A METHOD FOR INFERRING HIERARCHICAL DYNAMICS IN STOCHASTIC PROCESSES

OLOF GÖRNERUP and MARTIN NILSSON JACOBI
Complex Systems Group, Department of Energy and Environment, Chalmers University of Technology, 412 96 Göteborg, Sweden
olofgo@chalmers.se, mjacobi@chalmers.se

Complex systems may often be characterized by their hierarchical dynamics. In this paper we present a method and an operational algorithm that automatically infer this property in a broad range of systems; discrete stochastic processes. The main idea is to systematically explore the set of projections from the state space of a process to smaller state spaces, and to determine which of the projections that impose Markovian dynamics on the coarser level. These projections, which we call Markov projections, then constitute the hierarchical dynamics of the system. The algorithm operates on time series or other statistics, so a priori knowledge of the intrinsic workings of a system is not required in order to determine its hierarchical dynamics. We illustrate the method by applying it to two simple processes; a finite state automaton and an iterated map.

Keywords: Hierarchical dynamics; Model reduction; Coarse graining.

1. Introduction

Modularity and hierarchical organization play an important role when determining the character of a dynamical system. It could even be argued that hierarchical self-organization is a necessary condition for a system to display a high degree of complexity, see e.g. Simon [1]. Hierarchical dynamics is also a prerequisite for efficient model reduction. Then the general strategy when reducing the level of details in a model is to find a partition of the degrees of freedom (i.e. a projection of the phase space) which by itself form a system with Markovian dynamics; an observation which is also discussed in detail by Shalizi and Moore [2]. Conversely one can use the idea behind the time-delay embedding method for attractor reconstruction [3, 4] to convince oneself that a dynamical system without the Markov property should be reconstructed in a higher dimensional phase space in order to make sense as a causal model.

In a physical system, modularity is usually associated with separations in time and length scales. In this paper we focus on hierarchical decomposition of stochastic processes that, in general, are systems without direct physical interpretation. Despite the void of guidance from physical intuition, there are several methods that can be used for determining their hierarchical structure. For example, if one has full access to the inner workings of a process’ dynamics, i.e. the generative semi-group, Krohn-Rhodes theory can be used to decompose the semi-group as a hierarchical
Wreath product of finite groups and finite aperiodic semi-groups. Here do we present a method for hierarchical decomposition and reconstruction that instead operates on the sequence of states, i.e. a time series, that is generated by the process at hand. In this way, prior knowledge of the process’ intrinsic dynamics is not necessary.

1.1. Historical background

The line of ideas on decomposition of dynamical systems and identification of hierarchical dynamics can be traced back to the analysis of continuous symmetries in classical mechanics as advanced by Lie, Lagrange, Poisson, Jacobi and Noether. These reduction schemes result in the elimination of inactive, i.e. constant, degrees of freedom. In non-equilibrium statistical mechanics, dimensional reduction generally means going from an effectively deterministic model to a Langevin type model that includes a noise term. The noise stems from fast, usually chaotic, motion that on the time scale of the relevant (slow) degrees of freedom can be approximated as white noise, resulting in a Markovian dynamics for the slow degrees of freedom. This idea was first formalized by Zwanzig and has later matured into the theory of adiabatic elimination, see e.g. [7, 8]. Yet another situation frequently encountered in models of natural systems is dissipative driven processes. A generic feature of such systems is that fast degrees of freedom, due to large negative Lyaponov exponents associated with the dissipation, often relaxes to a quasi-fixed point, i.e. a point in the phase space that appears effectively fixed on the time scale of the fast dynamics, but changes on the time scale set by the slow degrees of freedom. The overall dynamics is therefore in this case slaved to a slow positively invariant manifold and the resulting dimensionality is reduced. This picture has been advanced by Haken in his work on self-organization [9]. Lately this idea has also been revitalized in the turbulence community, primarily by a proof of existence of inertial manifolds in a class of hyperbolic dynamical systems [10]. In the same spirit, positive invariant manifolds are used in model reduction schemes in chemical kinetics [11]. Finally it is also worth mentioning that the connection between chaotic dynamical systems and non-equilibrium statistical mechanics has recently been further clarified by the work of Ruelle et. al. [12, 13].

1.2. The method

We start with a discussion on the general problem of how to define a hierarchical organization in a dynamical system. From a constructive point of view, a hierarchical system should be composed of interacting modules that contain, in some sense, smaller interacting modules. The process could be repeated to recursively generate new hierarchical levels as illustrated in Figure. Conversely, a system is said to have

---

*Strictly speaking, Lie’s work is not limited to classical mechanics and can be applied to any differential equation.*
A Method for Inferring Hierarchical Dynamics in Stochastic Processes

Fig. 1. Three modes of a dynamical system. A schematic illustration of how recursive modularity builds up hierarchies. There are three interacting modules in (c) that each contains interacting modules from (b) that each in turn contains interacting modules from (a). Given the dynamics of the modules in (a), how can we derive the existence and dynamics of the larger modules in (b) and (c)?

Fig. 2. A transition \( s_t \to s_{t+1} = T_\Sigma(s_t) \) is decomposed into a transition in the Wreath product \( (X, T_X)((Y, T_Y)((Z, T_Z)) \). The state \((z_t, y_t, x_t)\) is mapped to \((z_{t+1}, y_{t+1}, x_{t+1}) = (z_t \cdot f_Z(y_t, x_t), y_t \cdot f_Y(x_t), x_t \cdot f_X)\) by the transformation \((f_Z, f_Y, f_X)\), where \( f_Z : Y \times X \to Z \), \( f_Y : X \to Y \) and \( f_X \in T_X \), and where dot denotes group operation. See [14] for further details. We may project away \( z_t \) as \( x_t \) and \( y_t \) evolves independently of \( z_t \), and we may project away \( y_t \) together with \( z_t \) since they are slaved by \( x_t \). We may not, on the other hand, project away \( y_t \) by itself since it influences the evolution of \( z_t \).

Hierarchical structure if it can be deconstructed through recursive decomposition of modular components.

It is natural to invoke an operation that, in a categorical sense, carries the structure of a product: the Wreath product, see Figure 2. Associated with the Wreath product, there exists a quotient operator, or a projection, which in essence defines divisibility of the system. The projection is in general a map from the full
Fig. 3. An original state space $\Sigma$ is reduced to a coarser state space $A$. The map $\Pi$ is a reduction. $T_\Sigma$ denotes the original update dynamics and $T_A$ is the dynamics induced on $A$ by $\Pi$. If $T_A$ is a Markovian dynamics, the diagram commutes and we say that $\Pi$ is a Markov projection.

Fig. 4. (a) Direct product. Both automata can be projected away as they act independently of each other. (b) Semidirect product. Only the left automaton can be projected away since it is dependent on, i.e. is slaved by, the right automaton.

The dynamics of the original system induces a dynamics on the projected state space. In principle any reducing map can be used as a hypothetical projection. However, crucial properties of the original dynamics are often lost in an arbitrary reduction. The most important such property is the Markov property. Only very special reductions, which we call Markov projections, do respect the fundamental character of the dynamics. When this happens, the diagram in Figure 3 commutes. We may say that the dynamics is divisible by the quotient used by the projection, e.g. in Figure 4. For a continuous deterministic dynamical system we call a projection

\[ \text{Lower dimensionality in the continuous case and lower cardinality in the finite discrete case.} \]
\[ \text{Normally a projection is also required to be idempotent, i.e. fulfill } P^2 = P. \text{ In our case however, the projection maps between two different spaces and idempotency is not well defined.} \]
\[ \text{Note that determinism is a special case of the Markov property.} \]
that respects the dynamics fiber preserving, although in this paper we focus on
dynamics generated by discrete stochastic processes. The idea behind the method
is to systematically test different reductions to see if they result in a Markovian
dynamics on the reduced state space. If a reduction passes the test we conclude
that it is a proper projection of the state space, i.e. a Markov projection.

The method introduced in this paper is inspired by computational mechanics
[15, 16] that derives optimal predictors of stochastic processes. The predictors,
termed $\epsilon$-machines, are automata whose nodes are equivalence classes, termed causal
states, of observed histories of states (i.e. semi-infinite sequences in the context of
stochastic processes) such that two histories are equivalent if they condition the
same probability distribution of future observed states. Causal states are connected
with transitions that are labeled with the current state of the observed process,
which completes the $\epsilon$-machine. An $\epsilon$-machine is the minimal and maximally efficient
model of the observed process [16] and may in practice be acquired approximately
e.g. from generated time series [18]. $\epsilon$-Machines are, in addition, Markov and one
may use them to infer hierarchical dynamics in terms of causal states.

The dynamics represented by an $\epsilon$-machine operates on the raw micro state space
of the observed process. It is therefore only a reduction if the original system has
less active states than the state space admits. Our method does, in contrast, infer
course grained dynamics on different hierarchical levels. Since inferred dynamics
in our case by definition exhibits the Markov property, its finite state automaton
representation is the minimal one and equivalent, subject to converting nodes to
transitions to the $\epsilon$-machine of the same coarse grained dynamics.

2. Markov projections

We will now describe our method in more detail. Before continuing, the reader may
want to consult the appendix for a brief review of the concepts that are central to
our approach and to see our use of notation.

2.1. General idea

Say that you have a symbol sequence $s$ that has been generated by a stochastic
process over some state space $\Sigma$:

$$s = (..., s_{t-1}, s_t, s_{t+1}, ...), \ s_\tau \in \Sigma.$$  \hspace{1cm} (1)

You wish to determine if the process exhibits hierarchical dynamics according to
the diagram in Figure and, if so, the nature of this hierarchy. To do this one may

\footnote{The idea of optimal predictors has been introduced on many different occasions, within different
contexts. See [17] for references and details.}

\footnote{For the reader inclined to compare our approach and computational mechanics in more detail,
it may be useful—in order to avoid initial confusion—to note that the states of a stochastic
process label the transitions of an $\epsilon$-machine, whereas they label the nodes in our automaton
representation.}
Fig. 5. The original symbol sequence \((..., s_{t-1}, s_t, s_{t+1}, ...)\) is projected onto a symbol sequence \((..., \tilde{s}_{t-1}, \tilde{s}_t, \tilde{s}_{t+1}, ...)\). Find a projection \(\Pi_A^\Sigma\) such that the imposed dynamics \(T_A\) has the Markov property. Then a state \(\tilde{s}_\tau\) only depends on its previous state \(\tilde{s}_{\tau-1}\).

use a general procedure in the following. Systematically examine the set of possible partitions of \(\Sigma\). For each examined partition \(A\):

1. Map \(s\) onto a new symbol sequence \(\tilde{s}\) and the corresponding process \(T_A\) with the projection \(\Pi_A^\Sigma: \Sigma \to A\),

\[
\tilde{s} = (... \\tilde{s}_{t-1}, \tilde{s}_t, \tilde{s}_{t+1}, ...), \quad \tilde{s}_\tau = \Pi_A^\Sigma(s_\tau),
\]

as in Figure 5.

2. Test if the coarse grained sequence \(\tilde{s}\) is constituted by Markovian dynamics.

The Markov property in step (2) may be identified in the realm of information theory as we discuss next.

2.2. Markov property measure

Let \(a_i, i = 1, 2, ..., |A|\), be specific elements in a partition \(A\) (not to be confused with \(s_\tau\), that are variables over elements in \(A\) at certain times \(\tau\)). For each state \(a_i\), let \(X_i\) be a stochastic variable of the past states preceding \(a_i\), and let \(Y_i\) be a stochastic variable of the subsequent state of \(a_i\). If \(X_i\) and \(Y_i\) are independent for all \(a_i \in A\), \(T_A\) is a Markov process. There are a number of different ways to quantify the degree by which, or probability that, two distributions are independent. One common method is the \(\chi^2\) test \([19]\). Its main purpose, however, is to provide the significance of association between two variables, rather than the strength of association that we prefer. Although there indeed are measures of the strength of association based on \(\chi^2\) statistics, e.g. Cramer’s \(V\) and the contingency coefficient \(C\) \([19]\), these are ill-suited for our purposes as the former exhibits discontinuities with varying contingency table size, and as the latter requires tables with an equal number of rows as columns. In addition do the measures lack direct interpretations. Instead do we use the mutual information \(I(X_i; Y_i)\) between \(X_i\) and \(Y_i\) as a measure of dependence with respect to state \(a_i\). This measure grants clear-cut interpretations and, naturally, enables us
to use tools from information theory. The mutual information is defined as
\[ I(X_i; Y_i) = H(X_i) + H(Y_i) - H(X_i, Y_i), \]
where \( H(V) \) is the Shannon entropy
\[ H(V) = -\sum_{v \in \mathcal{V}} P(V = v) \log_2 P(V = v) \]
of a stochastic variable \( V \) drawn from \( \mathcal{V} \), and where \( H(U, V) \) is the analogous entropy of the joint distribution of two stochastic variables \( U \) and \( V \). \( I(X_i; Y_i) \) is the information one gains when replacing the separate distributions \( P(X_i) \) and \( P(Y_i) \) with their joint distribution \( P(X_i, Y_i) \). \( I(X_i; Y_i) \geq 0 \) with equality when \( X_i \) and \( Y_i \) are independent \[20\]. As a Markov property measure of a partition as a whole, we employ the expected mutual information
\[ \langle I \rangle = \sum_{a_i \in \mathcal{A}} P(a_i) I(X_i; Y_i), \]
where \( P(a_i) \) is probability of state \( a_i \). Similarly, we use the shorthand \( P(..., \tilde{s}_{t-1}, \tilde{s}_t, \tilde{s}_{t+1}) \) to denote the probability that a sequence of stochastic variables \( ... , \tilde{s}_{t-1}, \tilde{s}_t, \tilde{s}_{t+1} \) has the outcome \( ... , \tilde{s}_{t-1}, \tilde{s}_t, \tilde{s}_{t+1} \). Using the definition for conditional probabilities and explicitly representing past and futures as substrings, we can rewrite Eq. \ref{eq:expected_mutual_information} to
\[ \langle I \rangle = -\sum_{\tilde{s}_t, \tilde{s}_{t+1}} P(\tilde{s}_t, \tilde{s}_{t+1}) \log_2 P(\tilde{s}_t, \tilde{s}_{t+1}) + \sum_{\tilde{s}_t} P(\tilde{s}_t) \log_2 P(\tilde{s}_t) \]
\[ - \sum_{... , \tilde{s}_{t-1}, \tilde{s}_t} P(..., \tilde{s}_{t-1}, \tilde{s}_t) \log_2 P(..., \tilde{s}_{t-1}, \tilde{s}_t) \]
\[ + \sum_{... , \tilde{s}_{t-1}, \tilde{s}_t, \tilde{s}_{t+1}} P(..., \tilde{s}_{t-1}, \tilde{s}_t, \tilde{s}_{t+1}) \log_2 P(..., \tilde{s}_{t-1}, \tilde{s}_t, \tilde{s}_{t+1}) \]
\[ = \Delta H_2 - \Delta H_\infty, \]
where \( \Delta H_n \) is the slope of the block entropy of \( \tilde{S} \) at length \( n \) \[21\]. The expected mutual information \( \langle I \rangle \) between past symbols and the next symbol is in other words equivalent to the difference in expected uncertainty of a symbol conditioned on the preceding symbol and the expected uncertainty of a symbol conditioned on all preceding symbols. \( \langle I \rangle = 0 \) if one expects no reduction in the uncertainty of the current state from looking further back than one state.

In practice one acquires approximations of \( X_i \) and \( Y_i \) from a finite symbol sequence and finite history lengths. For each symbol \( a_i \in \mathcal{A} \), we set up a contingency table whose rows are values of \( X_i \) of occurring histories \( \tilde{s}_{t-n}, ..., \tilde{s}_{t-2}, \tilde{s}_{t-1} \) of length \( n \) (drawn from \( \mathcal{A}^n \)) that precede \( a_i \), and whose columns are values of \( Y_i \) of possible subsequent symbols \( \tilde{s}_t \) (drawn from \( \mathcal{A} \)) of \( a_i \). Element \((j, k)\) in the table is then the count that history \( j \) (according to some indexing) is followed by \( k \) and then \( a_k \).

Eq. \ref{eq:expected_block_entropy} generalizes to finite history lengths:
\[ \langle I_n \rangle = \Delta H_2 - \Delta H_{n+2}, \]
3. Examples

Before moving on to algorithmic details, we exemplify our method by employing it to two simple stochastic processes; a finite state automaton and an iterated map.

3.1. A simple automaton

Consider the automaton $T_\Sigma$ over the state space $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ in Figure 6. All possible 15 partitions are evaluated. For $p = 1/2$, there are four Markov projections, including those corresponding to the trivial partitions $A = \{\{\sigma_1\}, \{\sigma_2\}, \{\sigma_3\}, \{\sigma_4\}\}$ and $D = \{d_1\}$, where $d_1 = \Sigma$. The third process, $T_B$, is a bit-flip process over
Fig. 9. Expected mutual information in bits with respect to the 15 projections (of which two examples are labeled) of the process $T_\Sigma$ in Figure 8 as functions of the transition probability $p$. Note that $\langle I_2 \rangle$ of the Markov partitions is not visible since it is of the order of $10^{-6}$ bits. The statistics was collected from 1000 symbol sequences, each of length 1000.

The fourth projection gives $C = \{c_1, c_2\}$, where $c_1 = \{\sigma_1, \sigma_4\}$ and $c_2 = \{\sigma_2, \sigma_3\}$. $T_C$ is such that $c_1$ and $c_2$ are generated with probability $1-p$ and $p$ respectively, and $a_{c_1}$ is always followed by a $c_2$. For $p = 1/2$, this process is referred to as the golden mean process [22]. See Figure 9. The Markov projections are related according to the hierarchy in Figure 8, where the original process $T_\Sigma$ is the direct product of $T_B$ and $T_C$: $T_\Sigma = T_B \times T_C$. The number of Markov projections is dependent on the transition probability $p$. Figure 9 shows $\langle I_2 \rangle$ for the separate possible partitions as functions of $p$. Specifically, at $p = 0$ and $p = 1$, more than four Markov projections exist (e.g. $\{\{\sigma_1, \sigma_4\}, \{\sigma_2, \sigma_3\}\}$ and $\{\{\sigma_1, \sigma_2\}, \{\sigma_3, \sigma_4\}\}$) due to the elimination of transitions in $T_\Sigma$.

### 3.2. An iterated map

The second example concerns a process that belongs to a well-studied class of dynamical systems; iterated maps. They are discrete in time, but operate on continuous phase spaces. Our process, the Roof map, is defined over $[0, 1]$ according to

$$x_{n+1} = \begin{cases} \frac{1}{\alpha} x_n + \alpha, & \text{if } x_n < \alpha \\ \frac{x_n - 1}{\alpha - 1}, & \text{otherwise}, \end{cases}$$

where $\alpha$ is a parameter. We iterate the map and discretize the trajectory with four equally large bins. That is, $x_i \mapsto \sigma_1$ if $x_i \in [0, 1/4]$, $x_i \mapsto \sigma_2$ if $x_i \in [1/4, 1/2]$, etc.
Fig. 10. Roof map for $\alpha = 1/2$.

Fig. 11. Markovian dynamics (a) $T_A$ and (b) $T_B$ of discretized Roof map for $\alpha = 1/2$.

$x_i \mapsto \sigma_3$ if $x_i \in [1/2, 3/4]$ and $x_i \mapsto \sigma_4$ otherwise. For $\alpha = 1/2$, Figure 10 there are five Markov projections that correspond to the partitions

- $A = \{\{\sigma_1\}, \{\sigma_2\}, \{\sigma_3\}, \{\sigma_4\}\}$,
- $B = \{\{\sigma_1, \sigma_2\}, \{\sigma_3\}, \{\sigma_4\}\}$,
- $C = \{\{\sigma_1, \sigma_2\}, \{\sigma_3, \sigma_4\}\}$,
- $D = \{\{\sigma_1, \sigma_2, \sigma_3\}, \{\sigma_4\}\}$ and
- $E = \{\{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}\}$.

$T_A$ and $T_B$ are shown in Figure 11 whereas $T_C$ and $T_D$ both are isomorphic to the automaton in Figure 7(b) with $p = 1/2$. The Markov projections are related according to the hierarchy in Figure 12(b). Note that $E$ is the one-element partition that trivially implies Markovian dynamics. For $\alpha = 1/4$ and $\alpha = 3/4$, Figure 12(a)
and (c) respectively, the Markovian dynamics of finest resolution is not provided by $\mathcal{A}$. Instead, the partitions $\mathcal{F} = \{\{\sigma_1\}, \{\sigma_2, \sigma_3, \sigma_4\}\}$, for $\alpha = 1/4$, and $\mathcal{D}$, for $\alpha = 3/4$, are required, where $T_\mathcal{F}$ and $T_\mathcal{D}$ are isomorphic.

4. Algorithm

Although an exhaustive search through the set of all possible partitions works fine for small $|\Sigma|$, it is an impractical strategy for larger state spaces due to a combinatorial explosion. A state space with $n$ states allows for $B_n = \sum_{k=1}^{n} S(n, k)$ partitions, where $B_n$ is called the Bell number and $S(n, k)$ is the Stirling number. The latter is the number of ways to partition a set with cardinality $n$ into $k$ nonempty subsets. Both $B_n$ and $S(n, k)$ are large numbers (assuming $1 < k < n - 1$),

$$S(n, k) = \frac{1}{k!} \sum_{j=1}^{k} (-1)^{k-j} \binom{k}{j} j^n,$$

which calls for an approach other than sheer brute force.

4.1. Recursive partitions

There are some basic relations between partitions that allow us to avoid an exhaustive search of all possible partitions. When designing our algorithm, we exploit that the mutual information of a partition $\mathcal{A}$, with respect to a state $a_i$, is larger or equal to the mutual information of a partition $\mathcal{B} \ni a_i$ of $\mathcal{A}$, with respect to $a_i$. To see this inequality, consider the following. Let $\Pi$ be a projection from $\mathcal{A}$ to $\mathcal{B}$; $X_i$ and $\hat{X}_i$ stochastic variables of the past states preceding $a_i$ with respect to $\mathcal{A}$ and $\mathcal{B}$; and $Y_i$...
and $\tilde{Y}_i$ stochastic variables of the subsequent states of $a_i$ with respect to $A$ and $B$. Since $Y_i$ is a function whose range is $A$, it may be composed with $\Pi$. Then
\[
\Pi(Y_i) = \tilde{Y}_i.
\] Let $\Phi$ be a function that maps semi-infinite sequences of states from $A$ onto semi-infinite sequences of states from $B$,
\[
\Phi(\ldots, s_{t-3}, s_{t-2}, s_{t-1}) = (\ldots, \Pi(s_{t-3}), \Pi(s_{t-2}), \Pi(s_{t-1})).
\] Then
\[
\Phi(X_i) = \tilde{X}_i.
\] Since a function $g$ of a stochastic variable $V$ cannot increase the information about another stochastic variable $U$, $I(U; V) \geq I(U; g(V))$ (p. 35), we see that
\[
I(X_i; Y_i) \geq I(X_i; \Pi(Y_i)) \geq I(\Phi(X_i); \Pi(Y_i)) = I(\tilde{X}_i; \tilde{Y}_i),
\] as $I$ is symmetric. The inequality (12) is helpful since if we know that an element $b_i$ in a partition $B$ results in high mutual information, we may discard all partitions $A \ni b_i$ that projects onto $B$. This leads us to an algorithm that evaluates partitions in ascending cardinal order, i.e. from coarser to finer partitions.

4.2. Procedure

The components of the algorithm are the sets $S_p$ (previous partition elements) and $S_c$ (current partition elements), and the integer $l$ (level). Left arrow ($\leftarrow$) denotes assignment.

1. Initiation: $S_p \leftarrow 2^\Sigma$ (the power set of $\Sigma$), $S_c \leftarrow \emptyset$ and $l \leftarrow 2$.
2. For every partition of size $l$ composed from elements in $S_p$
   
   (a) Evaluate $\langle I_n \rangle$ and store Markov projections.
   
   (b) Add elements of size $\leq |\Sigma| - l$ with low mutual information $I_n$ to $S_c$.

   If no partitions can be composed or if $l = |\Sigma|$, stop.
3. $S_p \leftarrow S_c$, $S_c \leftarrow \emptyset$ and $l \leftarrow l + 1$. Go to step (2)

Partitions of size $l$ that are evaluated are thus those that can be composed from elements that have implied low mutual information at level $l - 1$. In such a way, partition elements that result in high mutual information are successively discarded as these may not improve due to Eq. (12).

4.3. Possible further pruning

The algorithm may indeed be subject to improvements. If we assume that the dynamics is ergodic then it follows that the cardinality of the process as a whole, i.e. $|\Sigma|$, has the cardinality of its components as divisors. In other words, if we have reason to assume that the time series that we observe includes all possible states
in the state space (in essence a weak ergodicity assumption), then we only need to try partitions with cardinality that divides $|\Sigma|$. The reason for this is straightforward. Assume that we combine two processes $T_A$ and $T_B$ with transition matrices $Q_A$ ($n \times n$) and $Q_B$ ($m \times m$) respectively. If the two processes are combined in a trivial way, i.e. there is no interaction between the sub-processes $T_A$ and $T_B$, then the total process has the transition matrix $Q = Q_A \otimes Q_B$, a $n \cdot m \times n \cdot m$ matrix. Our algorithm works from the other end. We are given a sequence of states from which we can estimate the total transition matrix. Then it is clear that the number of states in an independent sub-process must be a divisor of the number of states in the process as a whole.

More generally, two processes can be combined in a more non-trivial fashion where for example $T_A$ is slaved by $T_B$, i.e. the dynamics of $T_A$ is affected by the state of $T_B$ but not vice versa (e.g. as in Figure 4(b)). The resulting process is then described by a semi-direct product or a Wreath product. We will not go into the details of this algebraic structure here but refer the reader to any introductory text on group theory, for example [23]. The overall conclusion above is however still valid; the number of states in the process $T_B$ must be a divisor of the number of states in the combined process. We can still decrease the number of tested partitions, again, under the assumption that we have global coverage of the state space.

5. Discussion

We have presented a method for inferring hierarchical dynamics from observed time series. Alternatively we may say that the presented scheme detects components of a process that in themselves have Markovian dynamics. The possible usefulness of this, as well as other related methods for decomposing and reducing dynamical systems, is great. Essentially all related methods are, however, either tailored for a limited class of systems, or suffers from high computational complexity. Examples of methods that exhibit the latter are Krohn-Rodes theory, calculations of invariant manifolds, and Markov partitions [24]. Our method is no exception though. The added structure introduced in Section 4 does indeed reduce the number of potential partitions of the state space, but it is not enough to remedy a computational cost that in the worst case scales exponentially with the cardinality of the state space. We believe that this problem is generic to any reduction scheme. In practice one must hope that the process under analysis carries some additional structure that allows us to make further assumptions about which type of projections that make sense to test. For example, if the system is a cellular automaton, it is clear that the reduction should be faithful to the locality and translational invariance of the update rule, i.e. only projections acting locally and independent of the position on the underlying lattice should be evaluated. The reduction of possible partitions from such considerations often recasts the problem into the computationally feasible domain. Note that the tailoring needed is limited to the generation of test partitions, the rest of the algorithm remains unchanged. This feature is appealing from the
implementation standpoint.

We end this presentation by briefly mentioning the types of systems for which we hope that the method is useful. These may be spin systems (renormalization), lattice gases (automated detection of hydrodynamics variables), pattern forming cellular automata (it is speculated, and partially known, that these systems have hierarchies of descriptions), and interaction networks (identification of functional groups).

Acknowledgments

The authors would like to thank Anders Eriksson for helpful discussions and comments. This work was funded by PACE (Programmable Artificial Cell Evolution), a European Integrated Project in the EU FP6-IST-FET Complex Systems Initiative, and by EMBIO (Emergent Organisation in Complex Biomolecular Systems), a European Project in the EU FP6-NEST Initiative.

References

[1] Herbert A. Simon. The architecture of complexity. Proceedings of the American Philosophical Society, 106(6):467–482, 1962.
[2] Cosma Rohilla Shalizi and Cristopher Moore. What is a macrostate? subjective observations and objective dynamics, 2003.
[3] F. Takens. Detecting strange attractors in turbulence. In D. Rand and L.S. Young, editors, Dynamical Systems and Turbulence, Warwick 1980, page 366, Berlin, 1981. Springer.
[4] Norman H. Packard, James P. Crutchfield, J. Doyne Farmer, and Robert S. Shaw. Geometry from a time series. Physical Review Letters, 45:712–716, 1980.
[5] Kenneth Krohn and John Rhodes. Algebraic theory of machines. I. Prime decomposition theorem for finite semigroups and machines. Transactions of the American Mathematical Society, 116:450–464, 1965.
[6] R. Zwanzig. Ensemble methods in the theory of irreversibility. Journal of Chemical Physics, 33:1338, 1960.
[7] C. Gardiner. Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences, volume 13 of Springer Series in Synergetics. Springer, 3 edition, 2004.
[8] H. Risken and T. Frank. The Fokker-Planck Equation: Methods of Solutions and Applications. Springer Series in Synergetics. Springer, 2 edition, 1996.
[9] H. Haken. Synergetics, an Introduction: Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry, and Biology. Springer-Verlag, 3rd rev edition, 1983.
[10] C. Foias, G. R. Sell, and R. Temam. Inertial manifolds for non-linear evolutionary equations. Journal of Differential Equations, 73:309–353, 1988.
[11] A. Gorban and I. Karlin. Invariant Manifolds for Physical and Chemical Kinetics. Lecture Notes in Physics. Springer, 2005.
[12] D. Ruelle. Smooth dynamics and new theoretical ideas in nonequilibrium statistical mechanics. J. Stat. Phys., 95(1/2):393–468, 1999.
[13] J.R. Dorfman. An Introduction to Chaos in Nonequilibrium Statistical Mechanics, volume 14. Cambridge University Press, Cambridge, England, 1998.
[14] Attila Egri-Nagy. Algebraic hierarchical decomposition of finite state automata - A computational approach. PhD thesis, University of Hertfordshire, 2005.
Appendix A. Preliminaries

A.1. Stochastic processes

Here we consider dynamical systems in the form of discrete stochastic processes. They generate bi-infinite sequences $S$ of stochastic variables $S_t$,

$$S = (..., S_{t-1}, S_t, S_{t+1}, ...), \quad (A.1)$$

where each $S_t$ is drawn from a state space $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_n\}$. The subscript $t$ denotes the present state, whereas $t+i$ with negative and positive indices $i$ denote past and future states, respectively. We always assume that a process that generates $S$ is stationary, i.e. that it is invariant under time translation:

$$P(S_{t+n} = s_0, S_{t+n+1} = s_1, ..., S_{t+n+l} = s_l) = \quad (A.2)$$

$$P(S_{t+m} = s_0, S_{t+m+1} = s_1, ..., S_{t+m+l} = s_l), \quad (A.3)$$

for all $n, m \in \mathbb{Z}$, $l \in \mathbb{N}$ and $s_k \in \Sigma$.

A.2. Markov processes

A process is said to have the Markov property if future states are conditionally independent of past states, given the current state:

$$P(S_{t+n} = s_0 | ..., S_{t+n-3} = s_3, S_{t+n-2} = s_2, S_{t+n-1} = s_1) = \quad (A.4)$$

$$P(S_{t+n} = s_0 | S_{t+n-1} = s_1), \quad (A.5)$$

for all $n \in \mathbb{Z}$ and $s_k \in \Sigma$. Processes with this property are termed Markov processes.
A.3. **Partitions and projections**

By a partition $A$ of the state space $\Sigma$, we refer to a set of disjoint subsets whose union is $\Sigma$:

$$A = \{a_1, a_2, ..., a_n\}, \quad (A.6)$$

where $\bigcup_{i=1}^{n} a_i = \Sigma$ and $a_i \cap a_j = \emptyset$ for all $i \neq j$. A projection

$$\Pi^A : \Sigma \to A \quad (A.7)$$

is a function that maps elements of $\Sigma$ onto their respective elements in $A$.

Further, one may recursively partition a partition. A partition $B$ of another partition $A$ is a set of unions of disjoint subsets of $A$, where the union of the elements of $A$ and the union of the elements of $B$ are equal. That is,

$$B = \{b_1, b_2, ..., b_m\}, \quad (A.8)$$

where $b_i = \bigcup a_i, a_i \in A; b_i \cap b_j = \emptyset$ for all $i \neq j$, and $\bigcup_{i=1}^{n} a_i = \bigcup_{j=1}^{m} b_j = \Sigma$. The projection $\Pi^B_A$ from $A$ to $B$ is analog to (A.7).

A.4. **Example**

We conclude the preliminaries with a simple example. Consider the process $T_\Sigma$ given by the finite state automaton in Figure. $T_\Sigma$ acts on the alphabet $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$, is stationary, and fulfills the Markov property since the transition probabilities from each state are independent of previously visited states. There are 15 different possible partitions of $\Sigma$; for example $A = \{\{\sigma_1\}, \{\sigma_2, \sigma_3\}, \{\sigma_4\}\}, B = \{\{\sigma_1\}, \{\sigma_2, \sigma_3, \sigma_4\}\}, C = \{\{\sigma_1\}, \{\sigma_2\}, \{\sigma_3\}, \{\sigma_4\}\}$ and $D = \{\Sigma\}$, where $B$, e. g., is a partition of $A$. The projection $\Pi^B_A$, for instance, gives $\Pi^B_A(\sigma_1) = \{\sigma_1\}, \Pi^B_A(\sigma_2) = \Pi^B_A(\sigma_3) = \{\sigma_2, \sigma_3\}$ and $\Pi^B_A(\sigma_4) = \{\sigma_4\}$.