ \vdots & & \vdots \\ h_{(m-1)L+1} & \dots & h_{mL} \end{pmatrix} \begin{pmatrix} \tilde{\varphi}_1(x) \\ \vdots \\ \tilde{\varphi}_L(x) \end{pmatrix} \tag{2.13}$$

✓

where h_i now denotes the i-th coordinate of a vector $h \in \mathbb{R}^{mL}$.

Since in the next section we will introduce a method that identifies matrix-valued coefficients for NAR models in a way that is motivated by system identification methods such as Dynamic Mode Decomposition [TRL$^+$], Extended Dynamic Mode Decomposition [WKR] or Sparse Identification of Nonlinear Dynamics [BPKa, BPKb], we use the slightly weaker formulation of (2.11) which reads

$$x_{t+1} = \sum_{k=0}^{p-1} H_k \tilde{\varphi}(x_{t-k}) + \varepsilon^{t+1}. \tag{2.14}$$

where $\tilde{\varphi} = [\tilde{\varphi}_1, \dots, \tilde{\varphi}_L]^T$ and $H_k = (h_k)_{ij} \in \mathbb{R}^{m \times L}$.

2.2 The stochastic case

Let us consider a stochastic dynamic

usually denoted by ω

$$X^{t+1} = F(X^t, \sigma) \tag{2.15}$$

We need a probability space $(\Omega, \mathcal{B}, \mathbb{P})$, such that we can define the expectation $\mathbb{E}$.

where $\sigma \in \Omega$ is a random influence on F which is now defined as $F : \mathbb{X} \times \Omega \to \mathbb{X}$. We have to slightly modify our definition of $\mathcal{F}$ from the deterministic case and define

$$\mathcal{F} = \{f : \mathbb{X} \times \Omega \to \mathbb{X}\}.$$

As the function that maps values of this system to $\mathbb{Y}$, let us now consider

$$\bar{\xi}(X^t) := \mathbb{E}[\xi(X^t)].$$

Here it matters what is assumed about the distribution of X^0.

This is still consistent with the way we derived the macrodynamic from the Dyson formula in Equation (2.9) since the Koopman operator for stochastic systems [riMM19] is defined as

$$(\mathcal{K} \circ g)(X) = \mathbb{E}[g(F(X, \sigma))].$$

$\mathbb{E}\left[\xi(X^{t+1}) \mid X^0 = z\right]$

This means that on the left-hand side in (2.9) side we obtain $(\mathcal{K}^{t+1}\xi)(\overset{\cancel{X^0}}{\underset{z}{}}) = \cancel{\mathbb{E}[\xi(X^{t+1})]} =$ $\bar{\xi}^t(z)$ $\cancel{\bar{\xi}(X^{t+1})}$. only, if $\bar{\xi}^t(z) := \mathbb{E}\left[\xi(X^t) \mid X^0 = z\right]$

For the right-hand side, the last step in (2.9) now has to be modified: We see that

$$P\mathcal{K}\rho^k(X^{t-k}) = P(\rho^k\mathbb{E}[F(X^{t-k}, \sigma)]$$
$$= \varphi(\xi(X^{t-k})) < \varphi, \varphi >^{-1} \int_\Omega \int_X \varphi(\xi(X))^T \rho^k(F(X, \sigma)) d\mu(X) d\tilde{\mu}(\sigma)$$

where $\tilde{\mu}$ is a probability density over Ω.

We can thus obtain the identical structure of the macrodynamic as in (2.11) where for the computation of the coefficients h_k in (2.10), the integral over Ω had to be added. Note that for this we did not even have to change the definition of P. When computing the expected value of a function in a stochastic system, the extension of the Koopman operator to the expected value takes care of the consistency of the steps that were done in the deterministic case.

please check this again in light of the above

3 Sparse Identification of Nonlinear Autoregressive Models (SINAR)

We propose here a new method of model identification for coefficients H_k in (2.11) that is an extension of the Sparse Identification of Nonlinear Dynamics (SINDy) algorithm from [BPKa, BPKb, KKB18]. SINDy can be used to identify the governing equations of a dynamical system from data: Let a Markovian dynamical system be given by

$$x_{t+1} = f(x_t) \in \mathbb{R}^m. \tag{3.1}$$

deterministic or stochastic?

Suppose f is unknown for us but we have access to data

$$\mathbf{X} = \begin{bmatrix} | & & | \\ x_0 & \dots & x_{T-1} \\ | & & | \end{bmatrix}, \mathbf{X}' = \begin{bmatrix} | & & | \\ x_1 & \dots & x_T \\ | & & | \end{bmatrix}$$

then in SINDy we try to approximate each coordinate of f by a linear combination of basis functions $\theta_i : \mathbb{R}^m \to \mathbb{R}$ and define

$$\Theta(x) = \begin{bmatrix} \theta_1(x) \\ \vdots \\ \theta_v(x) \end{bmatrix}, \Theta(\mathbf{X}) = \begin{bmatrix} \theta_1(x_0) & \dots & \theta_1(x_{T-1}) \\ \vdots & & \vdots \\ \theta_v(x_0) & \dots & \theta_v(x_{T-1}) \end{bmatrix}, \tag{3.2}$$

To this end, we fit a coefficient matrix $\Xi \in \mathbb{R}^{m \times v}$ to the data $\mathbf{X}, \mathbf{X}'$ by finding

$$\Xi = \arg\min_{\Xi} \|\mathbf{X}' - \Xi\Theta(\mathbf{X})\|_F \tag{3.3}$$

so that

$$x_{t+1} \approx \Xi\Theta(x_t). \tag{3.4}$$

In order to only obtain the basis functions from Θ that are dominant for the relation between x_{t+1} and $\Theta(x_t)$, we enforce a sparsity constraint using the LASSO regression algorithm [Tib] in which a regularisation term is added onto the coefficient matrix. For every row $\mathbf{X}'_i$ of $\mathbf{X}'$, we solve

$$\Xi_i = \arg\min_{\Xi_i} \|\mathbf{X}'_i - \Xi_i\Theta(\mathbf{X})\|_F + \lambda\|\Xi_i\|_1. \tag{3.5}$$

The use of the 1-norm yields that the solution will be sparse if we set $\lambda > 0$ appropriately. Sparse models will often times be less accurate than non-sparse models. However, what we gain through a sparse right-hand side of (3.4) is a better interpretability of the model since only the dominant terms have been identified as influential to the dynamic. It is vital to set λ so that the loss of accuracy is minimal compared to the gain in interpretability.

Now, when we suspect that in the dynamical system (3.1) x_{t+1} depends not only on x_t but on memory terms, too, we can apply the SINDy algorithm to suitably transformed data to obtain a nonlinear autoregressive model as in (2.14) with sparse coefficients, i.e. with only few basis functions with non-zero coefficients: Selecting a memory depth

p and denoting

$$\tilde{x}_t := \begin{bmatrix} | \\ x_t \\ | \\ \vdots \\ | \\ x_{t-p+1} \\ | \end{bmatrix} \in \mathbb{R}^{mp}, \tag{3.6}$$

let us define as data matrices the *Hankel* matrix

$$\tilde{\mathbf{X}} = \begin{bmatrix} x_{p-1} & \dots & x_{T-1} \\ \vdots & & \vdots \\ x_0 & \dots & x_{T-p} \end{bmatrix} = \begin{bmatrix} | & & | \\ \tilde{x}_{p-1} & \dots & \tilde{x}_{T-1} \\ | & & | \end{bmatrix} \text{ and } \mathbf{X}' = \begin{bmatrix} | & & | \\ x_p & \dots & x_T \\ | & & | \end{bmatrix}.$$

This is a similar concept to the Hankel-DMD algorithm proposed in [AM17] or the Hankel-alternative view of Koopman (HAVOK) analysis from [BBP$^+$].
Again we choose basis functions

$$\tilde{\Theta}(x) = \begin{bmatrix} \tilde{\theta}_1(\tilde{x}) \\ \vdots \\ \tilde{\theta}_v(\tilde{x}) \end{bmatrix}, \tilde{\Theta}(\tilde{\mathbf{X}}) = \begin{bmatrix} \tilde{\theta}_1(\tilde{x}_{p-1}) & \dots & \tilde{\theta}_1(\tilde{x}_{T-1}) \\ \vdots & & \vdots \\ \tilde{\theta}_v(\tilde{x}_{p-1}) & \dots & \tilde{\theta}_v(\tilde{x}_{T-1}) \end{bmatrix} \tag{3.7}$$

for example

$$\tilde{\Theta}(\tilde{X}_t) = \begin{bmatrix} (x_t)_1^2 \\ (x_t)_1(x_t)_2 \\ \vdots \\ \sin((x_{t-1})_1) \\ \vdots \\ (x_{t-2})_m(x_{t-3})_1 \end{bmatrix}, \tag{3.8}$$

and minimize for every row of $\mathbf{X}'$

$$\tilde{\Xi}_i = \arg\min_{\tilde{\Xi}_i} \|\mathbf{X}'_i - \tilde{\Xi}_i\tilde{\Theta}(\tilde{\mathbf{X}})\|_F + \lambda\|\tilde{\Xi}_i\|_1. \tag{3.9}$$

Then with the basis functions with non-zero coefficients in $\tilde{\Xi}$, we have derived a nonlinear autoregressive model that approximates the evolution of x:

$$\begin{aligned} ?\ \curvearrowright\ \tilde{x}_{t+1} &= \tilde{\Xi}\tilde{\Theta}(\tilde{x}_t) \\ \Leftrightarrow (x_{t+1})_i &= \sum_{j=1}^{v} \tilde{\Xi}_{ij}\tilde{\theta}_j(\tilde{x}_t). \end{aligned} \tag{3.10}$$

By deleting all columns of $\tilde{\Xi}$ that only contain zeros, which should be many if we enforce the sparsity constraint, we get a reduced matrix and thus a low number of terms on the right-hand side of (3.10). We have thus identified a sparse nonlinear autoregressive model so that we call this extension of SINDy Sparse Identification of Nonlinear Autoregressive Models (SINAR). Note that for a memory depth of $p = 1$, SINDy and SINAR are equivalent. Figure 2 illustrates the different structures of SINDy and SINAR.

define the term "autoreg. model" somewhere. Readers might not know what it means.

By choosing

$$\Theta(\tilde{x}_t) = \begin{bmatrix} \tilde{\varphi}_1(x_t) \\ \vdots \\ \tilde{\varphi}_L(x_t) \\ \vdots \\ \tilde{\varphi}_1(x_{t-p+1}) \\ \vdots \\ \tilde{\varphi}_L(x_{t-p+1}) \end{bmatrix} \tag{3.11}$$

with $\tilde{\varphi}_1, \dots, \tilde{\varphi}_L$ being scalar-valued functions as in Equation (2.12) in Section 2, we can directly fit the coefficients H_k of the model (2.14) which was derived through the Mori-Zwanzig formalism previously. Then $\tilde{\Xi}$ has the ~~form~~ blockwise form

$$\tilde{\Xi} = \begin{bmatrix} & \cancel{\mid} & & & & \cancel{\mid} & \\ \cancel{-} & H_0 & \cancel{-} & \dots & \cancel{-} & H_{p-1} & \cancel{-} \\ & \cancel{\mid} & & & & \cancel{\mid} & \end{bmatrix} \in \mathbb{R}^{m \times pL} \tag{3.12}$$

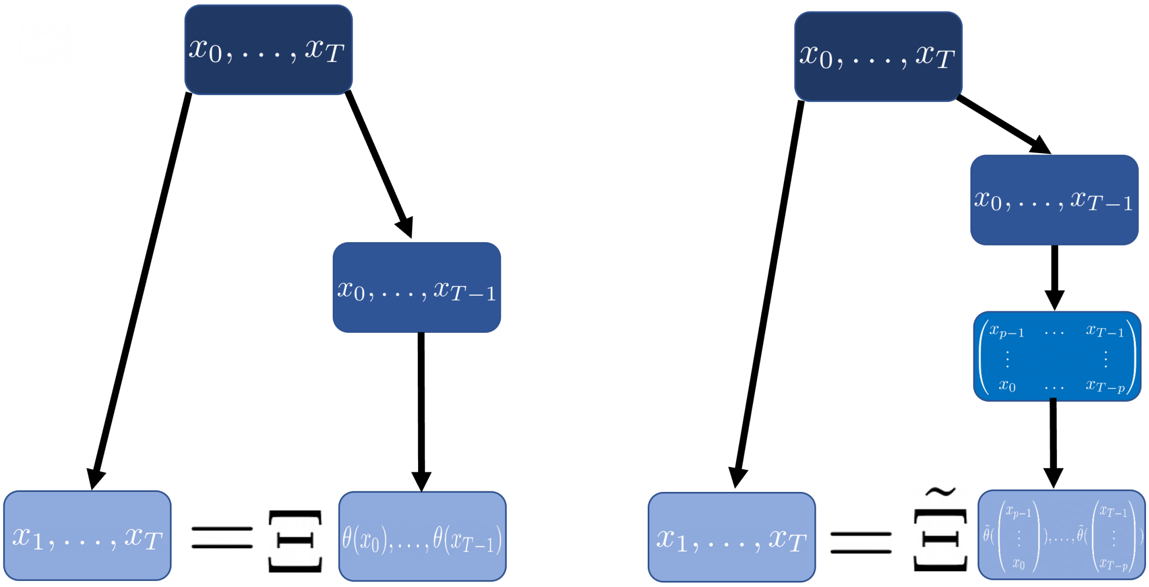
and

$$\tilde{\Xi}\tilde{\Theta}(\tilde{x}_t) = \sum_{k=0}^{p-1} H_k \begin{pmatrix} \tilde{\varphi}_1(x_{t-k}) \\ \vdots \\ \tilde{\varphi}_L(x_{t-k}) \end{pmatrix}. \tag{3.13}$$

The covariance of the noise term ε^{t+1} in (2.14) can be estimated in the common way for linear or nonlinear AR models [BD91, LL19] by calculating the statistical covariance between $\mathbf{X}'$ and $\tilde{\Xi}\tilde{\Theta}(\mathbf{X})$.

explain in an appendix.

- Please use the hyperref package, so that eqn numbers are hyperlinked
- Would be nice to see (briefly) in a remark that for linear dynamics and linear basis functions one retains the (usual?) AR model estimation. Is that true? (of course, without sparsification)


$x_0,\dots,x_T$
$x_0,\dots,x_{T-1}$
$x_1,\dots,x_T$
$=\Xi$
$\theta(x_0),\dots,\theta(x_{T-1})$
$x_0,\dots,x_T$
$x_0,\dots,x_{T-1}$
$\begin{pmatrix} x_{p-1} & \dots & x_{T-1} \\ \vdots & & \vdots \\ x_0 & \dots & x_{T-p} \end{pmatrix}$
$x_1,\dots,x_T$
$=\tilde{\Xi}$
$\tilde{\theta}\left(\begin{pmatrix} x_{p-1} \\ \vdots \\ x_0 \end{pmatrix}\right),\dots,\tilde{\theta}\left(\begin{pmatrix} x_{T-1} \\ \vdots \\ x_{T-p} \end{pmatrix}\right)$

✓

Figure 2: Sketch of the SINDy algorithm (left) and SINAR (right). SINAR contains the additional step of creating a Hankel matrix.

4 Example: An ~~augmented~~ extended Henon system

We will now demonstrate the emergence of memory terms in the case of inaccessible variables in the sense of the Mori-Zwanzig formalism on an example of a dynamical system and use SINAR to detect an NAR model that reconstructs the dynamic.
The classical Henon system [Hé76] describes a 2-dimensional system that is one of the most famous examples for systems with chaotic behaviour, i.e. where slightly deviated initial conditions lead to a significantly different trajectory. Its dynamic is given by

$$\begin{aligned} x_{t+1} &= 1 - a x_t^2 + y_t \\ y_{t+1} &= b x_t. \end{aligned} \tag{4.1}$$

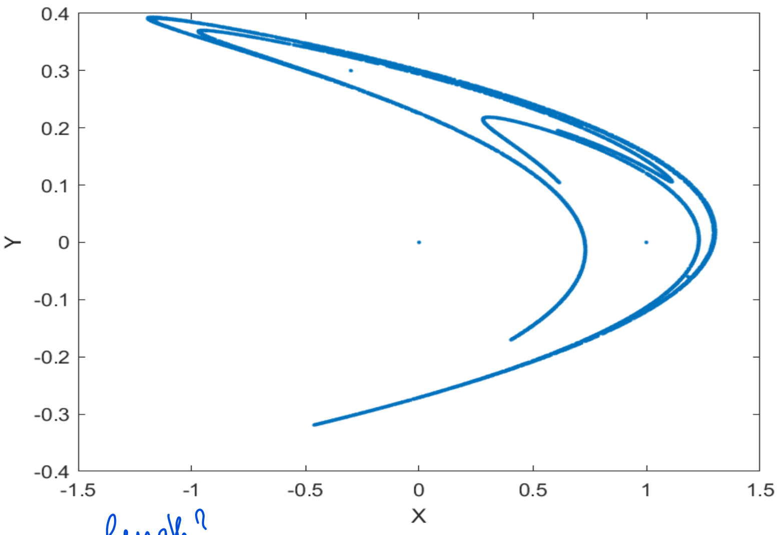
As we can observe, y_t is nothing more than a scaled and time-delayed version of x_t. If we now consider x as the relevant and y as the irrelevant variable - this means the in the Mori-Zwanzig formalism the space $\mathcal{H}$ is given by all functions depending on only x - , we can still express the evolution of x exactly with dependence on the past two values of x by plugging in the equation for y_{t+1} into the equation for x_{t+1}:

$$x_{t+1} = 1 - a x_t^2 + b x_{t-1}.$$

Let us now consider an augmented version of the Henon system

$$\begin{aligned} x_{t+1} &= 1 - a x_t^2 + y_t \\ y_{t+1} &= b x_t + c y_{t-1} \end{aligned} \tag{4.2}$$

that is visualized in Figure 3.



length?

Figure 3: Trajectory of the 2-dimensional augmented Henon system (4.2) with $a = 1.3, b = 0.3, c = 0.3$ and initial values $x_0 = y_0 = 0$. Optically, it is very similar to the classical Henon system but the additional term plays a crucial role in the prediction.

Now y is more than only a scaled and time-delayed version of x. If we try to express x_t only in dependence of its own past terms and without values of y then we do not get a system with a finite memory depth but an infinite one:

$$\begin{aligned} x_{t+1} &= 1 - ax_t^2 + bx_{t-1} + cy_{t-1} \\ &= 1 - ax_t^2 + bx_{t-1} + cbx_{t-2} + c^2 y_{t-2} \\ &= 1 - ax_t^2 + bx_{t-1} + cbx_{t-2} + c^2 b_{t-3} + c^3 y_{t-3} \qquad (4.3) \\ &= 1 - ax_t^2 + \sum_{j=1}^{t} c^{j-1} bx_{t-j} + c^{t+1} y_0, \end{aligned}$$

which can be quickly shown by induction on t.

We have hereby derived an equation of the form of the Mori-Zwanzig equation (2.9) for this simple example: The term $1 - ax_t^2$ is the optimal prediction, i.e. the Markovian approximation using the relevant variables x_t. The sum $\sum_{j=1}^{t} c^{j-1} bx_{t-j}$ contains the memory terms depending on past values of x and the term $c^t y_0$ is the noise term with information about the irrelevant, or for us inaccessible, variable y. highlight eqn! ✓

We now apply the SINAR algorithm to data of a trajectory of the augmented Henon system and demonstrate the increase in performance by using memory terms compared to applying the usual Markovian SINDy algorithm.

We set as parameters $a = 1.3, b = 0.3, c = 0.4$ and initial values $x_0 = y_0 = 0$. Then, for

why 0.4 here but 0.3 in the fig. above?

example, the exact model parameters up to a memory depth of 3 in Equation (4.3) is

$$x_{t+1} = 1 - 1.3x_t^2 + 0.3x_{t-1} + 0.12x_{t-2} + 0.048x_{t-3} + \mathcal{O}(c^3). \tag{4.4}$$

As basis functions we choose

$$\Theta(\tilde{x}_t) = \begin{bmatrix} 1 \\ x_t^2 \\ x_t \\ \vdots \\ x_{t-p+1} \end{bmatrix}.$$

give a formula

We now generate a trajectory of length $T = 300$. We use the first 75% of the trajectory for the training in SINAR and the remainder for validation. In Figure 4 we see how the relative Euclidean validation error decreases for increasing memory depth p. Predicted were the evolutions of x, y and the full pair (x, y) with data about only the variables that were to be predicted. It is interesting to note how deep a memory depth is necessary to get an accurate prediction. The chaotic nature of the system, i.e. small deviations at one point in time causing significant deviations in the long term behaviour, yields that even coefficients of the form bc^j for $j \approx 25$ have to be taken into account and are recovered by SINAR. For the full system (x, y), the system is Markovian and the prediction error is very small even for $p = 1$.

✓

!

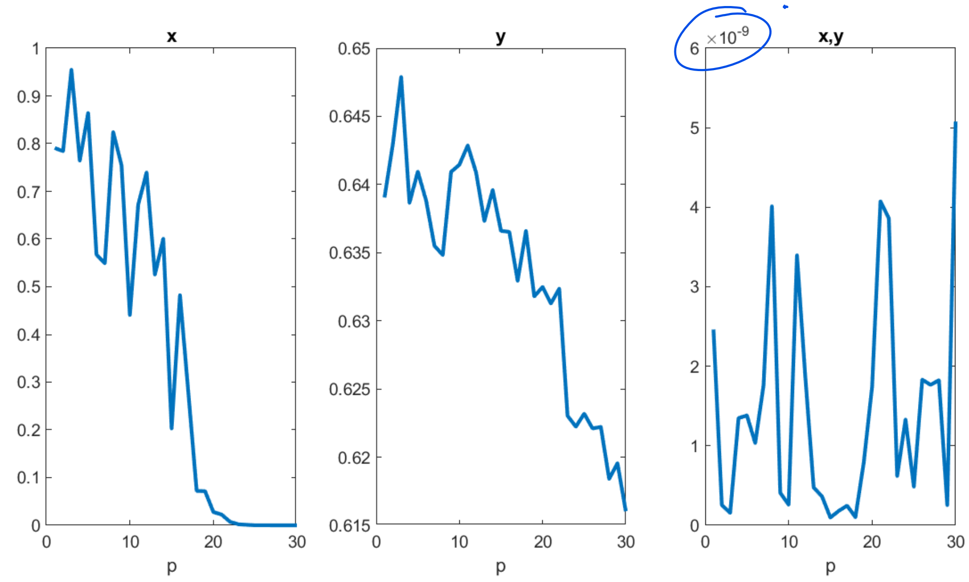


Figure 4: Relative error of prediction for SINAR on different (pairs of) visible variables of the augmented Henon system with different memory depths p on the x-axis. The prediction accuracy improves with increasing memory depth. In SINAR, we chose $\lambda = 0$. As parameters in the augmented Henon system, we chose $a = 1.3, b = 0.3, c = 0.4$.

- Could you make an error plot showing the Hausdorff-distance of the attractors of the true and the estimated problem in dependence of p?

5 Agent-based model for opinion dynamics

Let us now discuss the example of a network-based model of agents that change their opinions on a topic based on the opinions of their neighbors in the network. We use a time-discrete agent-based model (ABM) in [Mis12].
Suppose there are N agents and each agent has exactly one out of m different opinions, denoted by $1, \dots, m$. The vector X, which comes from

$$\mathbb{X} = \{1, \dots, m\}^N, \tag{5.1}$$

then represents the opinions of each agent. The neighborhoods of all agents are represented by the symmetric adjacency-matrix $A \in \{0,1\}^{N \times N}$ where $A_{ij} = 1$ means that agents i and j are neighbors to each other and $A_{ij} = 0$ otherwise. Let $N_i := \#(j : A_{ij} = 1)$ be the number of neighbors of an agent. The diagonal entries of A are set to 1, so that every agent is neighboring to itself.
Let the procedure of opinion change be given by the following rule:
In every time step, every agent picks one of his neighbors in the network uniformly at random and changes his opinion with probability $\alpha_{m'm''}$ where m' is the opinion of the agent and m'' is the opinion of the selected neighbour. This results in the term

$$\mathbb{P}[X_i^{t+1} = m'' | X_i^t = m'] = \alpha_{m'm''} \frac{\#(j : A_{ij} = 1 \text{ and } X_j^t = m'')}{N_i} \text{ for } m' \neq m''$$

which we denote by $p_i^t(m', m'')$.
The probability for an agent not to change his opinion thus is

$$\mathbb{P}[X_i^{t+1} = m' | X_i^t = m'] = 1 - \sum_{m'' \neq m'} p_i^t(m', m'').$$

We can now state the microdynamic defined by F. It is given by

$$F(X, \sigma)_i = X_{\sigma_i},$$

where at every time step σ_i is drawn uniformly at random from the set of neighbors of agent i, i.e. $\sigma_i \sim \mathcal{U}(j : A_{ij} = 1)$.
In algorithmic form, the agent-based model is executed in the following way:

you only consider the neighbor choosing, and not the accept/reject decision here!

Algorithm 1: Agent-based opinion change model

```
Choose end time $T$, number of agents $N$, network adjacency matrix $A$, opinion
  change coefficients $\alpha_{m',m''}$
for $t = 1, \dots, T$ do
    for $i = 1, \dots, N$ do
        Draw $j$ from $\{j : A_{ij} = 1\}$ uniformly at random
        With probability $\alpha_{X_i^t, X_j^t}$: $X_i^{t+1} = X_j^t$
    end
end
```

We now define as the *concentrations* of opinions the function

$$\xi(X) = \frac{1}{N} \begin{pmatrix} \#X_i = 1 \\ \vdots \\ \#X_i = m \end{pmatrix}.$$

It turns out that for a complete network, i.e. $A_{ij} = 1 \; \forall i, j$, we can derive a macrodynamic for the expected evolution of

$$x_t := \xi(X^t) \tag{5.2}$$

that does not include memory terms. It is given by

$$\mathbb{E}[(x_{t+1})_{m'} | x_t] = (x_t)_{m'} + \sum_{m'' \neq m'} (\alpha_{m''m'} - \alpha_{m'm''})(x_t)_{m''}(x_t)_{m'} \text{ for } m' = 1, \dots, m. \tag{5.3}$$

This equation can be derived as follows: In case of a complete network, $p_i^t(m', m'') \equiv p^t(m', m'')$ is independent of i because the concentrations of opinions among neighbors are equal for all agents since they all have the same neighbors. Then

$$p^t(m', m'') = \alpha_{m'm''}(x_t)_{m''}. \tag{5.4}$$

In every time step, every agent with opinion m' chooses its opinion in the next time step with respective probabilities $p^t(m', m'')$ for all opinions $m'' \neq m'$ and probability $1 - \sum_{m'' \neq m'} p^t(m', m'')$ for keeping opinion m'. Since the number of these agents is given by $N \cdot (x_t)_{m'}$, the expected absolute number of agents that change their opinion from

Can one see (5.3) direcly from Mori-Zwanzig in the full-network case?
(Give it a try, but don't pursue if it gets too complicated)

m' to m'' is given by

$$\begin{aligned}
&\mathbb{E}[\#\text{Agents changing opinion from } m' \text{ to } m'']\\
&= \sum_{i:X_i^t=m'} p^t(m',m'')\\
&= N\cdot (x_t)_{m'}\cdot p^t(m',m'')\\
&= N\cdot (x_t)_{m'}\cdot \alpha_{m'm''}\cdot (x_t)_{m''}.
\end{aligned} \tag{5.5}$$

This is the expected absolute number of agents that change their opinion from m' to m''. This means, that from this term alone, the concentration $(x_t)_{m'}$ of m' is reduced by $\frac{1}{N}$ times this term, which is $\alpha_{m'm''}(x_t)_{m''}(x_t)_{m'}$. Since at the same time agents with opinion m'' can change their opinion to m' with probability $\alpha_{m''m'}(x_t)_{m''}(x_t)_{m'}$, we have to subtract the analogous term for $\mathbb{E}[\#\text{Agents changing opinion from } m'' \text{ to } m']$ and the factor $(\alpha_{m''m'} - \alpha_{m'm''})$ comes in.
In consequence, for a complete network, the loss of information about X does not yield loss of information about the evolution of x. However, this is generally not the case for incomplete networks. In that case, the projection from the macro- to the microdynamic yields the application of the Mori-Zwanzig procedure from Section 2. An example that explicitly shows that probabilities for future states of the macrodynamic depend on past values is given in the Appendix.

5.1 Recovering the macrodynamic in case of an incomplete network

We now create realisations of the ABM with network that consists of two equally sized clusters of agents. Edges between agents from different clusters exist but are few. Inside the clusters, agents are not necessarily connected with each other, either. To this end, we create networks of an even number N of agents and divide them into two clusters of size $\frac{N}{2}$ each. Inside every cluster, for every pair of two agents, a link is installed with probability p_{inside}. Two agents from different clusters are connected with probability $p_{between}$. From a realisation $X^0 \dots, X^T$ we deduce the concentrations of opinions $[x_0,\dots,x_T] = [\xi(X^0),\dots,\xi(X^T)]$. We create multiple realisations of length upper/lower index? T that we denote by $\underline{\mathbf{X}}^1,\dots,\mathbf{X}^r$ with the same parameters and divide these data into training data $\underline{\mathbf{X}}_1,\dots,\mathbf{X}_{train}$ and validation data $\mathbf{X}_{train+1},\dots,\mathbf{X}_{\cancel{T}}^{r}$. ?Subsequently, we execute the SINAR method with different memory depths p on the training data. For this, the SINAR method can straightforwardly be modified for multiple trajectories by defining data matrices $\mathbf{X}' = [\mathbf{X}'_1,\dots,\mathbf{X}'_{train}]$ and $\tilde{\mathbf{X}} = [\tilde{\mathbf{X}}_1,\dots,\tilde{\mathbf{X}}_{train}]$ in the notation of Section 3. We then compute the reconstruction errors of the validation data for each value of $p = 1,\dots,p_{max}$. For the reconstruction, we divide each realisation of

M was #opinions in sec 2
here it is a validation time interval, if I get it right

the validation data into blocks of length M and create realisations of the NAR model determined with SINAR. We calculate the relative Euclidean distance between reconstruction and data for each block and take the mean over all those. As starting values of every reconstruction, we always use the last p values of the previous block so that for every value of p, predictions are made for the same time indices from the data.

i.e., exact initialization, right?

5.1.1 Case 1: A complete network

For $p_{inside} = p_{between} = 1$, the network is complete and there should be no improvement of the prediction by allowing memory terms.
We set $N = 5000, T = 300$ and $A_{ij} = 1 \ \forall i, j$. The number of different opinions is $m = 3$. As coefficients $\alpha_{m',m''}$ we choose

$$\begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} \end{pmatrix} = \begin{pmatrix} 0 & 0.275 & 0.05 \\ 0.05 & 0 & 0.275 \\ 0.275 & 0.05 & 0 \end{pmatrix} \tag{5.6}$$

As initial concentrations we assign values to the X_i^0 so that $\xi(X^0) = (0.45, 0.1, 0.45)^T$. Since the entries of $\xi(X^t)$ always sum up to 1, information about the concentrations of opinions 1 and 2 immediately yields the concentration of opinion 3 so that we use SINAR to find an NAR model for the evolution of $\xi(X)_1$ and $\xi(X)_2$ only and omit the redundant information $\xi(X)_3$. As the block length in the validation data, we use $M = 10$. We can already write down the macrodynamic since it is given in Equation (5.3) (see Appendix for details):

$$\begin{aligned} (x_{t+1})_1 &= (1 - \alpha_{31} - \alpha_{13})(x_t)_1 + (\alpha_{13} - \alpha_{31})(x_t)_1^2 + (\alpha_{21} - \alpha_{12} - \alpha_{31} + \alpha_{13})(x_t)_1(x_t)_2 \\ (x_{t+1})_2 &= (1 - \alpha_{32} - \alpha_{23})(x_t)_2 + (\alpha_{23} - \alpha_{32})(x_t)_2^2 + (\alpha_{12} - \alpha_{21} - \alpha_{32} + \alpha_{23})(x_t)_1(x_t)_2. \end{aligned} \tag{5.7}$$

Inspired by this structure, we choose as basis functions in SINAR

$$[\tilde{\varphi}_1, \ldots, \tilde{\varphi}_L](x_t) = [(x_t)_1, (x_t)_2, (x_t)_1^2, (x_t)_2^2, (x_t)_1(x_t)_2] \tag{5.8}$$

so that

$$\tilde{\Theta}(\tilde{x}_t) = [\underbrace{(x_t)_1, (x_t)_2, (x_t)_1^2, (x_t)_2^2, (x_t)_1(x_t)_2}_{\text{Markovian terms as in (5.7)}}, \underbrace{(x_{t-1})_1, (x_{t-1})_2, (x_{t-1})_1^2, (x_{t-1})_2^2, (x_{t-1})_1(x_{t-1})_2, \ldots}_{\text{Memory terms}}]^T. \tag{5.9}$$

We create $r = 20$ realisations of which we use 12 for training the the others for validation. We let the sparsity parameter $\lambda = 0$ and $\lambda = 0.05$ to test how the accuracy decreases with a sparser model. Since the macrodynamic (5.7) is Markovian, we obtain for the prediction error of the validation data no improvement by allowing memory terms (Figure 5) for neither the 10- nor the one-step prediction error.

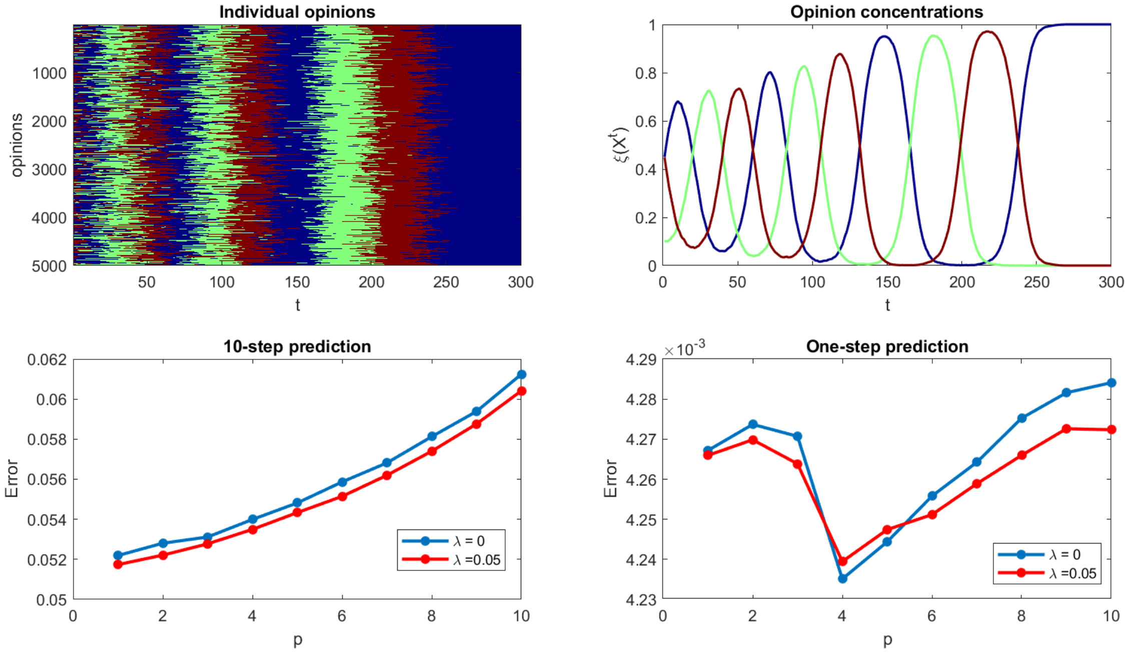


Figure 5: Top left: One realisation of the microdynamic. Every column of the graphic represents the opinion of each of the 5000 agents at one point in time. Blue color represents opinion 1, green represents opinion 2 and red represents opinion 3. Top right: Corresponding realisation of the macrodynamic $\xi(X)$ that represents the concentrations of opinions among all agents. We can observe oscillatory behaviour since agents with opinion 1 tend to change their opinion to 2 and analogously from 2 to 3 and from 3 to 1. Bottom: 10-step and one-step relative prediction errors of the NAR models determined by SINAR for different memory depths p with $\lambda = 0$ and $\lambda = 0.05$. As expected, the prediction error does not decrease with higher memory depth than $p = 1$.

5.1.2 Case 2: A two-cluster network

We now construct a network with $N = 5000$ agents, divided into two clusters of size 2500 each. Inside every cluster, $p_{inside} = 1$ so that all agents are connected with each other within a cluster. We set $p_{between} = 0.0001$. Again, $M = 3$ and $\alpha_{m',m''}$ are the same as in case 1. As the starting condition, we let opinions in the first cluster be distributed by $(0.8, 0.1, 0.1)$ and in the second cluster by $(0.1, 0.1, 0.8)$. If the initial concentrations in both clusters were equal then they would evolve in a quite similar way in parallel so that the macrodynamic would essentially be the same as in the complete network case. With the initial concentrations being so different, it is possible that an opinion that is dominant in one cluster at one point in time but only sparsely represented in the other, can become popular through the links between agents from different clusters.

This will cause the difference in behaviour of the evolution of concentrations compared to the complete network.

We create $r = 20$ realisations of length $T = 600$ and again use 12 for training, the remaining for validation. Memory terms become immediately significant, as the error graphs illustrate (Figure 6).

The non-sparse and sparse solutions only deviate slightly from each other in their accuracy, but the sparse solution gives a significantly more compact model. For example, for $p = 2$, we obtain for the coefficients $\tilde{\Xi}$

$$\begin{aligned} \lambda = 0 : \tilde{\Xi} &= \begin{bmatrix} 2.01 & 0.02 & -0.06 & -0.04 & 0.03 & -1.01 & -0.02 & 0.063 & 0.04 & -0.03 \\ -0.08 & 1.89 & 0.03 & 0.09 & 0.01 & 0.08 & -0.89 & -0.03 & -0.09 & -0.01 \end{bmatrix} \\ \lambda = 0.05 : \tilde{\Xi} &= \begin{bmatrix} 1.9463 & 0 & 0 & 0 & 0 & -0.9467 & 0 & 0 & 0 & 0 \\ 0 & 1.9531 & 0 & 0 & 0 & 0 & -0.9534 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{5.10}$$

so that for $\lambda = 0.05$ the NAR model is given by

$$\begin{aligned} (x_{t+1})_1 &= 1.9463(x_t)_1 - 0.9467(x_{t-1})_1 \\ (x_{t+1})_2 &= 1.9531(x_t)_2 - 0.9534(x_{t-1})_2. \end{aligned} \tag{5.11}$$

The system isn't ergodic right? (Fig 5 top right shows that it reached an absorbing state)

If we were to allow random opinion choice with a small probability (to destroy abs. states), could we give a stationary distribution of the system? (In an "expected" fashion, as the system itself is chosen randomly, to start with).

I ask this, because it would be neat to show an expected error plot.

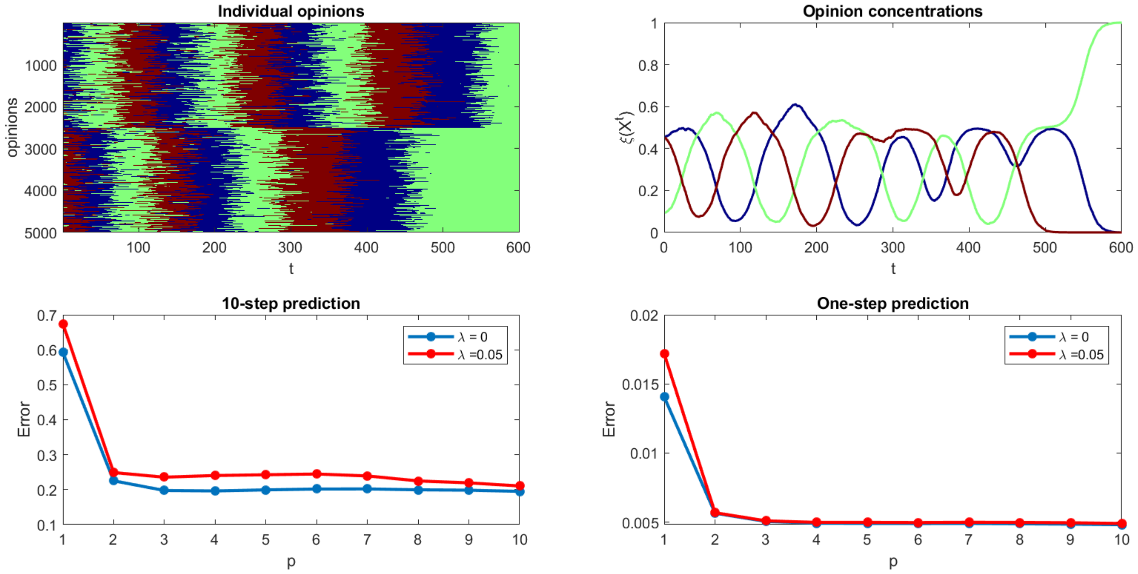


Figure 6: Top left: One realisation of the microdynamic. Every column of the graphic represents the opinion of each of the 5000 agents at one point in time. Blue color represents opinion 1, green represents opinion 2 and red represents opinion 3. Top right: Corresponding realisation of the macrodynamic $\xi(X)$. Again there is oscillatory behaviour but also where dominant opinions in one cluster are transported to the other cluster and become dominant there while becoming less dominant in the first cluster. This is the reason for plateaus and short dips as in the red graph at time 300 - 400. Bottom: 10-step and one-step relative prediction errors of the NAR models determined by SINAR for different memory depths p with $\lambda = 0$ and $\lambda = 0.05$. A memory depth of $p = 2$ yields a significant decrease in the prediction errors compared to Markovian predictions. For $p > 2$ the errors do not change much, however.

5.1.3 Case 3: A five-cluster network

We repeat the same procedure as with the two-cluster network but with five clusters of equal size 1000. Again, all agents within a cluster are connected with each other and $p_{between} = 0.0001$. The $\alpha_{m',m''}$ are identical to the ones used in the first two examples. As starting conditions we let opinions in the different clusters be drawn according to different distributions for each cluster. Those distributions are $(0.8, 0.1, 0.1), (0.1, 0.1, 0.8), (0.1, 0.8, 0.1), (0.3, 0.4, 0.3)$ and $(0.5, 0.3, 0.2)$.

Similar to when we used a two-cluster network, memory terms become important for predictions of the evolution of the microdynamic. In contrast to the example with the two-cluster network, performance does not stagnate for higher memory depths than 2 but converges to 0.2 for the 10-step predictions (Figure 7).

For $\lambda = 0.05$, we obtain the NAR model

$$\begin{aligned}(x_{t+1})_1 &= 1.7058(x_t)_1 - 0.1262(x_t)_1(x_t)_2 - 0.7042(x_{t-1})_1 + 0.12(x_{t-1})_1(x_{t-1})_2\\(x_{t+1})_2 &= 1.7479(x_t)_2 - 0.7479(x_{t-1})_2.\end{aligned} \tag{5.12}$$

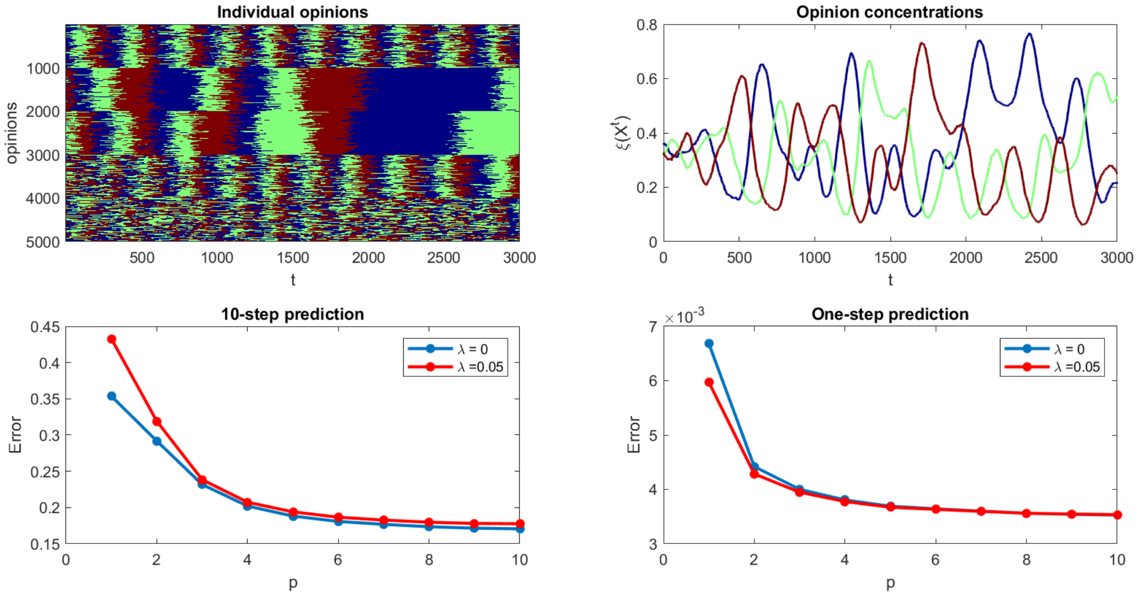


Figure 7: Top left: One realisation of the microdynamic. Every column of the graphic represents the opinion of each of the 5000 agents at one point in time. Top right: Corresponding realisation of the macrodynamic $\xi(X)$. The behaviour is much more complex than in the first two cases. Bottom: 10-step and one-step relative prediction errors of the NAR models determined by SINAR for different memory depths p with $\lambda = 0$ and $\lambda = 0.05$. Had a memory depth of $p = 2$ yielded approximately the same prediction accuracy as higher memory depths for a two-cluster network, the prediction error decreases here smoothly for higher values of p.

Which conclusions do you draw from these experiments?

A Appendix

A.1 Example for need of memory terms in case of an incomplete network

We will see in the following minimal example, that for an incomplete network, the knowledge of x_t is not sufficient to make the best possible prediction for $\mathbb{E}[x_t]$. Rather, the inclusion of x_{t-1} gives a different expected value for x_t.

Let us use $N = 3$ agents and two opinions that we denote by *black* and *white*. Assume, two of the three agents share the black opinions, phrasing it now as *two nodes are black*, so $x_t = (\frac{2}{3}, \frac{1}{3})^T$. The network forms a line, so that agents 1 and 3 are connected to agent 2 but not to each other. All $\alpha_{m'm''}$ are set to 1. Assuming a uniform distribution about the possible opinion vectors $X \in \mathbb{X}$ on the microscale, i.e. $\mathbb{P}[X^t = X|x_t] \equiv const$, the probability that $\mathbb{P}[x_{t+1} = (1,0)^T|x_t = (\frac{2}{3}, \frac{1}{3})^T]$ can be computed as follows:

Let us denote by $(\bullet - \circ - \bullet)$ the opinion vector in which agents 1 and 3 share the black opinion and agent 2 has the white opinion and analogously for different opinion vectors. Then by the law of total probability,

$$\begin{aligned}
&\mathbb{P}[x_{t+1}=(1,0)^T|x_t=(\frac{2}{3},\frac{1}{3})^T]=\\
&\underbrace{\mathbb{P}[x_{t+1}=(1,0)^T|X^t=(\bullet-\bullet-\circ)]\mathbb{P}[X^t=(\bullet-\bullet-\circ)|x_t=(\frac{2}{3},\frac{1}{3})^T]}_{(I)}\\
&+\underbrace{\mathbb{P}[x_{t+1}=(1,0)^T|X^t=(\bullet-\circ-\bullet)]\mathbb{P}[X^t=(\bullet-\circ-\bullet)|x_t=(\frac{2}{3},\frac{1}{3})^T]}_{(II)}\\
&+\underbrace{\mathbb{P}[x_{t+1}=(1,0)^T|X^t=(\circ-\bullet-\bullet)]\mathbb{P}[X^t=(\circ-\bullet-\bullet)|x_t=(\frac{2}{3},\frac{1}{3})^T]}_{(III)}\\
&=\underbrace{\mathbb{P}[x_{t+1}=(1,0)^T|X^t=(\circ-\bullet-\bullet)]\frac{2}{3}}_{(I)+(II)}\\
&+\underbrace{\mathbb{P}[x_{t+1}=(1,0)^T|X^t=(\bullet-\circ-\bullet)]\frac{1}{3}}_{(III)}\\
&=\frac{2}{3}\cdot\frac{1}{2}\cdot\frac{2}{3}\cdot 1+\frac{1}{3}\cdot\frac{1}{2}\cdot\frac{2}{3}\cdot\frac{1}{2}=\frac{4}{18}+\frac{1}{18}=\frac{5}{18}\approx\mathbf{0.2778}.
\end{aligned}$$

To clarify the last two steps, the factors of $\frac{2}{3}$ and $\frac{1}{3}$ come in because we assumed a uniform distribution of the opinion vectors given the opinion concentrations. Plus, the opinion vectors $(\bullet-\bullet-\circ)$ and $(\circ-\bullet-\bullet)$ are symmetric and hence yield the same transition probabilities for X^t to time step $t+1$. Remember that the probability for a node to change its opinion to, respectively, keep its opinion at, black is given by the number of its neighbors,including itself, with black opinion divided by its total number of neighbors.

Now, if we additionally have the information that $x_{t-1}=(\frac{2}{3},\frac{1}{3})^T$, then it holds that $\mathbb{P}[x_{t+1}=(1,0)^T|x_t=(\frac{2}{3},\frac{1}{3})^T]\neq\mathbb{P}[x_{t+1}=(1,0)^T|x_t=(\frac{2}{3},\frac{1}{3})^T,x_{t-1}=(\frac{2}{3},\frac{1}{3})^T]$, because:

$$\begin{aligned}
&\mathbb{P}[x_{t+1}=(1,0)^T|x_t=(\frac{2}{3},\frac{1}{3})^T,x_{t-1}=(\frac{2}{3},\frac{1}{3})^T]=\\
&\sum_{X,\tilde{X}\in\mathbb{X}}\mathbb{P}[x_{t+1}=(1,0)^T|X^t=X]\mathbb{P}[X^t=X|X^{t-1}=\tilde{X},x_t=(\frac{2}{3},\frac{1}{3})^T]\underbrace{\mathbb{P}[X^{t-1}=\tilde{X}|x_{t-1}=(\frac{2}{3},\frac{1}{3})^T]}_{=\frac{1}{3}\text{ by assumption}}
\end{aligned}$$

We find that the probabilities for X^t under X^{t-1}, assuming that exactly two nodes in X^t are black, are given as in the following table:

$\downarrow X^{t-1}, \rightarrow X^t$	$(\bullet - \bullet - \circ)$	$(\bullet - \circ - \bullet)$	$(\circ - \bullet - \bullet)$
$(\bullet - \bullet - \circ)$	2/3	1/3	0
$(\bullet - \circ - \bullet)$	2/5	1/5	2/5
$(\circ - \bullet - \bullet)$	0	1/3	2/3

By summing up the columns and dividing by 3 since all options for X^{t-1} are assumed to be equally probable, this gives that

$$\mathbb{P}[X^t = (\bullet - \bullet - \circ)|x_{t-1} = (\frac{2}{3}, \frac{1}{3})^T] = \frac{16}{45}$$
$$\mathbb{P}[X^t = (\bullet - \circ - \bullet)|x_{t-1} = (\frac{16}{45}, \frac{1}{3})^T] = \frac{13}{45}$$
$$\mathbb{P}[X^t = (\circ - \bullet - \bullet)|x_{t-1} = (\frac{2}{3}, \frac{1}{3})^T] = \frac{16}{45}$$

As above, when we only assumed knowledge about x_t, we need the probabilities $\mathbb{P}[x_{t+1} = (1,0)^T|X_t]$. These are given by

$$\mathbb{P}[x_{t+1} = (1,0)^T|X^t = (\bullet - \bullet - \circ)] = \frac{1}{3}$$
$$\mathbb{P}[x_{t+1} = (1,0)^T|X^t = (\bullet - \circ - \bullet)] = \frac{1}{6}$$
$$\mathbb{P}[x_{t+1} = (1,0)^T|X^t = (\circ - \bullet - \bullet)] = \frac{1}{3}$$

With this, we obtain

$$\mathbb{P}[x_{t+1} = (1,0)^T|x_t = (\frac{2}{3}, \frac{1}{3})^T, x_{t-1} = (\frac{2}{3}, \frac{1}{3})^T] = 2\cdot\frac{16}{45}\cdot\frac{1}{3}+\frac{13}{45}\cdot\frac{1}{6} = \frac{32}{135}+\frac{13}{220} = \frac{64+13}{280} \approx \mathbf{0.2821}.$$

This is unequal to the value of ≈ 0.2778 that we computed for $\mathbb{P}[x_{t+1} = (1,0)^T|x_t = (\frac{2}{3}, \frac{1}{3})^T]$.
The additional knowledge that $x_{t-1} = (\frac{2}{3}, \frac{1}{3})^T$ makes it slightly less probable that $X^t = (\bullet - \circ - \bullet)$ ($\frac{13}{45}$ instead of 1/3) which is the opinion vector that gives a lower probability that $X^{t+1} = (\bullet - \bullet - \bullet)$ in the next time step all nodes are black than that $X^{t+1} = (\circ - \bullet - \bullet)$ or $(\bullet - \bullet - \circ)$.

A.2 Derivation of Equation (5.7)

With $M = 3$ opinions, Equation (5.3) reads

$$(x_{t+1})_1 = (x_t)_1 + (\alpha_{21} - \alpha_{12})(x_t)_1(x_{t+1})_2 + (\alpha_{31} - \alpha_{13})(x_t)_1(x_t)_3$$
$$(x_{t+1})_2 = (x_t)_2 + (\alpha_{12} - \alpha_{21})(x_t)_1(x_{t+1})_2 + (\alpha_{32} - \alpha_{23})(x_t)_2(x_t)_3$$
$$(x_{t+1})_3 = (x_t)_3 + (\alpha_{13} - \alpha_{31})(x_t)_1(x_{t+1})_3 + (\alpha_{23} - \alpha_{32})(x_t)_2(x_t)_3.$$

Using $(x_t)_3 = 1 - (x_t)_1 - (x_t)_2$, we get

$$\begin{aligned}
(x_{t+1})_1 &= (x_t)_1 + (\alpha_{21} - \alpha_{12})(x_t)_1(x_t)_2 + (\alpha_{31} - \alpha_{13})(x_t)_1(1 - (x_t)_1 - (x_t)_2) \\
(x_{t+1})_2 &= (x_t)_2 + (\alpha_{12} - \alpha_{21})(x_t)_1(x_t)_2 + (\alpha_{32} - \alpha_{23})(x_t)_2(1 - (x_t)_1 - (x_t)_2).
\end{aligned}$$

Rearranging gives

$$\begin{aligned}
(x_{t+1})_1 &= (1 - \alpha_{31} - \alpha_{13})(x_t)_1 + (\alpha_{13} - \alpha_{31})(x_t)_1^2 + (\alpha_{21} - \alpha_{12} - \alpha_{31} + \alpha_{13})(x_t)_1(x_t)_2 \\
(x_{t+1})_2 &= (1 - \alpha_{32} - \alpha_{23})(x_t)_2 + (\alpha_{23} - \alpha_{32})(x_t)_2^2 + (\alpha_{12} - \alpha_{21} - \alpha_{32} + \alpha_{23})(x_t)_1(x_t)_2.
\end{aligned}$$

This is Equation (5.7).

References

[ADP14] Ken Aho, DeWayne Derryberry, and Teri Peterson. Model selection for ecologists: The worldviews of aic and bic. *Ecology*, 95:631–6, 03 2014.

[AM17] Hassan Arbabi and Igor Mezic. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM J. Appl. Dyn. Syst., 16(4), 2096–2126*, 2017.

[BBP$^+$] Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, Eurika Kaiser, and J. Nathan Kutz. Chaos as an intermittently forced linear system. *Nature Communications volume 8, Article number: 19 (2017).*

[BD91] Peter J. Brockwell and Richard A. Davis. *Time Series: Theory and Methods.* Springer, 1991.

[BPKa] Steven L. Brunton, Joshua Proctor L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems.

[BPKb] Steven L. Brunton, Joshua Proctor L. Proctor, and J. Nathan Kutz. Sparse identification of nonlinear dynamics with control(sindyc). *IFAC-PapersOnLine Volume 49, Issue 18, 2016, Pages 710-715.*

[CHK02] Alexandre J. Chorin, Ole H. Hald, and Raz Kupferman. Optimal prediction with memory. *Physica D 166 (2002) 239–257*, pages 1487–1503, 2002.

[Hé76] M. Hénon. A two-dimensional mapping with a strange attractor. *Comm. Math. Phys.*, 50(1):69–77, 1976.

[HO93] Per Hansen and Dianne O'leary. The use of the l-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.*, 14:1487–1503, 11 1993.

[KK08] Sadanori Konishi and Genshiro Kitagawa. *Information Criteria and Statistical Modeling.* Springer, 2008.

[KKB18] Eurika Kaiser, J. Nathan Kutz, and Steven L. Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. A 474: 20180335*, 2018.

[Koo31] B. O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

[LL19] Kevin K. Lin and Fei Lu. Data-driven model reduction, wiener projections, and the mori-zwanzig formalism. *arXiv:1908.07725v1*, 2019.

[Mis12] Arvind Kumar Misra. A simple mathematical model for the spread of two political parties. *Nonlinear Analysis: Modelling and Control, 2012, Vol. 17, No. 3, 343–354*, 2012.

[riMM19] Nelida Črnjarić Žic, Senka Maćešić, and Igor Mezić. Koopman operator spectrum for random dynamical systems. *Journal of Nonlinear Science*, 2019.

[Tib] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological) vol. 58, no. 1, 1996, pp. 267–288.*

[TRL$^+$] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics, 2014, 1 (2) : 391-421.*

[WKR] M.O. Williams, I.G. Kevrekidis, and C.W. Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *J Nonlinear Sci 25, 1307–1346 (2015).*

[Zwa01] Robert Zwanzig. *Nonequilibrium Statistical Mechanics.* Oxford University Press, 2001.