Memory-based reduced modelling and data-based estimation of opinion spreading

Niklas Wulkow\(^1\), Péter Koltai\(^1\), and Christof Schütte\(^{1,2}\)

\(^1\)Department of Mathematics and Computer Science, Freie Universität
Berlin, Germany

\(^2\)Zuse Institute Berlin, Germany

June 16, 2020

Abstract

We investigate opinion dynamics based on an agent-based model, and are interested in predicting the evolution of the percentages of the entire agent population that share an opinion. Since these opinion percentages can be seen as an aggregated observation of the full system state, the individual opinions of each agent, we view this in the framework of the Mori–Zwanzig projection formalism. More specifically, we show how to estimate a nonlinear autoregressive model (NAR) with memory from data given by a time series of opinion percentages, and discuss its prediction capacities for various specific topologies of the agent interaction network. We demonstrate that the inclusion of memory terms significantly improves the prediction quality on examples with different network topologies.

Keywords— Memory-based model, Sparse model identification, Mori-Zwanzig formalism, Nonlinear autoregressive model, Opinion dynamics, Agent-based model

1 Introduction

Political opinion polls capture how the opinions of people within a society regarding a certain topic or their current voting preferences are distributed. Individual opinions do not have to be constant but rather are subject to change induced by impactful events or the opinions of their peers which is formalized under the term conformity in [Sta15]. There have been recent advances in simulating the process in which members of a society change their opinions, see, e.g., [BLA11, KLT07, Mis12, LBW\(^+\)12, NKB08, BG12, BCN17] and the review articles
This is in part due to increasing computing power which enables to carry out agent-based models that simulate behaviour of members of a synthetic population, such as members of a society, on the microscale by emulating the decision-making rules. The agents are often treated as the nodes of a network while an edge between two nodes means that these agents are neighbors of each other and thus influence each other’s respective opinions.

One is often not interested in modelling, or predicting, which person has which opinion but rather, as in polls, what the percentage of each opinion within the society is. There is ample interest in deriving dynamics for the evolution of these percentages.

In this article, we will present a framework which identifies the governing equations for the dynamics of opinion percentages for different types of networks, more exactly, how the governing equations can be inferred from data on the opinion percentages. To this end, we will emulate the decision making process with a simple agent-based model (ABM) that is based on the assumption of conformity and inspired by the ABM in [Mis12]. Introductions into agent-based modelling in general can be found in [JSW98] and [LJMR09] and specifically into agent-based models for opinion dynamics in [Ban16]. We investigate the cases of complete and incomplete interaction networks. More precisely, we will consider complete networks where every agent interacts with all others, as well as incomplete networks in which there are subcommunities within the society that have few links between each other. As we will show, in the case of a complete network, one can identify a Markovian model for the dynamics of the opinion percentages using standard well-mixedness arguments known from the mean-field approaches or population limits, e.g., for predator-prey models [Ber92]. However, arguments used for that case do not hold up in cases when the network is not complete. We will show how to use information from the past (memory) via a kind of delay embedding of the dynamics to describe the evolution of opinion percentages in the general case.

The exact reason for the inclusion of memory will formally be derived in Section 2 by using the Mori–Zwanzig formalism [Zwa01, LL19, CHK02]. Inspired by problems in statistical physics, the Mori–Zwanzig formalism explains how in the case of only low-dimensional observations of a high-dimensional system being available, the evolution of these observations of the full system can be obtained by replacing the missing information of the full system by past information of these available observations. This is in light of the result of Takens [Tak81] that states that, under fairly generic assumptions, the delay embedding of the dynamics of an observable is diffeomorphic to the dynamics of the full system.

There are various techniques for the modelling of time-discrete dynamical systems which involve the memory of the system. An intuitive approach is comprised by Higher-order Markov Models [Raf85, Tuy18]. These models are defined by transition probabilities between discrete states where each state represents a sequence of cells of a discretization of the state space with a given length. Although these models can be powerful in investigating the long-term behaviour of the process by means of Markov State Models for Markovian processes [BPN14], they yield two problems: The loss of accuracy obtained from the discretization and an exponentially increasing number of states with increasing length of the sequences and
number of grid cells.

Another example is simplex projection as in [SM90] where, using Takens’ result, subsequent states of a system are predicted from relative next steps of similar patterns as its recent history. A younger modelling technique is Long short-term memory neural networks (LSTMs) [HS97, PD18] which is a subclass of recurrent neural networks and specifically designed for prediction of time series for which past information is vital. However, both these techniques provide little to no understanding the dynamical rules of the system: Simplex projection does not produce any model or dynamical law but rather uses a procedure similar to the nearest neighbors classification algorithm (see, e.g., [DGL13]). LSTMs, as most neural networks, typically have far too many parameters to admit interpretability. An additional means for forecasting of memory-dependent dynamical systems is the well-known class of autoregressive (AR) models [BD91], which describes the evolution of a system by a linear combination of its most recent states. Additionally, there exist variants of these AR models that are sparse [DZZ12, FSG+07] or nonlinear [Bil13] or comprise both aspects in application to a Singular Value Decomposition of a data matrix [BBP+16]. As we will see, linear (Markovian) systems cannot describe the evolution of opinion percentages even in the simplest case, but simple polynomial terms are sufficient for fully connected networks. We shall address this point with nonlinear AR (NAR) models, as derived through the Mori–Zwanzig formalism.

The novelty in our work lies in the usage of such NAR models, estimated from data, to describe the evolution of opinion percentages. Theoretical justification comes from the Mori–Zwanzig formalism applied to the special (clustered) network topologies that we consider. Even though the models that we derive merely predict an expected evolution of a randomly evolving quantity, large-deviation type results suggest that in the large-population limit the random fluctuations vanish [BDM14, WNWS20]. We will show that the prediction accuracy of the NAR models for the opinion percentages increases with larger memory depths. With this, we will deploy methods from data-driven (sparse) system identification—as in Dynamic Mode Decomposition [SS08, TRL+14, JSN13] or Sparse Identification of Nonlinear Dynamics [BPK16a]—to the field of opinion dynamics.

In Section 2 we start with recalling the derivation of NAR models for the evolution of observations through the Mori–Zwanzig formalism. In Section 3 we will present a method that detects coefficients of functions in these NAR models. This method will be demonstrated on a toy example derived from the Hénon map in Section 4, and used to increase the accuracy of prediction of opinion percentages in the case of incomplete networks in Section 5.

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

Below, we will model the spread of opinions inside a closed society by an agent-based model. It will consist of a high number $N$ of agents who change their opinions $X_i$, $i = 1, \ldots, N$, within a finite set of $M$ possible opinions over discrete time steps according to a rule that
is based on the opinions of themselves and other agents. This rule will be Markovian, or memory-free, i.e., the changes of opinions are only influenced by opinions in the current time step. These dynamics will be called the microdynamics. The state of the microdynamics at time \( t \) is denoted by \( X_t = ((X_t)_1, \ldots, (X_t)_N)^T \). The respective state space is denoted by \( \mathcal{X} \), and has cardinality \(|\mathcal{X}| = M^N\).

We will only be able to observe the percentages of opinions, i.e., the ratios of those among all agents for each of the \( M \) opinions. In this article, we are interested in identifying the dynamical rules of the evolution of the percentages of opinions, which we call the macrodynamics. Identifying the dynamics of low-dimensional observations of a higher dimensional system is a typical setup for the Mori–Zwanzig formalism \([Zwa01, CHK02, LL19]\). We will consider a general framework for this and show how it yields a nonlinear autoregressive model \([Bil13]\) for the macrodynamics. Later on we show how it can be applied to the specific case of the spread of opinions.

### 2.1 The setting: Microdynamics and observations

First we assume that the microdynamics is Markovian (memory-free) and deterministic. We consider the dynamical system \( F : \mathcal{X} \to \mathcal{X} \) that governs the microdynamics

\[
X_{t+1} = F(X_t) \in \mathcal{X}. \tag{2.1}
\]

Further, we denote the space of observations of the microdynamics (observables) by \( \mathcal{Y} \subseteq \mathbb{R}^m \) and the set of functions that map states of the dynamical system (2.1) to \( \mathcal{Y} \) by

\[
\mathcal{G} := \{ g : \mathcal{X} \to \mathcal{Y} \}.
\]

Let us denote the accessible, or relevant, variables by \( x = \xi(X) \in \mathcal{Y} \). Hence, \( \xi \in \mathcal{G} \) is a fixed observable. We suppose from here on that we do not have knowledge of the state of the microdynamics at any point in time but instead have the value of the function \( \xi \). Additionally, we define the subspace \( \mathcal{H} \) of functions in \( \mathcal{G} \) that depend only on these relevant variables and map to \( \mathcal{Y} \) as

\[
\mathcal{H} := \{ h \in \mathcal{G} \mid \exists \tilde{h} \in \tilde{\mathcal{H}} : h = \tilde{h} \circ \xi \}, \text{ where } \tilde{\mathcal{H}} = \{ \tilde{h} : \xi(\mathcal{X}) \to \mathcal{Y} \}.
\]

Functions in \( \mathcal{H} \) still depend on \( X \in \mathcal{X} \) but the information of \( \xi(X) \) is enough to evaluate them. When we write \( \tilde{h}(x) \) for \( x \in \mathcal{Y} \), we actually mean \( \tilde{h}(x) \) where \( \tilde{h} \) is the function in \( \tilde{\mathcal{H}} \) with the property that \( h = \tilde{h} \circ \xi \). An example is

\[
\mathcal{X} = \mathbb{R}^2, \quad \xi(X) = X_1 + X_2, \quad h(X_1, X_2) = (X_1 + X_2)^2 = \xi(X)^2.
\]

In this case it is enough to know the value of \( \xi(X) \) to evaluate \( h(X) \).
2.2 Approximating observations with projections

The goal is now to represent the evolution of a function from $G$ under the microdynamics with knowledge only about $\xi$ but not of the state of the microdynamics $X_t$. As illustrated in the following diagram, instead of taking one step of the microdynamics and then evaluating a function $g \in G$, we only have access to the observation $\xi(X)$ and want to evaluate $g(F(X))$ under the premise that $\xi(X) = x$.

\[ X \xrightarrow{F} F(X) \]
\[ \xi \downarrow \quad g \downarrow \]
\[ \xi(X) = x \xrightarrow{?} g(F(X)) \]

This means, we want to install a dynamical system that is dependent only on $\xi(X_t)$ but approximates all observations of the microdynamics as well as possible.

A simple example would be:

\[ X = \mathbb{R}^2, \quad \xi(X) = X_1 + X_2, \quad g(X) = X_1^2 + X_2. \]

If we only know $X_1 + X_2$, how do we compute $X_1^2 + X_2$? In contrast to $h$ in the above example, $g \notin H$.

If an exact representation is not possible, we define an approximation that depends only on $\xi(X)$. To this end, we define a projection operator $P : G \to H$ that maps a function depending on $X$ to a function depending on $\xi(X)$. We additionally define its complement $Q := Id - P$. We assume from now on that the microdynamics are stationary with an $F$-invariant probability distribution $\mu$ over $X$, so that when asking what $g(X)$ is, we assume that $X$ is distributed by $\mu$. Given this, a natural candidate for $P$ would be the conditional expectation with respect to $\mu$,

\[ (Pg)(x) = \mathbb{E}[g(X) \mid \xi(X) = x], \quad (2.3) \]

which is more precisely defined in Appendix [A.4].

This term represents exactly the question we should ask: What do we expect $g(X)$ to be if we know that $\xi(X) = x$? However, the integrals in (2.3) are often infeasible if we do not know $\mu$. We hence follow [LL19] until the end of Section [2.3] and define $P$ as the orthogonal projection onto the span of a set of linearly independent functions from $H$. This set consists of the columns of $\varphi = [\varphi_1, \ldots, \varphi_L]$:

\[ (Pg)(x) := \varphi(x) \langle \varphi, \varphi \rangle^{-1} \langle \varphi, g \rangle \quad (2.4) \]

where $x \in Y$ and the scalar product $\langle \cdot, \cdot \rangle$ is defined for matrix-valued functions $f : X \to \mathbb{R}^{m \times a}$ and $g : X \to \mathbb{R}^{m \times b}$ as

\[ (f, g) := \int_X f(X)^T g(X) \, d\mu(X) \in \mathbb{R}^{a \times b}, \]
which itself is matrix-valued. The term \( \langle \varphi, \varphi \rangle \) is a mass matrix that ensures that \( P \) is in fact an orthogonal projection. This orthogonal projection has the property that \( Pg \) is the closest function in \( \text{span}(\varphi) \) to \( g \) where closeness is measured by the scalar product, i.e., \( Pg \) minimizes
\[
\langle g - Pg, g - Pg \rangle = \int_X (g - Pg)^T (X)(g - Pg)(X) d\mu(X),
\]
which is the expected quadratic difference between \( g \) and \( Pg \).

Note that if \( \mathcal{H} \) is infinite-dimensional, one would need an infinite number of functions to yield that \( \text{span}(\varphi) = \mathcal{H} \). In this case the projection formalism is well defined if \( \mathcal{H} \) is closed. In practice, in this case for the computation that will follow one would choose a sufficiently rich finite set of functions so that \( \text{span}(\varphi) \) covers those parts of \( \mathcal{H} \) that are of interest.

### 2.3 Mori–Zwanzig representation of the macrodynamics

We will now show how to represent the evolution of a function in \( \mathcal{H} \) over time. With the Koopman operator \([\text{Koo31}]\) \( \mathcal{K} \) 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
\[
\mathcal{K}^{t+1} = \sum_{k=0}^t \mathcal{K}^{t-k} PK(QK)^k + (QK)^{k+1}.
\]  

(2.5)

The Dyson formula describes a way to iteratively split up the application of the Koopman operator to a function \( g \) into parts \( PKg \) and \( QKg \). Equation (2.5) yields, by application of both sides of the equation to \( \xi \) and evaluation at the initial value \( X_0 \) of the macrodynamics, that
\[
(\mathcal{K}^{t+1}\xi)(X_0) = \sum_{k=0}^t \mathcal{K}^{t-k}[PK(QK)^k\xi](X_0) + (QK)^{t+1}\xi(X_0)
\]
which results in: \( \xi(X_{t+1}) = \sum_{k=0}^t [PK(QK)^k\xi](X_{t-k}) + (QK)^{t+1}\xi(X_0) \)

(2.6)

Setting \( \rho^k := (QK)^k\xi \) yields: \( \xi(X_{t+1}) = \sum_{k=0}^t [PK\rho^k](X_{t-k}) + \rho^{t+1}(X_0) \)
\[
= \sum_{k=0}^t [P(\rho^k \circ F)](X_{t-k}) + \rho^{t+1}(X_0).
\]

We can replace \( X_{t-k} \) by \( x_{t-k} \) in the last 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 \( PK\rho^k \) and its equivalent formulations to indicate that \( P \) is a projection operator that works on the function \( K\rho^k \).

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 of \( \xi(X_{t+1}) \) that only uses \( \xi(X_t) \). The sum in the last row of (2.6) 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\footnote{It accumulates unobserved effects as witnessed by the complement projector $Q$. Note that it is expected to decay fast, if the system mixes strongly (in the sense that $\mathcal{K}$ has a small spectral radius on the set of functions perpendicular to the constant function, which in turn is assumed to lie in the range of $P$). In this sense, the term ‘noise’ refers to negligible correlation to variables $x_{t-k}$ that contribute strongly to $\xi(X_{t+1})$.}. In total, the last row of (2.6) is called the Mori–Zwanzig equation.

Substituting the definition of $P$ as the orthogonal projection onto basis functions as in (2.4), we obtain

$$P(\rho^k \circ F)(x_{t-k}) = \varphi(x_{t-k})\langle \varphi, \varphi \rangle^{-1} \langle \varphi, \rho^k \circ F \rangle$$

$$= \varphi(x_{t-k})\langle \varphi, \varphi \rangle^{-1} \int X \varphi(\xi(X))^T \rho^k(F(X)) d\mu(X) =: \varphi(x_{t-k})h_k \in \mathbb{R}^m \quad (2.7)$$

with vector-valued coefficients $h_k = \langle \varphi, \varphi \rangle^{-1} \int X \varphi(\xi(X))^T \rho^k(F(X)) d\mu(X)$.

Finding a suitable approximation of the non-accessible noise term $\rho^{t+1}(X_0)$ in (2.6) is generally a non-trivial task and depends on properties of the microdynamics. Examples are discussed in [LC17, HEVEDB10, KDG15]. From this point onwards, we will make the simplification of replacing $\rho^{t+1}(X_0)$ by a zero-mean stochastic noise term $\varepsilon_{t+1} \in \mathbb{R}^m$. A typical practice is to let $\varepsilon_{t+1}$ be a zero-mean Gaussian random variable as, e.g., in [LL19, LBL16]. With this, we obtain the macrodynamics

$$x_{t+1} = \sum_{k=0}^{t} \varphi(x_{t-k})h_k + \varepsilon_{t+1}. \quad (2.8)$$

In the form of the diagram (2.2), through the Mori–Zwanzig formalism, we get

$$X \xrightarrow{F^t} F^t(X) \xrightarrow{\xi} \xi(X) = x \xrightarrow{2.6} \xi(F^t(X)) \quad (2.9)$$

2.4 Macrodynamics as a nonlinear autoregressive process

If it is reasonable to assume a sufficiently fast decay of the terms $h_k$ with increasing $k$, the memory terms that lie far in the past have negligible influence [HHSN07, VVR17, CHK00, ZDV18]. In light of (2.6) and (2.7) it is sufficient that the $\rho^k$ decay fast. To understand when this is the case, we recall $\rho^k = (Q\mathcal{K})^k\xi$, and assume the range($P$) $\approx \mathcal{H}$, i.e., functions parametrized by $\xi$ are well approximated by the chosen approximation space. Then, $\rho^k$ decays fast if $Q\mathcal{K}$ has a small norm, which is the case if $F$ mixes well functions that are perpendicular to $\mathcal{H}$. In other words, the dominant modes of $\mathcal{K}$ should align well with the space $\mathcal{H}$. For
quantitative statements we refer to \[ZDV18\].

Thus, in order to obtain a feasible number of memory terms, from now one we approximate
the dynamics by ending the sum in (2.8) with \( k = p - 1 \) instead of \( k = t \), i.e., by truncating
the terms \( \varphi(x_{t-p})h_p, \ldots, \varphi(x_0)h_t \). Regarding the selection of an appropriate value for the
memory depth \( p \) there are various methods such as Information Criteria \[KK08\] or \[ADP14\] or
the L-curve method \[HO93\]. We have thus derived a nonlinear autoregressive model (NAR)
over \( x \).

As a nonlinear autoregressive model with memory depth \( p \) we define a dynamical system
of the form \[Bil13,AH96\]

\[ x_{t+1} = f(x_t, \ldots, x_{t-p+1}) + \varepsilon_{t+1}. \] (2.10)

A common structure for \( f \) is

\[ x_{t+1} = H_0 f(x_t) + \cdots + H_{p-1} f(x_{t-p+1}) + \varepsilon_{t+1}. \]

with matrix-valued coefficients \( H_i \) and a vector-valued function \( f \).

A special case is the well-known family of linear auto-regressive models \[BD91\], for which
\( f \) is linear,

\[ x_{t+1} = H_0 x_t + \cdots + H_{p-1} x_{t-p+1} + \varepsilon_{t+1}. \]

Note that Equation (2.8) does not have the form of NAR models as (2.10), since the coeffi-
cients \( h_k \) are vector-valued and the basis functions \( \varphi \) are matrix-valued, opposed to having
matrix-valued coefficients and vector-valued basis functions (Figure 1). It turns out that

\[ x_{t+1} = \varphi(x_t) h_0 + \varphi(x_{t-1}) h_1 + \cdots \]

\[ x_{t+1} = H_0 \varphi(x_t) + H_1 \varphi(x_{t-1}) + \cdots \]

Figure 1: Top: Form of the nonlinear autoregressive model (2.8). Bottom: Form of
classical nonlinear autoregressive models (2.10).

(2.10) is in fact a special case of the formulation we have derived here because by choosing
scalar-valued functions $\tilde{\varphi}_1, \ldots, \tilde{\varphi}_L$ and defining

$$\varphi(x) = \begin{pmatrix}
\tilde{\varphi}_1(x) & \ldots & \tilde{\varphi}_L(x) & 0 & \ldots & 0 \\
0 & \ldots & \tilde{\varphi}_1(x) & \ldots & \tilde{\varphi}_L(x) & 0 & \ldots & 0 \\
\vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \ldots & 0 & \tilde{\varphi}_1(x) & \ldots & \tilde{\varphi}_L(x) & 0 & \ldots & 0
\end{pmatrix} \in \mathbb{R}^{m \times mL} \quad (2.11)$$

we find that

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

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

Since in Section 3 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 \[\text{WKR14,TRL}^{\text{+14}}, \text{Extended Dynamic Mode Decomposition WKR14}\] or Sparse Identification of Nonlinear Dynamics \[\text{BPK16a,BPK16b}\], we use the formulation of (2.8) which reads

$$x_{t+1} = \sum_{k=0}^{p-1} H_k \tilde{\varphi}(x_{t-k}) + \epsilon_{t+1}. \quad (2.12)$$

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

### 2.5 Stochastic microdynamics

Let us consider stochastic dynamics

$$X_{t+1} = F(X_t, \omega_t)$$

where $\omega_t \in \Omega$ is a random influence on $F$ which is now defined as $F : \mathbb{R}^n \times \Omega \rightarrow \mathbb{R}^n$. We will assume that the noise process $\omega_t$, $t \in \mathbb{N}$, is i.i.d. with law $\mathbb{P}$. In this case we only strive to forecast the expected macrodynamics, and define the (stochastic) Koopman operator as

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

The spaces $\mathcal{G}$ and $\mathcal{H}$, just as the projection $P$ remain unchanged. Naturally, to the derivation of the Mori–Zwanzig approximation we need to apply the necessary obvious modifications. E.g., the last step in (2.6) now has to be modified as:

$$[PK\rho^k](x_{t-k}) = \varphi(x_{t-k}) \langle \varphi, \varphi \rangle^{-1} \int_\Omega \int_X \varphi(\xi(X))^T \rho_k(F(X, \omega))d\mu(X)d\mathbb{P}(\omega).$$

We can thus obtain the identical structure of the macrodynamics as in (2.8) where for the computation of the coefficients $h_k$ in (2.7) the expectation with respect to $\mathbb{P}$ had to be added.
3 Sparse Identification of Nonlinear Autoregressive Models (SINAR)

We propose here a method of data-based identification for coefficients $H_k$ in (2.8) that is an extension of the Sparse Identification of Nonlinear Dynamics (SINDy) algorithm from [BPK16a, BPK16b, KKB18]. SINDy can be used to identify the governing equations of a Markovian—in our case, discrete time—dynamical system

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

from data

$$X = \begin{bmatrix} x_0 & \ldots & x_{T-1} \\ \vdots & \ddots & \vdots \\ x_{T-1} \end{bmatrix}, X' = \begin{bmatrix} x_1 & \ldots & x_T \\ \vdots & \ddots & \vdots \\ x_T \end{bmatrix}, X, X' \in \mathbb{R}^{m \times T}.$$

We will extend this method to Non-Markovian systems by applying SINDy to an extended version of $X$, the Hankel matrix

$$\tilde{X} = \begin{bmatrix} x_{p-1} & \ldots & x_{T-1} \\ \vdots & \ddots & \vdots \\ x_0 & \ldots & x_{T-p} \end{bmatrix}.$$

In essence, this is the concept used for the Hankel-alternative view of Koopman (HAVOK) analysis from [BBP+16], which identifies a linear autoregressive model on transformed coordinates obtained from a Singular Value Decomposition of the Hankel matrix to separate linear from non-linear, or even chaotic, behaviour of a Markovian system. In this section and by the choice of the name SINAR, we explicitly want to point out the connection of system identification methods for non-linear Markovian systems to their counterparts for non-linear Non-Markovian systems (with finite memory these are NAR systems) that can be derived through the Mori–Zwanzig formalism from Section 2.

3.1 SINDy: A short summary

We start with a short description of SINDy: In SINDy we try to approximate each coordinate of $f$ by a linear combination of basis functions $\theta_i : \mathbb{R}^m \rightarrow \mathbb{R}$ and define

$$\Theta(x) = \begin{bmatrix} \theta_1(x) \\ \vdots \\ \theta_v(x) \end{bmatrix}, \quad \Theta(X) = \begin{bmatrix} \theta_1(x_0) & \ldots & \theta_1(x_{T-1}) \\ \vdots & \ddots & \vdots \\ \theta_v(x_0) & \ldots & \theta_v(x_{T-1}) \end{bmatrix},$$
To this end, we fit a coefficient matrix $\Xi \in \mathbb{R}^{m \times v}$ to the data $X, X'$ by finding

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

so that

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

In order to only obtain the basis functions from $\Theta$ that are dominant for the relation between $x_{t+1}$ and $\Theta(x_t)$, we enforce a sparsity constraint using the LASSO regression algorithm \cite{Tib96} in which a regularisation term is added onto the coefficient matrix. Note that (3.2) decomposes to independent minimization problems for the rows of $\Xi$. For every row $X'_i$ of $X'$, we solve

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

(3.4)

The use of the 1-norm generates 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 (3.3) is a better interpretability of the model since only the dominant terms have been identified as influential to the dynamics. It is vital to set $\lambda$ so that the loss of accuracy is minimal compared to the gain in interpretability.

SINDy is related to the (first step of) the method of Dynamic Mode Decomposition (DMD) \cite{WKR14,TRL+14}, which aims at finding a linear connection between $x_t$ and $x_{t+1}$. To this end, one solves

$$A = \arg \min_{A} \| X' - AX \|_F.$$  

(3.5)

In a second step, DMD then uses $\Xi$ from (3.4) to uncover properties of the Koopman operator of the system. SINDy, instead, tries to explain the evolution of $x_t$ by basis functions that do not have to be linear. Still, essentially, the problem (3.5) is equivalent to (3.4) for $\Theta(x) = x$ and $\lambda = 0$. Further, there exists a sparse version of DMD \cite{JSN13}, where the sparsity constraint is enforced by the additive 1-norm regularisation as in (3.4). Then the emerging minimization problem is the same as (3.4) with $\Theta(x) = x$.

### 3.2 Extending SINDy to SINAR

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 (2.12) with sparse coefficients, i.e., with only
few basis functions with non-zero coefficients: Selecting a memory depth $p$ and denoting

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

let us define as data matrices the Hankel matrix

$$
\tilde{X} = \begin{bmatrix}
    x_p \ldots x_{T-1} \\
    \vdots \\
    x_0 \ldots x_{T-p}
\end{bmatrix} = \begin{bmatrix}
    \tilde{x}_{p-1} \ldots \tilde{x}_{T-1}
\end{bmatrix} \in \mathbb{R}^{mp \times (T-p)}
$$

and $X' = \begin{bmatrix}
    x_p \ldots x_T
\end{bmatrix} \in \mathbb{R}^{m \times (T-p)}.
$$

(3.6)

Again we choose basis functions

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

for example

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

and minimize for every row of $X'$

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

(3.7)

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

$$
x_{t+1} = \tilde{\Xi} \tilde{\Theta}(\tilde{x}_t) \in \mathbb{R}^m
$$

$$
\Leftrightarrow (x_{t+1})_i = \sum_{j=1}^v \tilde{\Xi}_{ij} \tilde{\theta}_j(\tilde{x}_t).
$$

(3.8)
By deleting all columns of ˜Ξ 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 (3.8). 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.

By choosing
\[
\tilde{\Theta}(\tilde{x}_t) = \begin{bmatrix}
\tilde{\varphi}_1(x_t) \\
\vdots \\
\tilde{\varphi}_L(x_t) \\
\vdots \\
\tilde{\varphi}_1(x_{t-p+1}) \\
\vdots \\
\tilde{\varphi}_L(x_{t-p+1})
\end{bmatrix}
\]
with \( \tilde{\varphi}_1, \ldots, \tilde{\varphi}_L \) being scalar-valued functions as in Equation (2.11) in Section 2, we can directly fit the coefficients \( H_k \) of the model (2.12) which was derived through the Mori–Zwanzig formalism previously. Then ˜Ξ has the blockwise form
\[
\tilde{\Xi} = [H_0, \ldots, H_{p-1}] \in \mathbb{R}^{m \times pL}
\]
and
\[
\tilde{\Xi} \tilde{\Theta}(\tilde{x}_t) = \sum_{k=0}^{p-1} H_k \begin{bmatrix} \tilde{\varphi}_1(x_{t-k}) \\ \vdots \\ \tilde{\varphi}_L(x_{t-k}) \end{bmatrix}.
\]

Of course, by choosing linear basis functions \( \Theta(\tilde{x}_t) = \tilde{x}_t \) and setting \( \lambda = 0 \), one obtains a well-known linear autoregressive model [BD91]. Except for the sparsity term, the determination of model coefficients as in (3.7) is exactly the Least Squares method commonly used for the linear AR models.

The covariance of the noise term \( \varepsilon_{t+1} \) in (2.12) can be estimated in the common way for linear or nonlinear AR models [BD91, LL19] by calculating the statistical covariance between \( \mathbf{X}' \) and \( \tilde{\Xi} \tilde{\Theta}(\mathbf{X}) \) (see Appendix A.6 for more details on both statements).

The following diagram sketches how system identification methods from different contexts are related. With DMD, SINDy, SINAR and AR models in mind, one can observe that in all of them, a minimization problem of the same form is solved: Given are data matrices \( \mathbf{X} \) and \( \mathbf{X}' \) which contain data points of the realisation of a, possibly memory-exhibiting, dynamical system that are shifted from each other by one time step. Then one tries to find a connection between both through a transformation of \( \mathbf{X} \) which is multiplied with a coefficient matrix by solving (omitting possible sparsity constraints)
\[
\Xi = \arg\min_{\Xi} \| \mathbf{X}' - \Xi \Theta(\mathbf{X}) \|_F
\]
Figure 2: Sketch of the SINDy algorithm (left) and SINAR (right). SINAR contains the additional step of creating a Hankel matrix.

In DMD, one tries to find a linear, Markovian, connection between $x_t$ and $x_{t+1}$, i.e. $\Theta(x) = x$. In SINDy, $X$ is transformed in a possibly nonlinear way in order to explain the evolution of systems for which a linear model might be inaccurate. Linear AR models look for a linear connection between a fixed number of past values of the system and its next value. The columns of $X$, in this case, contain not just data points of the system but sequences of data points of a fixed length. In this way, AR models can be seen as a form of delay-embedded DMD. SINAR brings together SINDy and AR models in the sense that one seeks a possibly nonlinear connection between past values of the system and subsequent ones.

4 Example: An 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 by means of an example of a dynamical system
and use SINAR to detect an NAR model that reconstructs the dynamics.

The classical Henon system \cite{Hen76} describes a two-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. The dynamical system is given by

\[
x_{t+1} = 1 - ax_t^2 + y_t \\
y_{t+1} = bx_t,
\]

where \(a, b\) are fixed parameters. As we can observe, \(y_t\) is nothing more than a scaled and time-delayed version of \(x_t\). We now consider \(x\) as the relevant and \(y\) as the irrelevant variable; this means in the Mori–Zwanzig formalism the space \(\mathcal{H}\) is given by all functions depending on only \(x\). We can then 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 \(x_{t+1}\):

\[
x_{t+1} = 1 - ax_t^2 + bx_{t-1}.
\]

Let us now consider an extended version of the Henon system

\[
x_{t+1} = 1 - ax_t^2 + y_t \\
y_{t+1} = bx_t + cy_t
\]  

(4.1)

whose dynamical behaviour is visualized in Figure 3.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.png}
\caption{Trajectory of length 5000 of the two-dimensional extended Henon system (4.1) with \(a = 1.3, b = 0.3, c = 0.3\) and initial values \(x_0 = y_0 = 0\). The first 1000 states are omitted here so that the trajectory has time to converge towards the attractor.}
\end{figure}

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 with an infinite one:

\[ x_{t+1} = 1 - ax_t^2 + bx_{t-1} + cy_{t-1} \]
\[ = 1 - ax_t^2 + bx_{t-1} + cbx_{t-2} + c^2y_{t-2} \]
\[ = 1 - ax_t^2 + bx_{t-1} + cbx_{t-2} + c^2b_{t-3} + c^3y_{t-3} \]
\[ = 1 - ax_t^2 + \sum_{j=1}^{t} c^{j-1}bx_{t-j} + c^{t+1}y_0, \]

which can be quickly shown by induction on \( t \).

We have hereby derived an equation of the form of the Mori–Zwanzig equation (2.6) 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 \( y \).

### 4.1 Reconstructing the extended Henon system with SINAR

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

We set as parameters \( a = 1.3, b = 0.3, c = 0.3 \) and initial values \( x_0 = y_0 = 0 \). Then, for example, the exact model up to a memory depth of 3 in Equation (4.2) is

\[ x_{t+1} = 1 - 1.3x_t^2 + 0.3x_{t-1} + 0.09x_{t-2} + 0.027x_{t-3} + \mathcal{O}(c^3). \]

As basis functions we choose monomials of the time-delayed coordinates up to second order without mixed terms between different delays,

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

### Short-term predictions

We now generate a trajectory of length \( T = 2000 \) out of which we erase the first 1000 steps to give the trajectory time to converge to the attractor. We then use the first \( T_{\text{train}} \) data points for training and the remaining \( 1000 - T_{\text{train}} \) for validation. With the training data, we
determine coefficients $\tilde{\Xi}$ for the basis functions in $\hat{\Theta}$ with SINAR for different memory depths $p$ and compute reconstructions $\hat{x}_{T_{\text{train}}+1}, \ldots, \hat{x}_{1000}$ of $x_{T_{\text{train}}+1}, \ldots, x_{1000}$ using Equation (3.8) with initial values $x_{T_{\text{train}}-p+1}, \ldots, x_{T_{\text{train}}}$. In essence, we recover the coefficients of the forms $a$ resp. $c^j b$ from Equation (4.2) until $j = p - 1$ and recompute values of the extended Henon system with the recovered coefficients. As error measure we use the relative Euclidean prediction error

$$\text{err}(\hat{X}') = \frac{\|X' - \hat{X}'\|_F}{\|X'\|_F}$$

(4.3)

where $X' = [x_{T_{\text{train}}+1}, \ldots, x_{1000}]$ denotes data points from the original trajectory and $\hat{X}' = [\hat{x}_{T_{\text{train}}+1}, \ldots, x_{1000}]$ data points from the reconstructed trajectory.

Although all coefficients are recovered up to an error of smaller than $10^{-14}$ when we use 800 time steps for training, the reconstruction becomes inaccurate after around 100 time steps which underlines the strongly chaotic nature of the system, i.e., small deviations at one point in time causing significant deviations in the long term behaviour. We thus use 920 time steps for training and only 80 time steps for validation to investigate how the relative Euclidean reconstruction error depends on the memory depth. Below we discuss how the attractor of the system is recovered using much longer reconstructions.

We see in Figure 4 how the relative Euclidean prediction error decreases for increasing memory depth $p$. Predicted was the evolution of $x$ with data about $x$. It is interesting to note how large a memory depth is necessary to get an accurate prediction for $x$ when $c = 0.3$ (Figure 4 (left)). The chaotic nature of the system yields that even coefficients of the form $bc^j$ for $j = 27$ have to be taken into account. Of course, for smaller $c$ such as $c = 0.03$, memory terms in (4.2) decay quicker and a memory depth of $p = 8$ is sufficient to yield an accurate prediction as can be seen in Figure 4 (right). For the full system $(x, y)$, the system is Markovian and the prediction error is unsurprisingly very small even for $p = 1$.

![Figure 4: Relative error of validation $\text{err}(\hat{X}')$ for SINAR on visible variable $x$ of the extended Henon system for two different values of $c$ with different memory depths $p$ on the x-axes. The prediction accuracy improves with increasing memory depth. Results based on SINAR with $\lambda = 0$. As parameters in the extended Henon system, we chose $a = 1.3, b = 0.3$ and $c = 0.3$ (left) resp. $c = 0.03$ (right). For every value of $p$, the same 80 time steps were taken into account for the reconstruction error.](image-url)
Attractor reconstruction

Although large deviations between original and reconstructed trajectories of $x_t$ occur after around 100 time steps, both trajectories remain on roughly the same set of points. We quantify this by the Hausdorff distance between the two-dimensional delay embeddings (see definition in Appendix A.7) of the original trajectory and each reconstructed trajectory. The Hausdorff distance denotes the maximal minimal distance of members of one set of points to another set. In other words, the Hausdorff distance between two sets is 0 if the sets are equal and big if there is a point in one set which is far away from all points in the other set.

We make predictions of 3000 time steps based on coefficients that were obtained with SINAR on data of 1000 time steps. In Figure 6 are depicted the two-dimensional delay embeddings of the original trajectory of $x$ and the reconstructed trajectories for $p = 1, 2, 5, 10$ and $p = 30$. There we see how already for $p = 2$ the original and reconstructed attractors look much more similar compared to $p = 1$. Figure 5 shows the Hausdorff distances for different memory depths. Similar to the relative Euclidean prediction error, the distance decreases with increasing $p$. The remaining error is due to the fact that the complicated geometry of the attractor is hard to approximate uniformly well with a finite set of points.

Figure 5: Hausdorff distances between original and reconstructed attractors with 3000 points of two-dimensional delay embeddings of $x$ for two different values of $c$ with different memory depths $p$ on the x-axes. Results based on SINAR with $\lambda = 0$ with parameters in the extended Henon system $a = 1.3, b = 0.3$ and $c = 0.3$ (left) resp. $c = 0.03$ (right).

\[ c = 0.3 \]

\[ c = 0.03 \]

\[ \text{Figure 5: Hausdorff distances between original and reconstructed attractors with 3000 points of two-dimensional delay embeddings of } x \text{ for two different values of } c \text{ with different memory depths } p \text{ on the x-axes. Results based on SINAR with } \lambda = 0 \text{ with parameters in the extended Henon system } a = 1.3, b = 0.3 \text{ and } c = 0.3 \text{ (left) resp. } c = 0.03 \text{ (right).} \]

\[ 2^{\text{Coverage of a two-dimensional object of diameter 2 by 3000 points results in a mesh size } \approx 2/\sqrt{3000} \approx 0.03. \text{ This is the same order of magnitude as the error we observe.}} \]
Figure 6: Two-dimensional delay embedded attractors with 3000 points of the extended Henon system with $a = 1.3, b = 0.3, c = 0.3$. Original (upper left) and reconstructed ones based on SINAR with $\lambda = 0$. For $p \geq 2$, differences are difficult to see but exist as the Hausdorff distances in Figure 5 indicate.

5 Application to an 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. Suppose, we can only observe the percentages of agents inside the network that share each opinion but not which agent exactly has which opinion, as in an anonymous opinion poll. Describing the evolution of these percentages is the identical setting of the Mori–Zwanzig formalism that we discussed in Section 2, since they are simply observations of hidden microdynamics. We will demonstrate the efficacy of NAR models in predicting the evolution of opinion percentages, compared with Markovian models. We use a time-discrete agent-based model (ABM), similar to the concept of modelling opinion changes in a population explained in [Mis12]. The ABM in [Mis12], however, is theoretically time-continuous. To apply the Mori–Zwanzig formalism to a time-continuous microdynamics we refer the interested reader to literature such as [CHK00, CHK02].
5.1 Formulating the ABM

The ABM is given as follows: Suppose there are $N$ agents and each agent has exactly one out of $M$ different opinions, denoted by $1,\ldots,M$. The vector $X_t$, which comes from $X = \{1,\ldots,M\}^N$, then represents the opinions of each agent at time $t$ and $(X_t)_i$ denotes the opinion of agent $i$ at time $t$. 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 of 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 its own neighbour.

Let the procedure of opinion changing be given by the following rule: In every time step, every agent picks one of its neighbors in the network uniformly at random and changes its opinion with adaption 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

$$P[(X_{t+1})_i = m'' | (X_t)_i = m'] = \alpha_{m'm''} \frac{\#(j : A_{ij} = 1 \text{ and } (X_t)_j = m'')}{N_i}$$

for $m' \neq m''$, which we denote by $p_t^i(m',m'')$. The probability for an agent not to change its opinion thus is

$$p_t^i(m',m') = P[(X_{t+1})_i = m' | (X_t)_i = m'] = 1 - \sum_{m'' \neq m'} p_t^i(m',m'').$$

In algorithmic form, the agent-based model is executed in the following way:

**Algorithm 1:** Agent-based opinion change model

1. Choose end time $T$, number of agents $N$, network adjacency matrix $A$, opinion change coefficients $\alpha_{m'm''}$, initial opinions $X_0$
2. for $t = 0,\ldots,T$ do
3. \hspace{1em} for $i = 1,\ldots,N$ do
4. \hspace{2em} Draw $j$ from $\{j : A_{ij} = 1\}$ uniformly at random (Choose neighbour)
5. \hspace{2em} Draw $u_i \sim U[0,1]$\n6. \hspace{2em} If $u_i < \alpha_{(X_t)_i,(X_t)_j}$; $(X_{t+1})_i = (X_t)_j$ (Adapt neighbour’s opinion)
7. \hspace{1em} end
8. end

To clarify the notation, remember that $(X_t)_i$ and $(X_t)_j$ denote the opinions of agents $i$ and $j$ at time $t$. Hence $\alpha_{(X_t)_i,(X_t)_j}$ is the adaption probability of opinion $(X_t)_j$ given that an agent has opinion $(X_t)_i$.

We can now state the so-defined microdynamics by

$$X_{t+1} = F(X_t,\omega_t)$$

where at every time step, $\omega_t$ denotes a tuple consisting of $N$ agents that represents the chosen
neighbour of each agent plus numbers \( u_i \sim U[0, 1] \) that govern the adaption probability \( \alpha_{(X_t)_i,(X_t)_j} \) as in Algorithm 1. To be more precise, \( \omega_t \) has the form

\[
\omega_t = (j_1, \ldots, j_N, u_1, \ldots, u_N), \quad j_i \sim U\{j : A_{ij} = 1\}, \quad u_i \sim U[0, 1].
\]

\( F \) then is given by

\[
(X_{t+1})_i = F(X_t, \omega_t)_i = \begin{cases} 
(X_1)_{ji} & \text{if } u_i < \alpha_{(X_t)_i,(X_t)_j} \\
(X_t)_i & \text{otherwise.}
\end{cases}
\]

This way of stating the microdynamics seems complicated compared to the more intuitive option of denoting by \( (\omega_t)_i \) the new opinion of the \( i \)-th agent, distributed by \( [p_{t_1}((X_t)_i, 1), \ldots, p_{t_M}((X_t)_i, M)] \). However, this would mean that the distribution of \( \omega_t \) would change over time, since the \( p_{t_i} \) depend on \( (X_t)_i \). For the Mori–Zwanzig formalism, this would prevent us from applying the procedure of skew-shift systems introduced in Section 2.5 where we drew all \( \omega_t \) a priori and thus independently of the \( X_t \). By using the notation of \( \omega_t \) denoting a tuple of neighbors \( j_i \) and random numbers \( u_i \) that are compared to the adaption coefficients, we can draw the whole sequence of \( \omega_t \) independently of the \( X_t \) and maintain consistency with the notation of skew-shift systems.

5.2 Deducing macrodynamics from the ABM

We now define as the opinion percentages the function

\[
\xi(X) = \frac{1}{N} \left( \begin{array}{c}
\#X_1 = 1 \\
\vdots \\
\#X_i = M
\end{array} \right)
\]

and are interested in modelling how these percentages evolve over time. It turns out that for a complete network, i.e., \( A_{ij} = 1 \forall i, j \), we can derive macrodynamics for the expected evolution of

\[
x_t := \xi(X_t),
\]

that do not require memory terms. They are given by

\[
\mathbb{E}[(x_{t+1})_{m'}|x_t] = (x_t)_{m'} + \sum_{m'' \neq m'} (\alpha_{m'm''} - \alpha_{m''m'}) (x_t)_{m''} (x_t)_{m'} \text{ for } m' = 1, \ldots, m. \tag{5.1}
\]

This equation can be derived as follows: In case of a complete network, \( p_{t_i}^i(m', m'') \equiv p_t^i(m', m'') \) is independent of \( i \) because the percentages of opinions among neighbors are equal for all agents since they all have the same neighbors. Then

\[
p_t^i(m', m'') = \alpha_{m'm''}(x_t)_{m''}.
\]
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 $m'$ to $m''$ is given by

$$E[\#\text{Agents changing opinion from } m' \text{ to } m''] = \sum_{i : (x_t)_i = m'} p_t(m', m'') = N \cdot (x_t)_{m'} \cdot \alpha_{m'm''} \cdot (x_t)_{m''}.$$  

This is the expected absolute number of agents that change their opinion from $m'$ to $m''$. This means, that from this term alone, the percentage $(x_t)_{m'}$ 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 $E[\#\text{Agents changing opinion from } m'' \text{ to } m']$ 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 $x$.

In order to align this with the Mori–Zwanzig formalism from Section 2, this means that

$$PK\xi = \mathcal{K}\xi,$$

thus $QK\xi = 0$,

$$PK\rho^k \quad \text{with } \rho^k = (QK)^k \xi$$

occur. This yields for $k > 0$ that $\rho^k = (QK)^{k-1} (QK) \xi$ where terms of the form

$$PK\rho^k$$

do not vanish. In this way we can see that memory terms are not required for the dynamics of $\xi$ if the network is complete.

However, this is generally not the case for incomplete networks. An example that explicitly shows that probabilities for future states of the macrodynamics depend on past values is given in Appendix A.1. In this case, it is no longer sufficient to know $\xi(X)$ instead of $X$ to state $\mathcal{K}\xi = E[\xi(F)]$. In other words, Equation (5.2) is no longer valid so that the $\rho^k$ do not vanish. In this case, by using as $P$ the orthogonal projection onto basis functions we were able to find approximate representations of the terms $P(\rho^k \circ F)$ in Equation (2.6).

With this we derived an NAR model. Here lies another part of the value of the application of the Mori–Zwanzig formalism: It installs that the structure of the ensuing macrodynamics in Equation (2.6) is additive, i.e., it can be written as a sum of transformations of individual memory terms as opposed to, e.g., containing products of memory terms. In this way, we can restrict the search for a good model of the macrodynamics to dynamics of this form. For an incomplete network, that is still sufficiently densely connected, we expect the microdynamics to be in expectation still close to that of a complete network. Thus, in such a case we expect
\[ QK\xi \approx 0, \text{ even if (5.2) does not hold exactly. Consequently, assuming dense connectedness, } \]
\[ \text{the opinion percentages should allow for a description of their evolution with a small memory depth.} \]

In the following, we will use SINAR to identify NAR models of this form suggested by the Mori–Zwanzig formalism.

### 5.3 Recovering the macrodynamics in case of an incomplete network

We now create realisations of the ABM with networks that consist of equally sized clusters of agents. Edges between agents from different clusters exist but are few. Inside the clusters, all agents are connected with each other. To this end, we create networks of an even number \( N \) of agents and divide them into clusters of equal size. Two agents from different clusters are connected with probability \( p_{\text{between}} \).

From the same initial state and with the same parameters, we create multiple realisations of the form \([X_0, \ldots, X_T]\) of the ABM and deduce the percentages of opinions \([x_0, \ldots, x_T]\) = \([\xi(X_0), \ldots, \xi(X_T)]\). We denote the realisations of the resulting macrodynamics by \(X_1, \ldots, X_r\) and divide these data into training data \(X_{1, \ldots, X_{\text{train}}}\) and validation data \(X_{\text{train}+1, \ldots, X_r}\).

Subsequently, we execute the SINAR method with different memory depths \(p\) on the training data. SINAR gives us NAR models that we use for the reconstruction of the validation data. For this, the SINAR method can straightforwardly be modified for multiple trajectories by defining data matrices \(X' = [X'_1, \ldots, X'_{\text{train}}]\) and \(\tilde{X} = [\tilde{X}_1, \ldots, \tilde{X}_{\text{train}}]\) in the notation of Section 3. We then compute the reconstruction errors of the validation data for each value of \(p = 1, \ldots, p_{\text{max}}\).

For the reconstruction, we divide each realisation \(X_i\) of the validation data into blocks of length \(l \geq p\). A block denotes \(l\) states \(x_i^{(j)} = [x_{ji}, \ldots, x_{(j+1)i-1}]\) while the next block will be \(x_i^{(j+1)} = [x_{(j+1)i}, \ldots, x_{(j+2)i-1}]\). We then compute a reconstruction \(\hat{x}_i^{(j)} = [\hat{x}_{ji}, \ldots, \hat{x}_{(j+1)i-1}]\) of this block with the NAR model obtained with SINAR for which we use the last \(p\) values of the previous block as starting values. As in Equation (4.3), we calculate the relative Euclidean error between reconstruction and data for each block by

\[
\text{err}(\hat{x}_i^{(j)}) = \frac{\|x_i^{(j)} - \hat{x}_i^{(j)}\|_F}{\|x_i^{(j)}\|_F}.
\]

Afterwards, we take the mean over all \(\text{err}(\hat{x}_i^{(j)})\) to measure the performance of the NAR model.

Since the entries of \(\xi(X_i)\) always sum up to 1, information about the percentages of opinions 1, \ldots, \(M-1\) immediately yields the percentage of opinion \(M\) so that we use SINAR to find an NAR model for the evolution of the percentages of the first \(M-1\) opinions only and omit the redundant information \(\xi(X)_M\). For the reconstruction error, we compare data about the percentages of only the first \(M-1\) opinions with their reconstructions. This NAR model does not necessarily ensure that the predicted first \(M-1\) percentages stay between 0 and 1 and their sum is at most 1. Since we make short-term predictions only, however, there
will at most be only slight deviations from this property.

In the form of the diagrams (2.2) and (2.9) from Section 2, the Mori–Zwanzig procedure applied to this concept can be described as

\[
\begin{align*}
\text{Opinions at time } t & \quad (X_t, \omega_t) \xrightarrow{F \text{ in ARM}} X_{t+1} \\
\text{Opinion concentrations at times } t - p + 1 \text{ to } t & \\
\xi & \quad \mathbb{E}[x_{t+1}] \\
\tilde{x}_t & \quad \text{Expected opinion concentrations at time } t + 1
\end{align*}
\]

**Case 1: A complete network**

For \( p_{between} = 1 \), 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

\[
\begin{pmatrix}
\alpha_{11} & \alpha_{12} & \alpha_{13} \\
\alpha_{21} & \alpha_{22} & \alpha_{23} \\
\alpha_{31} & \alpha_{32} & \alpha_{33}
\end{pmatrix} =
\begin{pmatrix}
0 & 0.165 & 0.03 \\
0.03 & 0 & 0.165 \\
0.165 & 0.03 & 0
\end{pmatrix}
\]

As initial percentages we assign values to the \((X_0)_i\) so that \( \xi(X_0) = (0.45, 0.1, 0.45)^T \).

As the block length in the validation data, we use \( l = 40 \). We can already write down the macrodynamics since they are given in Equation (5.1) (see Appendix A.2 for details):

\[
\begin{align*}
\mathbb{E}[(x_{t+1})_1| x_t] &= (1 + \alpha_{31} - \alpha_{13})(x_t)_1 + (\alpha_{13} - \alpha_{31})(x_t)_1^2 + (\alpha_{21} - \alpha_{12} - \alpha_{31} + \alpha_{13})(x_t)_1(x_t)_2 \\
&= 1.135(x_t)_1 - 0.135(x_t)_2^2 - 0.27(x_t)_1(x_t)_2, \\
\mathbb{E}[(x_{t+1})_2| x_t] &= (1 + \alpha_{32} - \alpha_{23})(x_t)_2 + (\alpha_{23} - \alpha_{32})(x_t)_2^2 + (\alpha_{12} - \alpha_{21} - \alpha_{23} + \alpha_{23})(x_t)_1(x_t)_2 \\
&= 0.865(x_t)_2 + 0.135(x_t)_2^2 + 0.27(x_t)_1(x_t)_2.
\end{align*}
\]

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

\[
[\tilde{\varphi}_1, \ldots, \tilde{\varphi}_L](x_t) = [(x_t)_1, (x_t)_2, (x_t)_1^2, (x_t)_2^1, (x_t)_1(x_t)_2]
\]

so that

\[
\tilde{\Theta}(\tilde{x}_t) = \left[(x_t)_1, (x_t)_2, (x_t)_1^2, (x_t)_2^1, (x_t)_1(x_t)_2, \ldots \right]^{\text{Markovian terms as in } (5.3)}
\]

\[
\left[(x_{t-1})_1, (x_{t-1})_2, (x_{t-1})_1^2, (x_{t-1})_2^1, (x_{t-1})_1(x_{t-1})_2, \ldots \right]^{T} \quad \text{Memory terms}
\]

Since (5.1), resp. (5.3), describe the expected evolution of the percentages and are thus in...
form of deterministic models, we omit the noise term $\varepsilon_{t+1}$ from Equation (2.12) which we assumed to satisfy $E[\varepsilon_{t+1}] = 0$.

We create $r = 20$ realisations of which we use 12 for training and the others for validation. We set the sparsity parameter to $\lambda = 0$ and to $\lambda = 0.05$ to test how the accuracy decreases with a sparser model. Since the macrodynamics (5.3) are Markovian, we obtain for the prediction error of the validation data no improvement by allowing memory terms (Figure 7) for neither the 40- nor the one-step prediction error. Note that the predictions with the sparse NAR model provide slightly better accuracy for large memory depths. This is, because small non-zero coefficients for memory terms improve the fit of the training data but cause errors in the prediction of the validation data, because the macrodynamics are Markovian. Through the sparsity constraint enforced, these non-zero coefficients for memory terms are cut off. The recovered sparse macrodynamics $p = 1$ reads

$$
(x_{t+1})_1 = 1.1353(x_t)_1 - 0.1351(x_t)_1^2 - 0.2709(x_t)_1(x_t)_2,
$$
$$
(x_{t+1})_2 = 0.8655(x_t)_2 + 0.1344(x_t)_2^2 + 0.2699(x_t)_1(x_t)_2,
$$

which is very close to the analytically derived macrodynamics (5.3).

Figure 7: Top left: One realisation of the microdynamics. Every column of the graphic represents the opinion of each of the 5000 agents at one point in time. Blue denotes opinion 1, green denotes opinion 2 and red denotes opinion 3. Top right: Corresponding realisation of the macrodynamics $\xi(X)$ that represent the percentages 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: 40-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 $p = 1$. 

25
Case 2: A two-cluster network

We now construct a network with $N = 5000$ agents, divided into two clusters of size 2500 each. We set $p_{\text{between}} = 0.0001$. Again, $M = 3$ and $\alpha_{m'nr'}$ 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 percentages in both clusters were equal then the percentages in both clusters would evolve in a quite similar way in parallel so that the macrodynamics would essentially be the same as in the complete network case. With the initial percentages 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 percentages compared to the complete network.

Moreover, in order to derive the Markovian macrodynamics in Equation (5.1), we needed that the probabilities for an agent $i$ to change its opinion $(X_t)_i$ at time $t$, which we denoted by $p_{t_i}((X_t)_i, m'')$, be independent of $i$. If the neighborhoods of different agents are generally different from each other, this is not longer the case. Especially so, if agents are distributed into different clusters, where opinion percentages might be very different. Thus, we cannot derive Markovian macrodynamics for this case but, in light of the Mori–Zwanzig formalism, we will need memory terms.

To show this, we create $r = 20$ realisations of length $T = 500$ and again use 12 for training, the remaining for validation. As block length, we choose $l = 20$. Memory terms become immediately significant, as the error graphs illustrate (Figure 8).

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 $\tilde{\Xi}$

$$
\lambda = 0 : \tilde{\Xi} = \begin{bmatrix}
2.04 & 0.03 & -0.07 & -0.08 & 0.02 & -1.05 & -0.02 & 0.07 & 0.07 & -0.02 \\
-0.05 & 1.88 & 0.00 & 0.11 & 0.06 & 0.06 & -0.89 & -0.01 & -0.12 & -0.05
\end{bmatrix}
$$

$$
\lambda = 0.05 : \tilde{\Xi} = \begin{bmatrix}
1.9691 & 0 & 0 & 0 & -0.9700 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.9662 & 0 & 0 & 0 & 0 & -0.9671 & 0 & 0 & 0
\end{bmatrix}
$$

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

$$(x_{t+1})_1 = 1.9691(x_t)_1 - 0.9700(x_{t-1})_1$$

$$(x_{t+1})_2 = 1.9662(x_t)_2 - 0.9671(x_{t-1})_2.$$

For $p = 1$, the NAR model obtained with SINAR ($\lambda = 0.05$) is

$$(x_{t+1})_1 = 1.0094(x_t)_1 - 0.053(x_t)_1(x_t)_2$$

$$(x_{t+1})_2 = 0.9894(x_t)_2 + 0.0574(x_t)_1(x_t)_2$$

With $\lambda = 0$, the obtained NAR model has other terms with non-zero coefficients, but these are small. In Figure 9, an example for the predictions of opinion percentages in one block...
using the NAR models with \( p = 1, 2 \) and 10 is depicted and compared to the corresponding data. As the error graphs in Figure 8 show already, the predicted percentages come closer to the percentages in the data with increasing memory depth.

Figure 8: Top left: One realisation of the microdynamics. Colours represent opinions as in Figure 7. Top right: Corresponding realisation of the macrodynamics \( \xi(X) \). Again there is oscillatory behaviour but also plateaus and short dips as in the red and green graphs at time 25 - 150. This is because at these times one opinion is dominant in one cluster but not present in the other. Through the links between the clusters, an opinion, that is not present in a cluster but dominant in the other one can be revived, e.g., the blue opinion in the upper cluster. Bottom: 20-step and one-step relative prediction errors of the NAR models determined by SINAR for different memory depths \( p \) with \( \lambda = 0 \) and \( \lambda = 0.05 \). Memory terms yield a significant decrease in the prediction errors compared to Markovian predictions.

In order to illustrate why memory terms improve the prediction accuracy, let us imagine for now that there are no links between the clusters. Then the evolutions of opinion percentages in both clusters run in parallel to each other and are Markovian as derived previously. The opinion percentages in the full network are then given by the averages of the cluster-wise percentages \( x^{(i)}_t \), i.e., \( x_t = \frac{1}{2}(x^{(1)}_t + x^{(2)}_t) \). This means, if we know \( x_t \), then there are various options for what \( x^{(1)}_t \) and \( x^{(2)}_t \) can be, all of which might result in different values for \( x^{(1)}_{t+1} \) and \( x^{(2)}_{t+1} \) and thus \( x_{t+1} \). If we are additionally given \( x_{t-1} \), this might yield possible values for \( x^{(1)}_{t-1} \) and \( x^{(2)}_{t-1} \) which themselves make some of the candidates for \( x^{(1)}_t \) and \( x^{(2)}_t \) unlikely, e.g., because they are far away from them. Thus, through the information of memory terms we can restrict the options for what the percentages inside each cluster are. We illustrate this in more detail in Appendices A.1 and A.3.

The links between the clusters have as consequence that within one cluster agents generally do not have identical opinion change probabilities since their neighborhoods are different. This yields additional need for memory terms since then not even for the macrodynamics in one cluster, a Markovian formulation can be derived.
Figure 9: Opinion percentages over one block of length 20 from the validation data and prediction evolutions with NAR models obtained with SINAR for $p = 1, 2$ and $10$ and $\lambda = 0$. Percentages from validation data are depicted with thin and predicted percentages with thick lines. With $p = 1$, the prediction accuracy is poor and improves drastically for $p = 2$. With $p = 10$, the predicted evolutions come even closer to the curves from the validation data.

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 $p_{\text{between}} = 0.0001$. 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 $(0.5, 0.3, 0.2)$. The evolution of the opinion percentages is now much more irregular compared to the previous examples. The oscillatory behaviour is still present but the amplitudes differ from time to time. Through the higher number of clusters, more randomness comes into the model since an opinion can be randomly spread from one cluster, where it is dominant, to another one, where it is not dominant, suddenly altering the evolution of percentages in this cluster and thus in the whole network.

We now show that, similar to when we used a two-cluster network, memory terms become important for predictions of the evolution of the microdynamics. This can be seen in Figure 10. Again, the mean relative error per block converges with increasing $p$. While in the two-cluster network example the performance did not improve visibly with $p > 10$, in this case we can get slightly lower errors for $p$ approaching 20.

For $p = 2$ and $\lambda = 0.05$, we obtain the NAR model

$$(x_{t+1})_1 = 1.8745(x_t)_1 - 0.8748(x_{t-1})_1$$

$$(x_{t+1})_2 = 1.8672(x_t)_2 - 0.8674(x_{t-1})_2.$$
For $p > 2$, the models show increasing complexity, e.g., for $p = 3$:

\[
(x_{t+1})_1 = 1.4662(x_t)_1 - 0.1188(x_t)_2 + 0.0552(x_t)_1^2 + 0.1318(x_t)_1(x_t)_2 + 0.2309(x_{t-1})_2 \\
- 0.1899(x_{t-1})_1(x_{t-1})_2 - 0.2021(x_{t-1})_2^2 - 0.4658(x_{t-2})_1 - 0.1060(x_{t-2})_2 \\
+ 0.1206(x_{t-2})_1^2 + 0.0644(x_{t-2})_2^2 \\
(x_{t+1})_2 = 1.3157(x_t)_2 - 0.3161(x_{t-2})_2.
\]

Again, we show as an example the predictions of percentages for one block of length 40 with memory depths 1, 2 and 10 (Figure 11).

![Figure 10: Top left: One realisation of the microdynamics. 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 macrodynamics $\xi(X)$. The behaviour is much more complex than in the first two cases. Bottom: 20-step and one-step relative prediction errors of the NAR models determined by SINAR for different memory depths $p$ with $\lambda = 0$ and $\lambda = 0.05$.](image-url)
Figure 11: Opinion percentages over one block of length 40 from the validation data and prediction evolutions with NAR models obtained with SINAR for \( p = 1, 2 \) and 20 and \( \lambda = 0 \). Percentages from validation data are depicted with thin and predicted percentages with thick lines. As in the example with a two-cluster network, we can see what the error graphs in Figure 10 indicate: The predicted evolutions are closer to the validation data with higher memory depth of the NAR model.

As in the example with the two-cluster network, we could see that a higher memory depth indeed increases the prediction accuracy for the evolution of the opinion percentages in the short term, i.e., for predictions of length 20 resp. 40. Plus, enforcing the sparsity constraint with the parameter \( \lambda \) in SINAR set to 0.05 yielded significantly sparser models while the prediction accuracy only suffered slightly.

6 Discussion

In this article, we have summarized how the evolution of observations of a dynamical system can be derived through the Mori–Zwanzig formalism, and how this can result in a nonlinear autoregressive model with memory. For the determination of model parameters, we have used methodology from data-driven system identification methods, inspired by SINDy [BPK16a]. We could then extend SINDy to SINAR which identifies sparse NAR models from data, thus deploying a common system identification method for non-Markovian systems.

We applied this to an agent-based model (ABM) that simulates the dynamics of opinion changes in a population. Assuming that all agents are equally strongly influenced by all other agents in the population, we showed that for the prediction of the percentages of opinions within the population memory terms are not necessary. However, for incomplete networks, this is no longer the case. Our methodology enabled us to make more accurate predictions for the percentages of opinions among the agents when the population of agents was defined by clusters with little influence between them. Additionally, sparse models obtained from enforcing a sparsity constraint in the estimation of NAR models in SINAR, gave almost equally good prediction accuracy as the non-sparse ones, while yielding far simpler models. In the context of opinion dynamics, such sparse models permit to point out more clearly
which opinions impact which and how.

In the future, the following challenges are to be addressed:

- In our methodology, we have assumed a noise term resulting from Mori–Zwanzig that was zero-mean. This allowed us to omit it when making predictions of the expected value of the opinion percentages. This simplifying assumption does not need to be true, and one could try to derive a more accurate representation for the noise term. As a result of this simplifying assumption, the NAR models we considered were deterministic, even for non-deterministic microdynamics. Introduction of explicit noise in the NAR models, e.g., by extending the approach outlined in [KNP +20], could improve their (statistical) predictive capacities.

- One could additionally choose a different projection $P$ in the Mori–Zwanzig formalism. The choice of an orthogonal projection on a finite set of basis functions explicitly yielded an NAR model. The right projection for a given system could inspire an optimal choice of basis functions, e.g., such that the memory depth is minimal.

- We have derived models that are stationary, i.e., do not change over time. Since the assumption of an equilibrium distribution over states of the microdynamics might not always hold, coefficients of the NAR model may become time-dependent. One could use a regime switching model as in [Hor11] that fixes coefficients for a time interval before changing them to other fixed values when the macrodynamics show certain behaviour, e.g., coefficients might be different depending on which opinion is dominating.

Acknowledgements

PK and CS acknowledge support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID: 390685689). NW thanks Luzie Helfmann, Jan-Hendrik Niemann and Alexander Sikorski for helpful discussions on the subject of opinion dynamics.

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 $E[x_{t+1}]$. Rather, the inclusion of $x_{t-1}$ gives a different expected value for $x_{t+1}$.

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 \(x_t = \left(\frac{2}{3}, \frac{1}{3}\right)^T\). The network forms a line, so that agents 1 and 3 are connected to agent 2 but not to each other. All \(\alpha_{m'n''}\) are set to 1. Assume prior information that yield for us a uniform distribution about the possible opinion vectors \(X \in \mathbb{X}\) on the microscale, i.e., \(\mathbb{P}[X_t = X|x_t] \equiv \text{const}\), the probability that \(\mathbb{P}[x_{t+1} = (1, 0)^T|x_t = \left(\frac{2}{3}, \frac{1}{3}\right)^T]\) 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,

\[
\mathbb{P}[x_{t+1} = (1, 0)^T|x_t = \left(\frac{2}{3}, \frac{1}{3}\right)^T] = 
\begin{align*}
&\mathbb{P}[x_{t+1} = (1, 0)^T|X_t = (\bullet - \circ - \circ)]\mathbb{P}[X_t = (\bullet - \circ - \circ)|x_t = \left(\frac{2}{3}, \frac{1}{3}\right)^T] \\
&+ \mathbb{P}[x_{t+1} = (1, 0)^T|X_t = (\bullet - \circ - \bullet)]\mathbb{P}[X_t = (\bullet - \circ - \bullet)|x_t = \left(\frac{2}{3}, \frac{1}{3}\right)^T] \\
&+ \mathbb{P}[x_{t+1} = (1, 0)^T|X_t = (\circ - \bullet - \bullet)]\mathbb{P}[X_t = (\circ - \bullet - \bullet)|x_t = \left(\frac{2}{3}, \frac{1}{3}\right)^T] \\
&= \mathbb{P}[x_{t+1} = (1, 0)^T|X_t = (\circ - \bullet - \bullet)] \frac{2}{3} \\
&+ \mathbb{P}[x_{t+1} = (1, 0)^T|X_t = (\bullet - \circ - \bullet)] \frac{1}{3} \\
&= \frac{2}{3} \cdot \frac{1}{3} \cdot \frac{2}{3} + \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{2}{3} = \frac{4}{18} + \frac{1}{18} = \frac{5}{18} \approx 0.2778.
\]

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 percentages. 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 \(x_{t-1} = \left(\frac{2}{3}, \frac{1}{3}\right)^T\), then it holds that

\[
\mathbb{P}[x_{t+1} = (1, 0)^T|x_t = \left(\frac{2}{3}, \frac{1}{3}\right)^T] \neq \mathbb{P}[x_{t+1} = (1, 0)^T|x_t = \left(\frac{2}{3}, \frac{1}{3}\right)^T, x_{t-1} = \left(\frac{2}{3}, \frac{1}{3}\right)^T],
\]

because:

\[
\mathbb{P}[x_{t+1} = (1, 0)^T|x_t = \left(\frac{2}{3}, \frac{1}{3}\right)^T, x_{t-1} = \left(\frac{2}{3}, \frac{1}{3}\right)^T] = 
\sum_{X:X \in \mathbb{X}} \mathbb{P}[x_{t+1} = (1, 0)^T|X_t = X]\mathbb{P}[X_t = X|X_{t-1} = \hat{X}, x_t = \left(\frac{2}{3}, \frac{1}{3}\right)^T] = \\
\left(\frac{2}{3}, \frac{1}{3}\right)^T \mathbb{P}[X_{t-1} = \hat{X}|x_{t-1} = \left(\frac{2}{3}, \frac{1}{3}\right)^T] \\
- \frac{1}{3} \text{ by assumption}
\]

We find that the probabilities for \(X_t\) under \(X_{t-1}\), assuming that exactly two nodes in \(X_{t-1}\) and \(X_t\) are black, are given as in the following table:

32
\[
\begin{array}{c|ccc}
X_{t-1} \rightarrow X_t & \bullet - \bullet - \circ & \bullet - \circ - \bullet & (\circ - \bullet - \bullet) \\
\hline
(\bullet - \bullet - \circ) & 2/3 & 1/3 & 0 \\
(\bullet - \circ - \bullet) & 2/5 & 1/5 & 2/5 \\
(\circ - \bullet - \bullet) & 0 & 1/3 & 2/3 \\
\end{array}
\]

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

\[
P[X_t = (\bullet - \bullet - \circ)|x_{t-1}] = \left(\frac{2}{3}, \frac{1}{3}\right)^T \approx \frac{16}{45}
\]

\[
P[X_t = (\bullet - \circ - \bullet)|x_{t-1}] = \left(\frac{2}{3}, \frac{1}{3}\right)^T \approx \frac{13}{45}
\]

\[
P[X_t = (\circ - \bullet - \bullet)|x_{t-1}] = \left(\frac{2}{3}, \frac{1}{3}\right)^T \approx \frac{16}{45}
\]

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

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

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

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

With this, we obtain

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

This is unequal to the value of \(\approx 0.2778\) that we computed for \(P[x_{t+1} = (1, 0)^T|x_t = \left(\frac{2}{3}, \frac{1}{3}\right)^T]\).

The additional knowledge that \(x_{t-1} = \left(\frac{2}{3}, \frac{1}{3}\right)^T\) 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 \(X_{t+1} = (\bullet - \bullet - \circ)\) than that \(X_{t+1} = (\circ - \bullet - \bullet)\) or \((\bullet - \bullet - \circ)\).

### A.2 Derivation of Equation \([5.3]\)

With \(m = 3\) opinions, Equation \([5.1]\) reads

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

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

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

33
Rearranging gives

\[(x_{t+1})_1 = (1 + \alpha_{31} - \alpha_{13})(x_t)_1 + (\alpha_{13} - \alpha_{31})(x_t)_1^2 + (\alpha_{21} - \alpha_{12} - \alpha_{31} + \alpha_{13})(x_t)_1(x_t)_2 \]
\[(x_{t+1})_2 = (1 + \alpha_{32} - \alpha_{23})(x_t)_2 + (\alpha_{23} - \alpha_{32})(x_t)_2^2 + (\alpha_{12} - \alpha_{21} - \alpha_{32} + \alpha_{23})(x_t)_1(x_t)_2. \]

This is Equation (5.3).

### A.3 Representations of uncoupled expected two-cluster dynamics

In this subsection, we discuss the derivation of NAR models for a network which consists of two equally-sized clusters without links between them. Having derived the expected dynamics for a complete network in Equation (5.1), we assume for now that the expected dynamics are identical with the true dynamics in order to investigate the macrodynamics if the agents behave perfectly as expected. We then get Markovian deterministic dynamics that describe the evolution of opinion percentages in each cluster. Their means are the opinion percentages in the whole network. The derivation of an NAR model for this property is analytically challenging but numerical results suggest certain structures of the macrodynamics dependent on the initial percentages.

**Macrodynamics inside the clusters.** Since the clusters represent complete networks of their own, we obtain for the opinion percentages \(x_t^{(i)}\) inside each cluster

\[(x_{t+1})_1^{(i)} = (1 + \alpha_{31} - \alpha_{13})(x_t^{(i)})_1 + (\alpha_{13} - \alpha_{31})(x_t^{(i)})_1^2 + (\alpha_{21} - \alpha_{12} - \alpha_{31} + \alpha_{13})(x_t^{(i)})_1(x_t^{(i)})_2 \]
\[(x_{t+1})_2^{(i)} = (1 + \alpha_{32} - \alpha_{23})(x_t^{(i)})_2 + (\alpha_{23} - \alpha_{32})(x_t^{(i)})_2^2 + (\alpha_{12} - \alpha_{21} - \alpha_{32} + \alpha_{23})(x_t^{(i)})_1(x_t^{(i)})_2. \]

(A.1)

With \(x_t = \frac{1}{2}(x_t^{(1)} + x_t^{(2)})\) and denoting \(a = \alpha_{31} - \alpha_{13}, b = \alpha_{21} - \alpha_{12} - \alpha_{31} + \alpha_{13}, c = \alpha_{32} - \alpha_{23}, d = \alpha_{12} - \alpha_{21} - \alpha_{32} + \alpha_{23},\) this gives

\[(x_{t+1})_1 = (1 + a) \frac{1}{2} ((x_t^{(1)})_1 + (x_t^{(2)})_1) - \frac{a}{2} ((x_t^{(1)})_1^2 + (x_t^{(2)})_1^2) + \frac{b}{2} ((x_t^{(1)})_1(x_t^{(1)})_2 + (x_t^{(2)})_1(x_t^{(2)})_2) \]
\[(x_{t+1})_2 = (1 + c) \frac{1}{2} ((x_t^{(1)})_2 + (x_t^{(2)})_2) - \frac{c}{2} ((x_t^{(1)})_2^2 + (x_t^{(2)})_2^2) + \frac{d}{2} ((x_t^{(1)})_1(x_t^{(1)})_2 + (x_t^{(2)})_1(x_t^{(2)})_2). \]
Even making the simplifying assumption that \( a = -c \) and \( b = -d = -2a \) as is the case for the coefficients we chose for the examples, we arrive at

\[
(x_{t+1})_1 = (1 + a) \frac{1}{2}((x_t^{(1)})_1 + (x_t^{(2)})_1) - \frac{a}{2}((x_t^{(1)})_1^2 + (x_t^{(2)})_1^2) - a((x_t^{(1)})_2(x_t^{(1)})_1 + (x_t^{(2)})_1(x_t^{(2)})_2)
\]

\[
(x_{t+1})_2 = (1 - a) \frac{1}{2}((x_t^{(1)})_2 + (x_t^{(2)})_2) - \frac{a}{2}((x_t^{(1)})_2^2 + (x_t^{(2)})_2^2) - a((x_t^{(1)})_2(x_t^{(1)})_1 + (x_t^{(2)})_1(x_t^{(2)})_2).
\]

From this, it seems impossible to find a closed Markovian expression for \( x_t \). In order to understand why memory terms should help to express the evolution of \( x_t \), note the following: Given \( x_{t-1} \) and \( x_t \), we could now find \( x_{t-1}^{(1)} \) and \( x_{t-1}^{(2)} \) so that these equations would yield those values for \( x_t^{(1)} \) and \( x_t^{(2)} \) whose average is \( x_t \). This set of pairs of \( x_t^{(1)} \) and \( x_t^{(2)} \) would significantly be limited compared to all pairs which have this \( x_t \) as their average. From these \( x_t^{(i)} \), we could compute subsequent values \( x_{t+1}^{(i)} \). Hence, we would have gained a more precise estimate of \( x_t^{(1)} \) and \( x_t^{(2)} \) and thus of \( x_{t+1} \). In the stochastic ABM, the evolution of \( x_t \) is originally stochastic if it represents the percentages of opinions of agents. Hence, one would not search for the \( x_t^{(i)} \) that exactly yield \( x_t \) but rather make this argument in terms of probabilities. We would then get different probabilities for the \( x_t^{(i)} \) dependent on what \( x_{t-1} \) is. Note, that this is an almost analogous argumentation to the one in Appendix A.1. Additional information about terms at time \( t - 1 \) alters the probabilities for what the inaccessible states at time \( t \) are. In Appendix A.1, these inaccessible states were the opinions of the three agents. In this setting, they are the \( x_t^{(i)} \).

**Simplified example: Linear dynamics inside the clusters.** Of course, a closed expression for the evolution of \( x_{t+1} \) that depends only on memory terms of \( x_t \) and not on the \( x_t^{(i)} \) is desirable. However, the analytical derivation of such an expression seems out of reach. Thus, as an example for much simpler macrodynamics inside each cluster, we illustrate how one can find a closed expression for the mean of two linear dynamics. For this, let

\[
x_t^{(1)} = \lambda_1 x_t^{(1)}
\]

\[
x_t^{(2)} = \lambda_2 x_t^{(2)}
\]

and

\[
x_t = \frac{1}{2} (x_t^{(1)} + x_t^{(2)}).
\]

Thus,

\[
x_t^{(i)} = \lambda_t x_0^{(i)}, \quad i = 1, 2
\]

and

\[
x_t = \frac{1}{2} (\lambda_t x_0^{(1)} + \lambda_t x_0^{(2)}).
\]

Then one can observe that

\[
x_{t+1} = \frac{(\lambda_1 + \lambda_2)}{2} x_t - \frac{\lambda_1 \lambda_2}{2} x_{t-1}
\]

35
since
\[
\frac{(\lambda_1 + \lambda_2)}{2} x_t - \frac{\lambda_1 \lambda_2}{2} x_{t-1} = \frac{1}{2}[(\lambda_1 + \lambda_2)(\lambda_1 x_{t-1}^{(1)} + \lambda_2 x_{t-1}^{(2)}) - \lambda_1 \lambda_2 (\lambda_1^{-1} x_0^{(1)} + \lambda_2^{-1} x_0^{(2)})]
\]
\[
= \frac{1}{2}[(\lambda_1^{t+1} x_0^{(1)} + \lambda_2^{t+1} x_0^{(2)}) + \lambda_1 \lambda_2 x_0^{(2)} + \lambda_2 \lambda_1 x_0^{(2)} - \lambda_2 \lambda_1 x_0^{(1)} - \lambda_1 \lambda_2 x_0^{(2)}]
\]
\[
= \frac{1}{2}(\lambda_1^{t+1} x_0^{(1)} + \lambda_2^{t+1} x_0^{(2)}) = \frac{1}{2}(x_t^{(1)} + x_t^{(2)}) = x_{t+1}.
\]

**Numerical results with symmetric initial percentages.** For the macrodynamics (A.1) of opinion percentages in a two-cluster network, we have not derived such a closed expression analytically. However, we can see numerically that almost exact models can be derived for a memory depth of \( p = 2 \) if we impose symmetric starting conditions, i.e., initial percentages that fulfill
\[
(x_0^{(1)})_1 = (x_0^{(2)})_2 = 1 - 2(x_0^{(1)})_2, \quad (x_0^{(2)})_1 = (x_0^{(1)})_2 = 1 - 2(x_0^{(2)})_2.
\]

To illustrate this, we create trajectories of length \( T = 900 \) of the deterministic dynamics (A.1) with initial percentages
\[
(x_0^{(1)})_1 = 0.8, \quad (x_0^{(2)})_2 = 0.1, \quad (x_0^{(2)})_1 = 0.1, \quad (x_0^{(2)})_2 = 0.8.
\]
and \( a = 0.135 \) which is also the case in the examples in Section 5.

From the first 500 time steps of the resulting \( x_t = \frac{1}{2}(x_t^{(1)} + x_t^{(2)}) \) we estimate the NAR model (with \( \lambda = 0 \) in SINAR)
\[
(x_{t+1})_1 = 1.21(x_t)_1 - 0.65(x_t)_2 + 0.27(x_t)_2^2 + 0.54(x_t)_1(x_t)_2
\]
\[
- 0.26(x_{t-1})_1 + 0.71(x_{t-1})_2 - 0.17(x_{t-1})_2^2 - 0.1(x_{t-1})_1(x_{t-1})_2 - 0.54(x_{t-1})_1(x_{t-1})_2
\]
\[
(x_{t+1})_2 = -0.82(x_t)_1 - 1.31(x_t)_2 - 0.27(x_t)_2^2 - 0.54(x_t)_1(x_t)_2
\]
\[
+ 0.68(x_{t-1})_1 - 0.16(x_{t-1})_2 - 0.30(x_{t-1})_2^2 - 0.03(x_{t-1})_1^2 + 0.54(x_{t-1})_1(x_{t-1})_2.
\]

With this model, we reconstruct the remaining 400 time steps in the data by computing a trajectory of length 400 with starting values given by \( x_{499} \) and \( x_{500} \) (Figure 12). The relative Euclidean error between both trajectories amounts to \( 2.4 \cdot 10^{-7} \). For the one-step prediction, i.e., mapping every two values \( x_{t-1} \) and \( x_t \) to \( x_{t+1} \) with the above model, the error is \( 1.5 \cdot 10^{-14} \). For larger memory depths, there is no improvement in prediction accuracy. This suggests that for these specific initial conditions the macrodynamics can be reproduced with memory depth \( p = 2 \).

**Numerical results with non-symmetric initial percentages.** For other initial percentages, we get quite different coefficients that significantly decrease the influence of the second-order terms \( (x_t)_1^2, (x_t)_2^2 \) and \( (x_t)_1(x_t)_2 \). Let
\[
(x_0^{(1)})_1 = 0.7, \quad (x_0^{(1)})_2 = 0.2, \quad (x_0^{(2)})_1 = 0.1, \quad (x_0^{(2)})_2 = 0.8.
\]
Then for $p = 2$, in the same manner ($\lambda = 0$), we obtain the model

\[
\begin{align*}
(x_{t+1})_1 &= 2.09(x_t)_1 - 0.01(x_t)_2 - 0.09(x_t)_1^2 + 0.01(x_t)_2^2 - 0.15(x_t)_1(x_t)_2 \\
&\quad - 1.09(x_{t-1})_1 + 0.01(x_{t-1})_2 + 0.09(x_{t-1})_1^2 - 0.02(x_{t-1})_2 + 0.15(x_{t-1})_1(x_{t-1})_2 \\
(x_{t+1})_2 &= -0.04(x_t)_1 - 1.90(x_t)_2 - 0.01(x_t)_1^2 - 0.08(x_t)_2^2 + 0.15(x_t)_1(x_t)_2 \\
&\quad + 0.04(x_{t-1})_1 - 0.90(x_{t-1})_2 - 0.08(x_{t-1})_1^2 - 0.16(x_{t-1})_1(x_{t-1})_2.
\end{align*}
\]

The original trajectories and the trajectories obtained from this model are depicted in Figure 13. The one-step prediction error improves for memory depths larger than $p = 2$ (Figure 14). Since with NAR models obtained from the trajectories for these initial percentages, the
predicted trajectories diverge, the full prediction error is not shown.

Figure 14: One-step prediction error for the NAR models obtained from trajectories of $x_t$ with initial percentages of the $x_t^{(i)}$ as given in (A.4).

In summary, for a network that consists of two clusters which are uncoupled but fully connected internally, the expected macrodynamics are given by the mean of the expected intra-cluster dynamics. Assuming the dynamics to have no variance and hence to be deterministic, given in Equation (A.1), with symmetric initial percentages, a memory depth of 2 is enough for us to generate an almost exact NAR model for the macrodynamics. However, for non-symmetric initial percentages, the ensuing best-fitting NAR models with the basis functions we use are not accurate in the long-term. This seems to be in part due to the fact, that for non-symmetric initial percentages, the trajectories show more complex behaviour which no longer consists of periodic oscillations but is rather more irregular. This could cause the best-fitting NAR models to then be dominated by linear terms. Results about to which degree one can analytically derive NAR models for both symmetric and non-symmetric initial percentages require further research.

A.4 Definition of the conditional expectation

Let states $X \in \mathbb{X}$ be distributed according to $\mu$. Let us define for $\xi \in \mathcal{G}$ the level sets $L_x := \{X \in \mathbb{X} : \xi(X) = x\}$. Then, through the coarea formula \cite{Fed96}, the expectation of a function $g \in \mathcal{G}$ with $g \in L^1(\mathbb{X})$ can be written as

$$E_\mu[g(X)] = \int_\mathbb{X} f(X) d\mu(X) = \int_{\xi(\mathbb{X})} \int_{L_x} f(X) \mu(X) \det(\nabla \xi(X)^T \nabla \xi(X))^{-\frac{1}{2}} \, dx \, d\sigma_x(X)$$

where $\sigma_x$ is the Hausdorff measure on $L_x$. Then, the conditional expectation of $f(X)$ given that $\xi(X) = x$ is (see, e.g., \cite{BKK+18})

$$E_\mu[g(X) | \xi(X) = x] = \frac{1}{\Gamma(x)} \int_{L_x} g(X) \mu(X) \det(\nabla \xi(X)^T \nabla \xi(X))^{-\frac{1}{2}} \, d\sigma_x(X),$$

38
where $\Gamma(x)$ is a normalization constant.

### A.5 Determination of coefficients of linear AR models

A linear autoregressive model with zero-mean Gaussian noise has the form

$$x_{t+1} = \sum_{i=0}^{p-1} H_i x_{t-i} + \varepsilon_{t+1}, \quad \varepsilon_{t+1} \sim \mathcal{N}(0, \Sigma^T \Sigma).$$

The best linear unbiased estimator (BLUE) [Pla49, BK81] for the $H_i$ is the Least Squares minimizer $\hat{\Xi} = [H_0, \ldots, H_{p-1}]$, given by

$$\hat{\Xi} = \arg\min_{\hat{\Xi} = [H_0, \ldots, H_{p-1}]} \|X' - \hat{\Xi}\Xi\|_F,$$

where $\bar{X}$ and $X'$ are defined as in (3.6).

Omitting the sparsity constraint, SINAR solves the problem

$$\hat{\Xi} = \arg\min_{\hat{\Xi}} \|X' - \hat{\Xi}\Theta(\bar{X})\|_F.$$

If $\Theta(\bar{x}) = \bar{x}$, then this is precisely the Least Squares method for linear autoregressive models.

### A.6 Covariance of noise terms of NAR models

Assuming a relation of the form

$$x'_t = \Xi \Theta(x_t) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \Sigma^T \Sigma),$$

we find that

$$\text{Cov}(x'_t - \Xi \Theta(x_t)) = \text{Cov}(\varepsilon_t).$$

An unbiased estimator for the covariance of a random variable $y$ is the statistical covariance

$$\Sigma = \frac{1}{T-1} \sum_{t=1}^{T} (y_t - \bar{y})(y_t - \bar{y})^T,$$

where $\bar{y} = \frac{1}{T} \sum_{t=1}^{T} y_t$.

In order to estimate the covariance matrix of noise terms $\varepsilon_{t+1}$ in Equation (2.12), one has to substitute $x'_t$ by $x_{t+1}$ and $\Xi \Theta(x_t)$ by $\sum_{k=0}^{p-1} H_k \tilde{\varphi}(x_{t-k})$ to derive the form of Equation (2.12). Subsequently $y$ has to be substituted by $x_{t+1} - \sum_{k=0}^{p-1} H_k \tilde{\varphi}(x_{t-k})$ and we can calculate the statistical covariance of $\varepsilon_{t+1}$ in (2.12).
A.7 Hausdorff distance of delay embedding of trajectories

The Hausdorff distance between two non-empty compact sets measures the maximal minimal distance a point from one set has to the other set. It is commonly used to compare attractors of dynamical systems. The lower the Hausdorff distance between two sets, the more similar they are. From two trajectories $X' = [x_0, \ldots, x_T]$ and $\hat{X}' = [\hat{x}_0, \ldots, \hat{x}_T]$, we construct the delay embeddings with embedding depth $p$ as

$$D_p(X') = \left[ \begin{array}{c} x_{p-1} \\ \vdots \\ x_0 \end{array}, \begin{array}{c} x_p \\ \vdots \\ x_1 \end{array}, \ldots \right], \quad D_p(\hat{X}') = \left[ \begin{array}{c} \hat{x}_{p-1} \\ \vdots \\ \hat{x}_0 \end{array}, \begin{array}{c} \hat{x}_p \\ \vdots \\ \hat{x}_1 \end{array}, \ldots \right].$$

We then calculate their Hausdorff distance as

$$\max\left(\max_{x \in D_p(X')} \min_{\hat{x} \in D_p(\hat{X}')} \|x - \hat{x}\|_2, \max_{\hat{x} \in D_p(\hat{X}')} \min_{x \in D_p(X')} \|x - \hat{x}\|_2\right).$$

References

[ADP14] K. Aho, D. Derryberry, and T. Peterson. Model selection for ecologists: the worldviews of AIC and BIC. *Ecology*, 95(3):631–636, 2014.

[AH96] H. An and F. Huang. The geometrical ergodicity of nonlinear autoregressive models. *Statistica Sinica*, 6:943–956, 1996.

[AY19] B. D. O. Anderson and M. Ye. Recent advances in the modelling and analysis of opinion dynamics on influence networks. *International Journal of Automation and Computing*, 16:129–149, 2019.

[Ban16] S. Banisch. *Markov Chain Aggregation for Agent-Based Models*, volume 1. Springer, 2016.

[BBP+16] S. Brunton, B. Brunton, J. Proctor, E. Kaiser, and J. Kutz. Chaos as an intermittently forced linear system. *Nature Communications*, 8, 2016.

[BCN17] P. Bolzern, P. Colaneri, and G. Nicolao. Opinion influence and evolution in social networks: A markovian agents model. *Automatica*, 100, 11 2017.

[BD91] P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*, volume 2. Springer, 1991.

[BDM14] T. Biancalani, L. Dyson, and A. J. McKane. Noise-induced bistable states and their mean switching time in foraging colonies. *Physical review letters*, 112(3):038101, 2014.

[Ber92] A. A. Berryman. The orgins and evolution of predator-prey theory. *Ecology*, 73(5):1530–1535, 1992.
[FSG+07] A. Fujita, J. Sato, M. Garay, R. Yamaguchi, S. Miyano, M. Sogayar, and C. Ferreira. Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC systems biology*, 1:39, 02 2007.

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

[HEVEDB10] C. Hijón, P. Español, E. Vanden-Eijnden, and R. Delgado-Buscalioni. Mori–Zwanzig formalism as a practical computational tool. *Faraday discussions*, 144:301–22; discussion 323, 2010.

[HHSN07] I. Horenko, C. Hartmann, C. Schütte, and F. Noe. Data-based parameter estimation of generalized multidimensional Langevin processes. *Phys. Rev. E*, 76:016706, 2007.

[HO93] P. Hansen and D. O’leary. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.*, 14:1487–1503, 1993.

[Hor11] I. Horenko. On analysis of nonstationary categorical data time series: Dynamical dimension reduction, model selection, and applications to computational sociology. *Multiscale Modeling & Simulation*, 9:1700–1726, 2011.

[HS97] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 1997.

[JSN13] M. Jovanovic, P. Schmid, and J. Nichols. Sparsity-promoting dynamic mode decomposition. *Physics of Fluids*, 26, 2013.

[JSW98] N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1:7–38, 03 1998.

[KDG15] D. Kondrashov, M. D.Chekroun, and M. Ghil. Data-driven non-Markovian closure models. *Physica D: Nonlinear Phenomena*, 297:33–55, 2015.

[KK08] S. Konishi and G. Kitagawa. *Information Criteria and Statistical Modeling*, volume 1. Springer, 2008.

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

[KLT07] P. Klimek, R. Lambiotte, and S. Thurner. Opinion formation in laggard societies. *EPL (Europhysics Letters)*, 82, 2007.

[KNP+20] S. Klus, F. Nüske, S. Peitz, J.-H. Niemann, C. Clementi, and C. Schuette. Data-driven approximation of the Koopman generator: Model reduction, system identification, and control. *Physica D*, 406:132416, 2020.
[Sta15] C. Stangor. *Social Groups in Action and Interaction*, volume 2. Routledge, 2015.

[Tak81] F. Takens. Detecting strange attractors in turbulence. 898:366–381, 1981.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, no. 1, 58:267–288, 1996.

[TRL+14] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1:391–421, 2014.

[Tuy18] L. Tuyen. A higher order Markov model for time series forecasting. *International Journal of Applied Mathematics and Statistics*, 57:1–18, 2018.

[VVR17] S. C. Venkataramani, R. C. Venkataramani, and J. M. Restrepo. Dimension reduction for systems with slow relaxation. *J Stat Phys, Section 4.4*, 167:892–933, 2017.

[WKR14] M. Williams, I. Kevrekidis, and C. Rowley. A data-driven approximation of the Koopman operator: Extending Dynamic Mode Decomposition. *Journal of Nonlinear Science*, 25, 2014.

[WNWS20] S. Winkelmann, J.-H. Niemann, S. Wolf, and C. Schütte. Agent-based modeling: Population limits, metastability and large deviations. 2020.

[XWX11] H. Xia, H. Wang, and Z. Xuan. Opinion dynamics: A multidisciplinary review and perspective on future research. *IJKSS*, 2:72–91, 2011.

[ZDV18] Y. Zhu, J. Dominy, and D. Venturi. On the estimation of the Mori-Zwanzig memory integral. *Journal of Mathematical Physics*, 59:103501, 2018.

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