Alma Mater Studiorum · Università di Bologna

Campus di Cesena
Scuola di Ingegneria E Architettura
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

COMPUTATIONAL MECHANICS:
from theory to practice

Tesi di laurea in Sistemi Intelligenti Robotici

RELATORE:
Chiar.mo Prof. Ing.
Andrea Roli

PRESENTATA DA:
Mattia Barbaresi

Sessione III
Anno Accademico 2016/2017

Abstract

In the last fifty years, computational mechanics has gained the attention of a large number of disciplines, ranging from physics and mathematics to biology, involving all the disciplines that deal with complex systems or processes. With ϵ-machines, computational mechanics provides powerful models that can help characterizing these systems. To date, an increasing number of studies concern the use of such methodologies; nevertheless, an attempt to make this approach more accessible in practice is lacking yet. Starting from this point, this thesis aims at investigating a more practical approach to computational mechanics so as to make it suitable for applications in a wide spectrum of domains. ϵ-machines are analyzed more in the robotics scene, trying to understand if they can be exploited in contexts with typically complex dynamics like swarms. Experiments are conducted with random walk behavior and the aggregation task. Statistical complexity is first studied and tested on the logistical map and then exploited, as a more applicative case, in the analysis of electroencephalograms as a classification parameter, resulting in the discrimination between patients (with different sleep disorders) and healthy subjects. The number of applications that may benefit from the use of such a technique is enormous. Hopefully, this work has broadened the prospect towards a more applicative interest.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rise of dynamical systems theory in the 60s and 70s led to a new confidence that complicated and unpredictable phenomena in the natural world were, in fact, governed by simple but nonlinear interacting entities. New mathematical concepts together with the power of simulations (and symbolic dynamics [Daw et al., 2003]) brought a way to figure out how such phenomena emerged over time and space. With information theory of Shannon [Shannon, 1948] and related studies [Kolmogorov, 1968], it became clear that information (with its dynamics) takes a fundamental role in the context of distributed computation. In addition, the discovery of cahos (and the cahotic phase) [Crutchfield and Young, 1988, Crutchfield and Young, 1989] and the studies on cellular automata, that have been studied from a large number of researches (e.g., [Wolfram, 1983, Grassberger, 1984, Mitchell et al., 1993]) have proved that there is a rich spectrum of unpredictability spanning from periodic to chaotic boundaries. Recently works have investigated the local information dynamics [Lizier, 2010] deepening questions on fundamental operations performed by a system: information storage, information transfer and information modification. These operations are particularly important from a theoretical perspective in complex systems science, where they are the subject of a number of important conjectures about the fundamental nature of distributed computation and its relationship to emergent complex behavior. One focus of such discussion is the dynamics of computation that is the manner in which computations unfold in time and are distributed across space. All these efforts lead to new perspective on complex systems raising questions on how to quantify the unpredictability and the organization of this complex phenomena. Notably, interest rises in the issue of pattern discovery; given data produced by a process, how can one extract meaningful and predictive patterns from it, without knowing in advance the kind of these patterns? What's the intrinsic structure of information flow and how it can be revealed? Generalizing these questions raises the need of some structure or topology for capturing (and for better understanding) distributed computation and its source of complexity [Crutchfield, 2012]. Information theory metrics alone do not capture the computational effort required in modeling these dynamics. It is therefore essential to reconstruct a model that captures the possible ways the computation may undertake.

Building upon the concept of the conditional entropy rate, a formulation resulting from these scenarios named computational mechanics was conceived. This approach attempts to answer the next quantitative questions: (i) how much historical information does a system store, (ii) where is that information stored, and (iii) how is it processed to produce future behavior? Based on the conditional dependences that exist between the intrinsic states of a dynamical process, an algorithm was conceived that reconstructs the structure of such *causal states*. A new observation of the process leads to a transition from a causal state to another. Causal states and their transitions form a deterministic automata; a particular type of hidden Markov model (edge-label emitting machine) called ϵ*-machine* and the entropy of its causal states is the *statistical complexity* of such process [Shalizi et al., 2001].

To date, computational mechanics has contribute to a range of results on theoretical approaches conducted mostly by Jim Crutchfield and his research group. These include Ising models [Feldman and Crutchfield, 2003], molecular systems [Nerukh et al., 2002], crystallography [Varn and Crutchfield, 2004] and so on. On the contrary, there are few application cases; two of which are for security[Whalen, 2010] and for anomaly detection[Xiang et al., 2008]. Given the wide range of fields of application that could exploit this methodology, it is anomalous that there are not yet works or projects to make this approach accessible to everyone; promoting its widespread diffusion. Essentially, what lacks is giving to this theory a more usable and practical accessibility; works that pave the way toward more operative investigations. Especially in robotics, where entropy measures are widely used, not only statistical complexity but the reconstructed models (ϵ-machines) can also be exploited.

So, this thesis attempts to investigate practical aspects of these applications with the aim of broadening the path toward this direction; through explorative experiments [Amigoni and Schiaffonati, 2016], concerning with the finite essence of real data and contingencies that are not met in the theoretical case, to identify the issues most closely related to the practical aspects. With this purpose, a reconstruction algorithm was examined and tested first. Then some experiments were conducted, trying to exploit these models on robots. Finally, this approach was applied on series extracted from electroencephalograms, investigating on sleep diseases, trying to show advantages these models can bring.

The remainder of this dissertation is organized as follows. Chapter 2 introduces the basic theoretical aspects; explaining the background context and the motivations behind this approach (and behind this work). Chapter 3, resumes preliminary tests conducted on the reconstruction algorithm chosen for experiments. These tests have served to comprehend the applicability of this algorithm and, at the same time, to understand its potential. Tests on the logistic map are conducted in this section too. Chapter 4 shows experiments taken on robot simulations regarding the random walk and the aggregation task. In this section focus is posed more on reconstructed models, trying to gather the ways in which they can be exploited. Finally, in chapter 5, are shown the results of the application of this algorithm attempted on electroencephalograms.

Chapter 2

Background

As an introduction of the basic concepts and background, this section considers some of the essential ideas and measures which describe the context and which will be used in the following chapters. The purpose is to give an intuitive notion of the principal ideas that have induced to conceive complexity under these terms and at the same time, that have motivated these studies.

Computational mechanics originates from various contexts including information theory, dynamical system theory, chaos theory, statistical mechanics and physics, with seminal papers from the '60s. Technical reviews can be found in [Crutchfield, 1994, Feldman, 1998, MacKay, 2003]. The aim is to summarize the logical steps that have led to computational mechanics. Why information theory and how computational mechanics follows from it.

Thus, the first section establishes the context and the basis of this dissertation. Additionally, basic notions of computational mechanics are given; causal states, their structure (the ϵ-machine) and resulting measure (statistical mechanics) are introduced. These are the core concepts of the theory, subjected by this study. Finally, to present a picture of the upcoming researches, a glimpse on recent works on this topic is provided.

2.1 Motivations

We thrive in a world made of predictable and unpredictable things. But nature is seldom simple; there are also structures, patterns, organization, complexity. We have an intuitive sense that some things are more "complex" than others. Anyhow, where does this complexity come from? It is concrete or just an illusion? What are patterns? How can they be discovered?

Measures drawn from information theory (like entropy, entropy rate or excess entropy) serve to quantify the randomness and the deterministic part of the process; namely, the complexity of the process. Moreover, it was understood the need of having a concise, unique and comparable representation of complexity. Nonetheless, these metrics are merely values; they do not have the power (as metrics) of capturing patterns and structure. In fact, Markov models propose a structure (chains and hidden models) which also holds adequate mathematical and statistical properties. However these topologies are assumed a priori and then fitted with respect to data series [Shalizi et al., 2002]. Thus, this is not pattern discovery; what is needed is a reconstruction algorithm that infers the structure directly from the analyzed process.

The raise of computational mechanics and its latest mathematical refinements [Shalizi and Crutchfield, 2001] have consolidated the groundedness of statistical complexity and causal states. Its goal is to detect the structure of the information flow in a system. This is pattern discovery, and it can provide unexpected benefits in a vast range of fields and subjects. As well as in my interest.

Seen its wide applicability, what lacks is to make accessible the use of computational mechanics to a more immediate approach. The aim is to solve problems connected with the application of these models, avoiding the theoretical debate.

2.1.1 Why Information?

When we deal with unknown sources of information and uncertainty, we rely on probabilities, indeed statistics. This is what information theory does; it studies the quantification, storage, and communication of information. It was developed by Shannon in 1948 who has provided the notion of entropy [Shannon, 1948] while studying signals and the effect of noise on communication channels, retrieving notion of entropy used in physics for thermodynamic equilibrium (note that every process is a communication channel). Further investigations have led to two important quantities like entropy rate (the irreducible randomness of the system) and excess entropy (a measure of the complexity of the sequence, defined as the *effective measure complexity* in [Grassberger, 1986]).

These metrics provide a natural language for working with probabilities, evade semantic or meaning issues and measure nonlinear correlations. Moreover, they are comparable with almost all systems. They let us know how random a sequence of measurements is or how much information one measurement tells us about another. Unfortunatly, they cannot capture emergent patterns or any sort of structure of the information flow of the system. These metrics effectively provide measures of complexity, however this is not enough. The next example explains this concept.

Consider these (long or infinite) sequences of measurements generated by two distinct processes:

$$\begin{aligned} S_1 = &\quad \ldots 0001110101001101 \ldots \\ S_2 = &\quad \ldots 0101010101010101 \ldots \end{aligned}$$

Suppose that S_1 is generated by a random binary process[1] and S_2 from the period-2 process[2]. Now apply information theory measures to these se-

[1]The random binary process generates random sequences from the alphabet {0,1}

[2]The period-2 process generates binary strings alternating an *1* and a *0*

quences. Since they have the same distribution, then

$$H[S_1] = H[S_2] \quad = \quad -\sum_{i \in A} p_i \times \log_2(p_i) \quad = \quad 1$$

where A is the alphabet, $\{0, 1\}$, and p_i is intended as the probability of the symbol i. Note that, in this case, the entropy is maximized since the probabilities of 1s and 0s are the same ($p_i = \frac{1}{2}$, for each i). However, in both case the entropy indicates that the two processes are equally (and highly) random. This is true in the first case (S_1) but it is not true for the second (S_2). The entropy gives us the measure of uncertainty associated with distributions; so, based on this value we consider both series "equally predictable". The entropy rate of S_1 is 0. This process does not need memory. That is, the process of S_1 is the coin flip. Conversely, the process of S_2 have zero entropy rate but remember the last symbol makes perfectly predictable the series. It is deterministic. If last occurence was 1 the next one must be a 0, if the last was a 0 than the next will be a 1. So, S_1 differs from S_2 but, in this case, we cannot perceive it.

Other measures can capture this difference; for example, if we consider Kolmogorov-Chaitin complexity [Grunwald and Vitányi, 2004] we note that S_1 is "not compressible" (it is random) meanwhile S_2 has a much more compact representation. Anyway, even if we have a measure (like KC-complexity for this case) that can capture some sort of complexity, we cannot capture the way the information is processed yet. The KC-complexity is suited for this example but if we consider more complex processes (not random neither totally deterministic ones) we realize that also this metric is not adequate. Consider for example the even process[3]. It is not "compressible" or "reducible" for Kolmogorov-Chaitin (like the random binary process) but, on the contrary, it has a structured intrinsic process.

In short, all these measures are useful and adequate to quantify various aspects of a system but are contextual to their objectives. So, what's missing?

[3]The even process consists of two states, and generates binary strings where blocks of an even number of *1*s are separated by an arbitrary number of *0*s

Where are patterns? Another step has yet to be done.

2.1.2 The need of a structure

So, information theory measures cannot capture dependency structure. Where is information? How is transferred? Structure, indeed, is taken to be a statement about the relationship between system's components. That is, the more intricate the "correlations" between the system's constituents, the more structured its underlying distribution. For a glance of what is intended for "structure", ϵ-machines of the processes taken above as examples are shown in Figure 2.1.

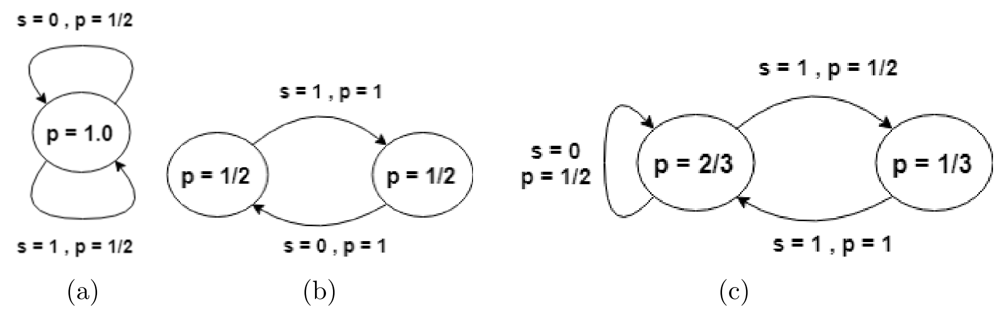


Figure 2.1: ϵ-machines of the processes analyzed above; a)random binary process, b)period-2 binary process and c)the even process. Nodes and arcs are the states and the outcomes of the process, respectively. Each of them has its probability (with label p). The label s on arcs, represents the emitted symbol. Note that the sum of the probability values of the outgoing arcs for each node is always 1.

These are the models reconstructed from series of the processes of the previous examples (S_1, S_2 and the even process). By exploiting these structures, the flow of information becomes easily readable. Further, these structures are unique and minimal. This means that are comparable too. For example, the structure of the random binary process in Figure 2.1 (a), is the same of the coin flip. This is clearly evident; changing the output alphabet ({0,1} with {head, tail}) is enough.

Moreover, the even process, in Figure 2.1 (c), cannot be represented by a finite Markov chain. It can be represented by a finite hidden Markov model, however. In particular, its ϵ-machine provides the most compact presentation.

This effort also provides new perspectives. Markedly, computational mechanics brings the foundation for several related problems; inference versus experimentation, information flow within and between systems, process decomposition and perception-action cycles [Barnett and Crutchfield, 2015].

2.2 Computational Mechanics

Built on measures lent by information theory and the theory of information transmission of Shannon and Kolmogorov (with studies on entropy rate and excess entropy, 1960s), it was developed from efforts in the '70s and early '80s to identify strange attractors in fluid turbulence [Packard et al., 1980] and in the middle of 1980s to estimate the deterministic part of the equations of motion from time series [Crutchfield and McNamara, 1987]. In providing a mathematical and operational definition of structure it addressed weaknesses of these early approaches to discover patterns in natural systems [Crutchfield, 2017].

Unlike the entropy metrics, computational mechanics makes use of the notions of formal computation (from Chomsky hierarchy [Chomsky, 1956]) to conceive a structure for a system's intrinsic information processing. It permits us to see how a system stores, transmits, and manipulates information. It considers the following situation. Given a stationary stochastic process, the semi-infinite "past" (or "history") of the process (at all times up to and including zero) has been observed. Now the "future" (all positive times) has to be predicted as accurately as possible.

The central objects of the theory are the causal states. They are defined as the elements of the minimal partition of the past that is sufficient for predicting the future of the process. The ϵ-machine, containing causal states and their transitions, constitutes the structure that describes (statistically) the behaviour of such process.

2.2.1 Causal States and ϵ-machine

Consider at some time t, the sequence

$$S = \ldots s_{t-2}s_{t-1}s_ts_{t+1}s_{+2}\ldots$$

taking values over some alphabet $A = \{s_1, s_2, \ldots\}$ like the symbolic series of the process (or system) which we would like to examine. How can we predict the next symbol? There is some sort of regularity (in the past) that helps in this prediction? At each time t, the semi-infinite past is $\overleftarrow{S} = \ldots s_{t-2}s_{t-1}$ and the semi-infinite future is $\overrightarrow{S} = s_{t+1}s_{+2}\ldots$. The key step is to identify states (in the past) with conditional probability distribution over future configurations. In building this model, there is no need to distinguish between different past configurations (called past morphs) that give rise to an identical state of knowledge about the future configurations (called future morphs) that can follow it. Two (L-length) "pasts", $\overleftarrow{s}^L_i$ and $\overleftarrow{s}^L_j$, are equivalent if and only if they give rise to the same probabilities over "futures". Note that each causal state is characterized by the probability distribution of its future morphs. It is independent if the system is in state $\overleftarrow{s}^L_i$ or in the state $\overleftarrow{s}^L_j$, the future is the same. So, by using this form of conditional probability, an equivalence relation $\sim$ on the space of all past configurations can be defined as

$$\overleftarrow{s}^L_i \sim \overleftarrow{s}^L_j \Leftrightarrow P(\overrightarrow{s}|\overleftarrow{s}^L_i) = P(\overrightarrow{s}|\overleftarrow{s}^L_j)$$

the classes induced by this equivalence are called causal states. Causal states incorporate all information about the future, that is available in the past. How do we actually organize this information? The ϵ-machine describes the mechanism of prediction. It is defined as a stochastic automaton (i.e., a machine): it has a set S of (causal) states and it is initialized by one of these states according to some initial probability distribution. At each time step t, depending on the current internal state S_t, an output symbol Y_{t+1} from alphabet A and a new internal state S_{t+1} are (stochastically) generated. This is modeled by a transition probability T from internal states to output symbols and internal states: $T : S \rightarrow P(A \times S)$. The ϵ denote that, in general, the measurements may not be direct indicators of the internal states. For example, the symbols may be a discretization of measurements that are continuous in space and/or time.

The model used to represent ϵ-machine is a particular type of hidden markov model; precisely, an edge-emitting machine. This choice is required to ensure the minimality of these models. Unlike HMMs, the topology of such a model (e.g., number of states and transitions) is not set a priori. This evades the modelling dilemma, related to the problem of "innovation" [Crutchfield, 1994].

Thus, in the end, we have obtained a structure (or model). What is missing is its measure.

2.2.2 Statistical complexity

According to Feldman and Crutchfield [Feldman and Crutchfield, 1998], a useful "statistical complexity" must not only obey the ordered-random boundary conditions of vanishing. It must also be defined in a setting that gives a clear interpretation to what structures are quantified.

Like said above, ϵ-machine (and its causal states) represents the intrinsic structure of the system. But what measure can be retrieved from this structure? Coming from information theory, it is natural to adopt its metrics once again. The resulting formula is

$$C\mu = -\sum_{s \in S} p(s) \cdot \log_2(p(s))$$

where S is the set of reconstructed causal state mentioned above. Recalling computational mechanics, to achieve optimal prediction of the system one needs only to remember the causal states. The statistical complexity thus measures the minimum amount of memory needed to perform optimal prediction. That is the entropy calculated on causal states. Statistical complexity measures discount for randomness and thus provide a measure of the regularities present in an object above and beyond pure randomness.

2.3 Recent developments

The latest work [Strelioff and Crutchfield, 2014] focuses on a subset of ϵ-machine. Called Bayesian Structural Inference (BSI), this method relies on a set of candidate topologies (unifilar HMM) to infer the process structure from data series. This eliminates the transient states and ensures the minimality of the model. Inferred models are guaranteed to be ϵ-machines, irrespective of estimated transition probabilities. Anyway, transient states remain useful in some context like anomaly detection or pure exploration of an unknown system or process (or phenomenon).

In [Grassberger, 2017b, Grassberger, 2017a] Grassberger points out that computational mechanics substantially represents the same mechanism of order theories based on uHMMs. On top of that, one of the motivating ideas behind the effort spent in this dissertation is that no matter what kind of reconstruction and other theoretic issues (and debates) this model involves. These kinds of reconstructions can provide models that can be applied and can be useful for almost all disciplines. Swarms in robotics or fluid flows in physics or the brain dynamics in neurosciences and so on. All must deal with the chaos; at least, these methods can assist us.

Last updates and publications ca be found on Crutchfield's Computational Mechanics Archive [Crutchfield, 2018].

Chapter 3

Tests

As in all theoretic models, some assumptions fall apart when passing from theory to practice; in the latter case problems rise due to the finite nature of data collected and the finite memory size, assumed infinite in the former one. To date, application tests are required. Unfortunately, there is no public stable code available and, in fact, there are not so many applications based on these *"machines"*.

Instead of creating a brand new implementation, we preferred to re-use and analyzing existing ones, as they were devised by reliable scholars. Besides, this saved us from exploring different options in the theoretical model. For example, the choice of the method of reconstruction (splitting or merging causal states? [Shalizi et al., 2001]). Thus, preliminary tests are conducted for choosing a reliable reconstruction algorithm. Additionally some "canonical example", like the even process or the logistic map, are observed. This served for understanding what and how the model captures the states and if the inferred structure can be useful for prediction or detection of anomalies.

This step tries to investigate these issues and it aims at understanding the limits of using ϵ-machines on real finite data. Hopefully, this may reveal some insights on applicability of these models.

3.1 Reconstruction algorithms

The main purpose of this step, is to discover a sound and well performing algorithm for reconstruction of ϵ-machines. Once obtained, these models (and metrics) can be used in robotics and IoT. Particularly in the context of swarm of robots, for its complex dynamics; here ϵ-machines can be used to infer or monit (anomaly detection) some behaviours of the swarm or simply to measure and analyze its complexity.

To date it is difficult to find well defined and properly documented reconstruction algorithms. Of the few implementations found, most are ad hoc and without official documentation or publication. Of these works, the last available found is the Brodu's one, referenced in [Brodu, 2011].

The Brodu's code is based upon Shalizi work [Shalizi and Shalizi, 2004] and, according to its author, also improves it. Compared to CSSR[1], the REMAPF[2] algorithm is faster. CSSR is slower because it calculates all the "pasts" for all lengths up to L. On the contrary the REMAPF considers only the "pasts" at given length L. For the same reason REMAPF is more stable; it doesn't have state explosion like the CSSR. Nevertheless, the results obtained are commensurate. Consequently, the REMAPF was chosen because it is faster and facilitates tests and experiments.

Thus, these tests verifies if Brodu's reconstruction algorithm works well and if reconstructed ϵ-machines, once applied, are useful as predictive models. This implementation does not only reconstruct the ϵ-machine; it also allows the application of a utility function that allows reconstructing the *decisional states*, which are a coarser level of the causal states. For the purpose of this thesis, however, this functionality has not been considered.

[1] Causal States Splitting Reconstruction

[2] Reconstruction of ϵ-machine in Predictive Framework

3.2 Preliminary tests

These preliminary tests serves to investigate on applicability issues. Is there a minimun number of occurences that allows reconstruction? What is the ratio between history lengths and number of occurences? Further, are perturbations captured? Is the reconstructed model correct? Series that the model reproduces are statistically the same of those used in input?

By means of symbolization, process (or system) dynamics are represented by strings. Every charachter (or symbol) represent a state (or the outcome) of the process. Thus, the string is the temporal succession of the outcomes of the analyzed process; that is, its symbolic dynamics. As transducers, ϵ-machines are able to (statistically) reproduce the process by which they have been reconstructed. For this reason, some tests have been conducted to verify the statistical correctness of the series reproduced by the model. For these verifications, some of the classical measurements of information theory (e.g. block entropy) have been applied. These measures were also applied to input series and compared to each other. The number of errors (of the even process) reproduced in output has also been taken into account, as the number of symols occurences.

The next section describes the kind of input strings (processes) used for this tests. Then, the procedure adopted and the measures used are explained. The last section shows results obtained.

3.2.1 Bynary series and the even process

The program takes strings as input; they are representations of the system. That is, these strings are sequences of succeding symbols (e.g., measurements) representing the process's dynamic.

For sake of semplicity and without loss of information, tests are conducted mainly on binary strings. Some tests with alpha-numerical series are taken too, but no differences are noted and indeed not mentioned. All possible types of series, corresponding to the types of process that generates them, are considered; from totally ordered to random strings, including periodic and perturbed ones. In particular, series generated from the even process (it consists of two states, and generates binary strings where blocks of an even number of *1*s are separated by an arbitrary number of *0*s) are tested. Despite its apparent simplicity the even process does not correspond to any finite Markov chain, and requires the power of an ϵ-machine to be reconstructed. Some examples from series with more than two symbols are reported too.

More complicated examples could be tested but complexity of the model grows too, as the number of states; then, in this tangle made of iteractions, could be difficult to comprehend dynamics and particularitites. And so, are reported only the characteristic and the more clear cases, ommitting the more complex ones. This is because the attention is placed more on the reconstructed model than on the complexity measure.

3.2.2 Procedure

First of all, Brodu's code was compared with Shalizi's one[3]. This preliminary step serves for verifying if Brodu's work is consistent with the CSSR, taken as benchmark. Resulting reconstructions are substantially the same; despite the authenticity of CSSR, Brodu's work is earlier and (for the author) more stable. So deciding to take the one or the other became only a personal decision.

From the available code[4], two components are used; the EvenProcess and the SymbolicSeries. The former is a simple even process generator. Used for generating some test series. The latter is the algorithm that processes symbolic series to generate the ϵ-machine.

The SymbolicSeries program ingests a string (or a number of strings) as input, and produces the causal states graph (the ϵ-machine). At this point, a piece of code reads this graph and generates from it the output series with the same number of occurrences of the input one. While accomplishing this,

[3]Source: `http://bactra.org/CSSR/`

[4]Source: `http://nicolas.brodu.net/recherche/decisional_states/index.html`

the number of symbols occurrences is collected as well as their probability distribution.

After having generated output another program is run, to apply remaining metrics (entropy, local entropies and block entropy for some L). This code works only on binary strings, and is not applied on alphanumeric ones. It is used only for a preliminary test on statistical verification of the reconstruction, where series with more than two symbols were not accounted.

Various functions are created to read different type of input series and feed the dataset of this algorithm. Ad example, for reading series of integers (with varying precision) or "blocks of chars" as a symbol. Integers can also be treated as strings; nevertheless, the function that scans them has been created in order to be able to decide the precision with which to interpret these data. The case of the "blocks of chars" as symbol is used in the experiments with robots to consider the state of many robots together, as single state, at each step. This has permitted more investigations.

3.2.3 Metrics

To verify the statistical properties of the reconstructed model, classical metrics from information theory have been used. From all other metrics available in this context (joint entropy, conditional entropy, mutual information, and non-statistical ones like Kolmogorov-Chaitin complexity) the most significative one seems to be (theoretically) the block entropy. It is the application of the entropy rate on finite length series with finite memories. That is, entropy rate is the block entropy taken to the limit. The entropy rate (or entropy density or the metric entropy) H_μ measures the unpredictability (in bits per symbol) of the sequence and it is considered one of the most effective measures of complexity.

Metrics used as statistics over series are:

- **the global entropy**
  $H(X) = -\sum_{x \in X} p(X = x) \times \log_2(p(X = x))$

- **the local entropies**, defined for each symbol x,
  $H(x) = -\log_2(p(X = x))$
- **the block-entropy**, it is defined for some blocks length L,
  $H(L) = -\sum_{S_1 \in A} \cdots \sum_{S_L \in A} P(S_1 \dots S_L) \times \log_2(P(S_1 \dots S_L))$.

Number of occurences and basic statistics (the number of occurrences and the number of generated errors) are monitored as well. These measurements will serve only as a verification of the statistical reliability of the series that are produced by the reconstructed model.

3.2.4 Results

The primary question is about the minimal number of occurrences needed for a plausible reconstruction and the trade off with the length of memory (or "pasts length"). This is the first and crucial step; if the reconstruction requires an order of magnitude that is not available in real datasets, no application can be made. We must take into account that the minimum number of occurrences depend also from the size of the vocabulary: if the alphabet is large, more occurrences are needed. From Shalizi's documentation of the CSSR, a rough guide-line is to limit L to no more than $\log N/\log k$, where k is the alphabet size and N is the number of occurrences. This is confirmed by these results.

So these results are presented first. Then, having fixed both memory length and the minimum number of occurrences, test with different input series are shown.

In Table 3.1 are shown errors[5] for each observation length and for each memory length. The number of errors generated in the output series represents how well the model captures the process dynamics. When occurrences are not enough (< 100), the model cannot be reconstructed or even when this happen, the result is not so fitting. This is the meaning of symbol *"-"*; the model was not reconstructed well, or it was not reconstructed at all.

[5] In the even process, an error is a block of an odd number of *1*s (like *"010"* or *"01110"*).

# observations	**memory lengths**					
	1	**2**	**3**	**4**	**5**	**10**
100	-	-	-	-	-	-
1000	~ 80	~ 60	~ 10	0	0	-
10000	~ 900	~ 600	0	0	0	0
100000	~ 8200	~ 5600	0	0	0	0

Table 3.1: Even process: generated errors for various series and memory lengths

When occurrences raise up (> 1000), things go better. The model now reproduces the real dynamics; with short memories, samples taken cannot capture the patterns which have longer periods (e.g., *"11"*). This situation does not change even with a higher number of observations. Differently, with memories of length greater than three errors vanish. These are only some cases from a large number of tries that confirm this trend too.

Another problem that affects the reconstruction, as mentioned more times, is the finite length of pasts. It causes some transient states to raise up. Due to this limit (in this case fixed at $\#symb \times memLen <= 64$), the algorithm cannot distinguish whether the last symbol *1* was emitted from one recurrent state or the other. Logically, it observes that 1/3 of the time the next symbol is a *0* in the data set and 2/3 of the time it is a *1*, matching the proportions of the symbols. Nevertheless, the probability of this transient state is negligible if compared with recurrent's ones and it varies, of course, with the proportion of memory length and the number of observations.

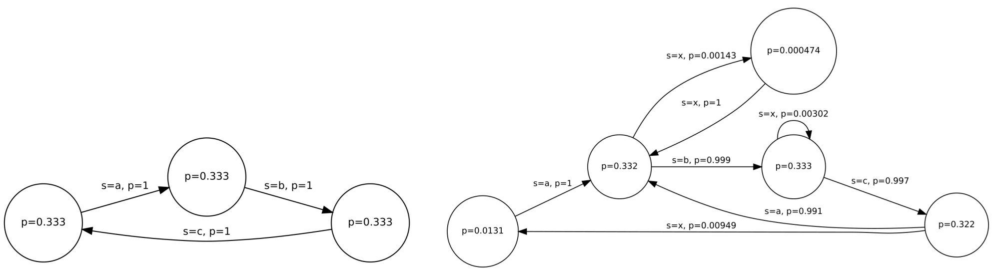
Metrics are respected. Brodu's complexity, calculated on ϵ-machines, resembles the entropy calculated on series. Although they are different sequences, input and output series have the same metrics. When distributions of symbols are respected, the global entropy is clearly respected too. Block entropy, which considers (all) block occurrences, should be a little bit sensitive; instead, like shows Table 3.2, is respected like the other measures. These

series	H(X)	H(*1*)	H(*0*)	H(L)					
				1	**2**	**4**	**8**	**10**	**20**
IN	**0.9169**	0.58	1.59	0.87	0.79	0.73	0.68	0.67	0.53
OUT	**0.9167**	0.58	1.59	0.87	0.79	0.73	0.68	0.67	0.53

Table 3.2: Statistics from input and ouput series (number of observations: 100000)

measurements serve only as a confirmation of the (statistical) correctness of the reconstructed model.

The even process and the other binary strings have been well reconstructed. However, this is the Brodu's benchmark too. Interesting things come up when different types of process are considered.

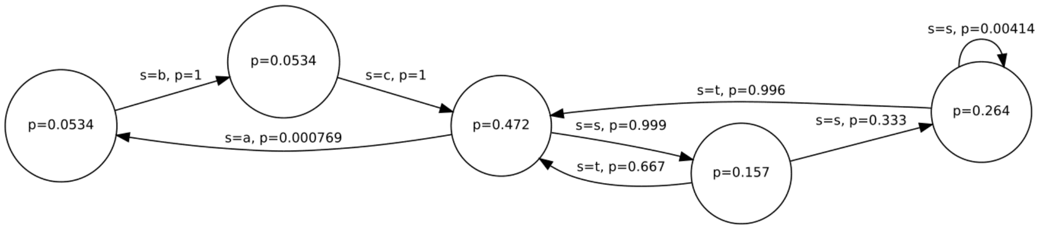


(a) (b)

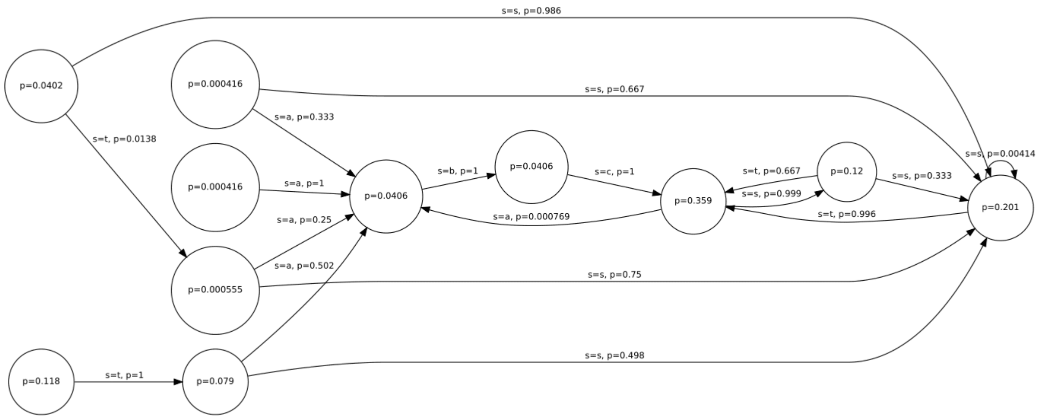
Figure 3.1: Other examples with various input series; a) 3-symbols periodic pattern generating series, in this example *"abc"*, and c) perturbed one with symbol *"x"*.

Figure 3.1 shows the model obtained from a pattern-only reproducing series (in this case: *"abcabcabc..."*) and its perturbations (with symbol *"x"*). When occurrences became consistent, transitions are caught within the model and new (often transient) states raise up. This is well managed by ϵ-machines.

In Figure 3.2, the models is generated with series where some pattern are inserted in arbitrary symbol sequences; in this case the pattern *"abc"* in series



(a)



(b)

Figure 3.2: Other examples with various input series; a) pattern *"abc"* inserted arbitrarily in *"s"* and *"t"* repeating series, b) transient states are shown too. If we look at the probabilities of transient states, we can see how low they are. So, by setting a threshold it would be possible to give the reconstruction the possibility of varying the *"sensitivity"* with respect to causal states inherent in the process.

like *"...ttststsst..."*. Graph including transient states is shown too. The recurrent model captures well the pattern *"abc"* and the principal distribution but if we examine the graph which also includes transient states, we can note that with transient states the model has caught also other interactions (those of perturbations). This means that the model captures all intrinsic dynamics of the process observed but not all are chosen as fundamentals. When we access this deepen world of transient contingencies, we can really understand the internal dynamics of a process; if errors, caused by finiteness, were not present, an even more clear representation could be result. This could pave the way toward anomaly detection methods based on this technique. That is, the recurrent model extracts the main interactions and states, when a more rigorous inspection is needed, transient states could indicate particular anomalies or perturbations that are not caught by the model itself.

Except for particular cases (where observations are not enough, when memory length is not sufficient or when perturbations are enough occurrent to be considered as real transitions, etc.), the model is well reconstructed, and statistics are respected. Furthermore, even with a relatively small dataset, the algorithm seems stable.

3.3 Logistic Map

Preliminary tests were useful to understand the effectiveness of the reconstruction and to verify the basic property of the model. The logistic map is a more realistic example. It is a canonical phenomenon and indeed well studied and known. It is often mentioned as a sample of how complex and chaotic behaviour can arise from very simple non-linear dynamical equations. It falls back under tests because, in this case, the process is well known, indeed we know what to expect. In particular its behaviour can be controlled by parameter r and so various dynamics arise from it and can be analyzed. It is a non-linear difference equation used frequently as model for population growth:

$$x_{t+1} = rx_t(1 - x_t)$$

By varying the parameter r, various behaviors are observed, in particular:

- Between 0 and 1, it converges to 0.
- Between 1 and 3, it will quickly approach the value $\frac{r-1}{r}$, independent of the initial value.
- From 3 increasing beyond 3.56, from almost all initial conditions the population will approach oscillations among 2, 4, . . . , 16, 32, etc. This behavior is an example of a period-doubling cascade.
- At $r \simeq 3.56995$, the onset of chaos, at the ending of the period-doubling cascade. From almost all initial conditions, we no longer see oscillations of finite period. Slight variations in the initial population yield dramatically different results over time, a prime characteristic of chaos (sensibility from initial condition).
- With r between 3.6 and 4, behaviour is chaotic.
- Beyond $r = 4$, almost all initial values eventually leave the interval $[0, 1]$ and diverge.

Therefore, by varying the parameter, all kinds of behaviour are reproducible and then be subjected to these tests. Knowing what the result will be, this test serves to verify, in practice, how the machine is reconstructed and what its topology represents. That is, what the causal states capture, what their interactions are and what they represent. This concedes us the possibility to understand how to read the results and how to interpret them, so it is considered a verification test and not an application test (the last chapter emphasizes this difference).

3.3.1 Finitness and the disequilibrium

In Figure 3.3, the statistical complexity of the logistic map is shown. Five series of 1000 occurrences were generated with initial value $x = 0.53$ and varying the parameter r (2.3, 2.8, 3.2, 3.5, 3.8).

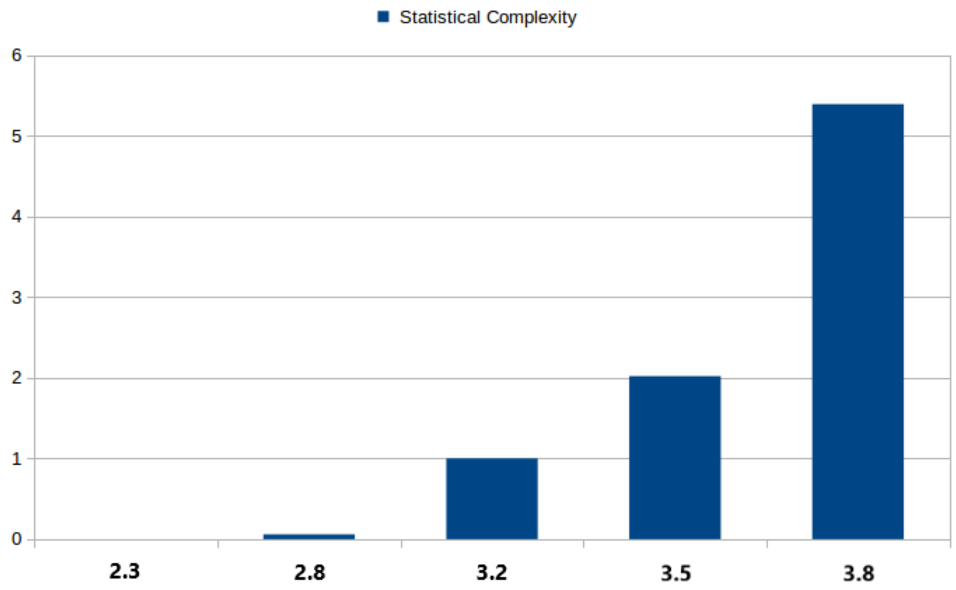


Figure 3.3: Statistical complexity of the logistic map

In Figure 3.3, something went wrong. It was said that a measure of complexity must respect the order-random boundaries; i.e. its value is 0 for both fully deterministic and totally chaotic processes. Instead, it is evident from the histogram that this is not the case. The statistical complexity is initially 0 and respects the logistical map trend for initial values but continues to grow even when the system becomes chaotic, and this is not correct.

So what has happened? This is an example of the problem of the finiteness of occurrences. Typical of when a theoretical model incorporating theoretical limits and theoretical convergences is conveyed to practice. In the theoretical case, when entering a chaotic phase, the signal takes on infinite values. This leads the reconstruction algorithm to create a unique causal state where all the transitions produced by the process start and arrive. This is because distribution becomes normal and equiprobable (IID) and no conditional dependency is found. So the model is reconstructed in the same way as the coin toss; that is, it is presented as totally random. Conversely, in the practical case, due to the accuracy of measuring instruments, machine precision, assumed rounding or symbolization and so on, the algorithm detects dependencies (or patterns) which are reported in the reconstructed model.

This is the most important practical problem. It behaves the opposite of what we expect. Is the reconstruction method wrong? Not necessarily. In this case the problem is the finiteness of data, that's it. It is only a practical aspect. And so, how can we resolve it?

In cases like this, where the process analyzed is basically random but some causal states are identified, the reconstructed model not only tends to have several states but the distributions of these are also substantially comparable. In other words, the reconstructed model resembles that of an IID process. For this reason, a first attempt was to divide the statistical complexity of the model by its number of causal states. Obviously this value can be applied to these cases but, in general, it does not behave well. It always remains to be discovered if given a process, this is completely chaotic or if it is extremely complex (i.e. it hides very long cycles, but not infinite) by observingthe number of its reconstructed states. Nevertheless, this can be a reference measure for some cases.

As shown in the Figure 3.4, the logistical map seems to be one of these cases. In this case, more precise values were used for the logistics map. What the figure above shows is exactly the behavior expected from a measure of complexity. Its complexity grows to r values around 3.5; i.e. as long as the logistic map's behaviour is linear. Then, for values between 3,565 and 3,5689, the edge of chaos. This is the phase where the logistics map reaches its maximum complexity. Here we are in a phase of deterministic chaos. After 3.6 the model follows very simple deterministic rules yet produces apparent randomness. This is chaos: deterministic and aperiodic. That is, behavior becomes completely chaotic, causing our measure of complexity to be 0.

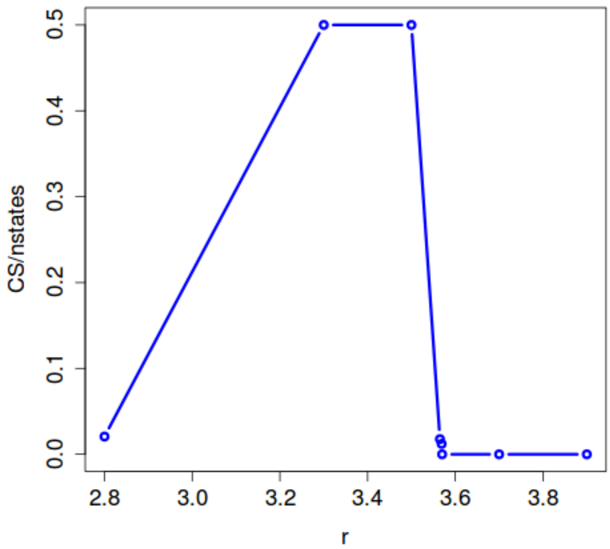


Figure 3.4: Statistical complexity of the logistic map divided by the number of causal states

Considering the number of causal states is therefore not always correct. So we had to find a way to weigh more unbalanced models, those with causal states that were not equiprobable. This is in order to favour systems where the high number of causal states was justified by a real complexity (very different from the equiprobability of the coin toss). Thus, further thoughts and investigations led to consider the disequilibrium of the system [Lopez-Ruiz et al., 1995]:

$$D = \sum_{i}^{N} (p_i - \frac{1}{N})^2$$

Defined in this way, disequilibrium would give an idea of the probabilistic hierarchy of the system. “Disequilibrium” would be different from zero if there are privileged, or more probable, states among those accessible and D = 0 on the limit of equiprobability. According to this definition, a crystal has maximum disequilibrium while the disequilibrium for an ideal gas vanishes.

At this point, the idea was to mediate the number of causal states in order to be able to balance the ratio of the formula in the first attempt. The resulting formula is

$$\frac{Statistical\ Complexity}{\#CS(1-D)}$$

In this way we tried to generalize the behaviour of the formula with the number of causal states, used previously. In the case of the logistic map (but also in the experiments that follow) the behaviour remains substantially unchanged.

r	Statistical Complexity	#CausalState	Complexity / #CS	D	Complexity / (#CS * (1-D))
2.8	0.0626027	3	0.0208676	0.653717	0.0602616
3.3	**1**	**2**	**0.5**	**0**	**0.5**
3.5	**2**	**4**	**0.5**	**0**	**0.5**
3.565	6.82698	423	0.0161394	0.0455293	0.0169093
3.5689	9.42063	762	0.012363	0.00141461	0.0123805
3.57	0	0	0	0	0
3.7	0	0	0	0	0
3.9	0	0	0	0	0

Table 3.3: Values of derived formulae from statistical complexity; with the number of causal states and with the disequilibrium.

In Table 3.3, values from derived measures are shown. These are the values depicted in Figure 3.4. This table is used as summary of these test. The formula with the disequilibrium behaves like that with the number of causal states. The number of causal states of the first row of the table (with r = 2.8) is not zero; this due to the issue on number of occurrences, discussed in the previous chapter. Statistical complexity for r=3.565 and r=3.5689 should be 0 (the edge of chaos is breached). Nevertheless with the adoption of derived formulae, this problem is mitigated. A remark should also be made on the model; the two underlined lines (with r=3 and r=4) in the table represent the period-doubling cascade. It is paramount to note that the number of reconstructed causal states indicates accurately the length of the cycle period.

In conclusion, we saw the tests on the logistics map were also successful. By combining statistical complexity with the number of causal states and disequilibrium, we obtain the desired effect. We are therefore ready to raise the shot and consider experiments in contexts tending towards more concrete cases; namely, robotics simulations.

Chapter 4

Experiments

Previous tests have confirmed the effectiveness of the algorithm. In addition, in doing so, we have deepened our knowledge of reconstructed models and the statistical complexity trends. Even the logistic map case, with some adaptations, has shown that the model captures well the system dynamics and that the measures can be significative.

But how can we use ϵ-machines now? What can we accomplish with such a model in our hands? The first thought was on robot simulations. As computer scientists, it is also the most logical solution; we could acquire not only the ability to generate data at will, but also the possibility of actually trying out the results in real environments. After all, experiments took their course taking us, in the end, to another direction.

So, being available the results of some experiments with robots on two classical exercises such as the *random walk* [Roli et al., 2018] and the *aggregation task* [Francesca et al., 2015], the attempt was to exploit the reconstructed models within these scenarios. As robot controllers, for anomaly detection or automatic design of robots, for example. Moreover, aggregation is a very compelling case. It is not stationary and theoretically all statistical metrics fail in such cases [Liu et al., 2014].

Some preliminaries thoughts, involving memory issue and data mapping, are discussed in the next section. Subsequently, aforementioned tasks and the related results are discussed.

4.1 Preliminaries

We tried to exploit the knowledge of the context (i.e., robot simulations, that generated the data) to understand how ϵ-machines can be useful. The random walk and the aggregation are well-known tasks in robotics. Additionally, they represent theoretical opposite processes; the former is stationary and ergodic, the latter is not-stationary and converges to a standstill situation.

In addition, since these data are related to swarms of robots, some experiments have been attempted considering at each time step the state of the swarm at once rather than the isolated state of each individual robot. Thus, various approaches have been attempted; varying lengths, precisions and mappings of the data for the reconstruction (symbolization).

The number of the occurrences was set to 1200 for all experiments. This is to attain more comparable results; data of aggregation had this dimensionality.

The memory issue

The length of the past and the future has been fixed to one (as previous examples). This because from the first tests made, it was the most suitable one. It makes the reconstruction algorithm more robust, and also makes the results comparable. As for analyzed series with "robot ensembles" as states, discussed below, where are considered one block for the past and one block for the future. In general, the same applies as before; as the length increases, the number of symbols increases too. And the model becomes more complicated. This is related always to the same motif (the same reason that causes a minor explosion of states in REMAPF respect to CSSR). Recalling what said in Chapter 2, a suitable memory length respects

$$L_{mem} \leq \frac{\log(N)}{\log(k)},$$

with k number of symbols and N the number of occurrences. For Shalizi the smaller this ratio, the better. This is one of the factors that influenced the choice of using a 1-length memory. Anyway, some trials with different lengths were carried out during experiments, especially in those cases of doubt, but no particular differences were found.

This issue is the frontier lay between theory and practice. All the trials involved in this thesis concern this question.

Blocks as series

During initial trials, mainly with aggregation data, we realized that analyzing the complete series could hide some interesting dynamics. Since this process is non-stationary, both the transient phase (where robots move, looking for the point of aggregation, trying to avoid collisions) and the final phase (when all robots reach the centre of the arena) were captured in the same model. As the case of bearing an excessive number of symbols compared with the inadequate number of occurrences. After some attempts (considering tests results), the idea was consequently to analyze smaller pieces of series. Since the series did not contain many measurements (1200) they were analyzed using sliding windows of fixed length. Various tests on length of windows were carried out; at the end, 300-occurrences length was chosen, the blocks precisely. A similar approach will be used for electroencephalograms too, in next chapter.

States as ensemble (symbolization)

Having data series arranged with more robots (multi-robot series are both for the "aggregation" and the "random walk"), another mapping of the states was attempted. In this experiment, states of the system are interpreted no longer considering a unique robot but a number of robots as a group. As example, if a robot can perform 3 moves (A, H and R) at a given step, it can be represented with three symbols. A group of two robots, instead, is represented by the symbols AA, HH, RR, AH, AH, HR, etc. (they are exactly $3^2 = 9$). This clearly causes an explosion of the symbols (i.e., system states) of the series and thus an explosion of the states of the model. Accurately, it is n^m with alphabet size $n = |A|$ and m the block length.

First tests are accomplished considering the status of all robots (as blocks) in the multi-robot series; the number of states and symbols is very high, the reconstructed model is also difficult to visualize. The computation time markedly increases. Reconstruction is not completed in 40 robot tests. The model is almost as complex as the series because samples are few to generalize, therefore the algorithm yields nothing more than a track equivalent to the observed series. It is like the previous case of blocks.

4.2 Random walk

In swarm robotics, used mainly as exploration algorithms, random walks are basic building blocks for the individual behaviour that support the emergent collective pattern. Although very simple, they are used in many disciplines. A random walk is a stochastic process which generates a path by steps. At each step the direction is chosen randomly. Implementing swarms where individuals behave in this manner results in an efficient strategy for exploring a given space; especially when there are no (environmental) cues that can drive motion. In this case it is implemented as straight movements and static rotations.

Series contain a lot of measurements in this case but, knowing that the process is stationary, we decided to size them to make results comparable with those of aggregation. Additionally, blocks of more limited size (greater was to much) are been analyzed too. The same approach is used for the aggregation.

4.2.1 Data series

Random walk data are collected on simulations from on a dodecagonal arena and from related ones simulated in an infinite arena. Those for the infinite arena are only for one robot at time. Those on dodecagonal exist for one, ten, twenty and forty robots. Each one comprises 10 simulations with 45, 90 and 180 degrees as turning angle and 10, 20 or 30 steps of S (symbol for straight) that robot does as first action, every cycle. So there are 90 series for each robot. Experiments are 450, overall. Analyzed data are symbolization of the simulated measurements. The input series are generated by robots that can make 3 moves at each step: turn right (R = right), turn left (L = left) and move forward (S = straight). So the series for each robot are of the type:

$$\dots SSRSSLSRRSRSLSSSRSL \dots$$

For the multi-robot series instead, is the column that indicates the single-robot series in time. So for example, a file representing the measurements of a ran simulation with 10 robots comprises series as

$$\begin{gathered} \dots \\ SSRRRRRRSR \\ SSRRRRRRRS \\ SSRSRSSRRS \\ \dots \end{gathered}$$

Reading the first column, we know that the first robot (as the second) has moved forward (S) for almost these three steps. Reading the last column lead to known that after turning right in the first step, the tenth robot has moved forward twice, and so on. The function which read robot ensembles as states treats every row of this file as a unique state. Thus, in the example above are represented also three step of a ten-robot ensemble. Note that this kind of mapping completely alters the state space, indeed the alphabet of the read series.

4.2.2 Results

After having developed several tests it was clear that with 1-symbol memory it was possible to reconstruct the same model for all robot series (regardless of the degree of rotation and the length of the period "straight"). In these cases, the ϵ-machine constructed with the "transient" states matches with that of the "recurrent" states.

Increasing the memory length (therefore, for tests with memory from 2 to 40) reveals many differences between reconstructed models; it becomes burdensome to compare or evaluate them. Here, the concrete problem is that within the same typology the reconstructed models do not seem to be so equal. There are also cases where these models do not capture all transitions (when the period of the series somehow combines with memory, transitions with "L" and "R" may not be present in the reconstructed model). These cases, however, are also compelling; some (in particular those of aggregation) manage to capture the sequence of "S" (to count its occurrences as a finite automaton) for example. Manifestly, the question of "augmented" memories (past and future size) would have to be investigated more thoroughly. The cause, however, may be the Brodu's algorithm, theoretically it not capture all the pasts (it is the question of considering all pasts up to length L).

In summary, two particular cases have been identified. First, the difference between the one robot model in the infinite arena (pure random walk) and the one in the infinite arena (with obstacle avoidance). The second concerns the differences within the finite arena (dodecagonal); between the various behaviours induced by the degrees of rotation and the number of steps of straight movement.

One robot

Initially, the main objective was to obtain robot controllers from the reconstructed ϵ-machines, so they were analyzed first. Models of the infinite arena were compared with the models of the single robot from the finite one. An interesting thing comes up. The model of the finite arena has a transition that the models of the infinite one do not have. Also in multiple series on the dodecagonal arena (10, 20 and 40 robots), the same model is always reconstructed for the single robot. Namely when each column (from block of 10, 20 o 30) was analyzed as if it was a series per se. These models are shown in Figure 4.1. The number of states, in the reconstructed model is always three. By no coincidence they are "left", "straight" and "right". Labels represent probabilities (p) of state and transition and the symbols (s) emitted on transitions.

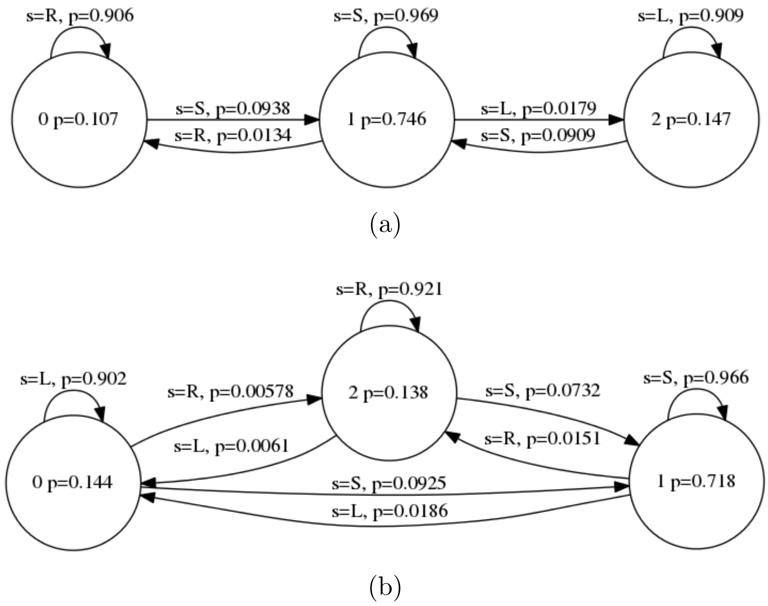
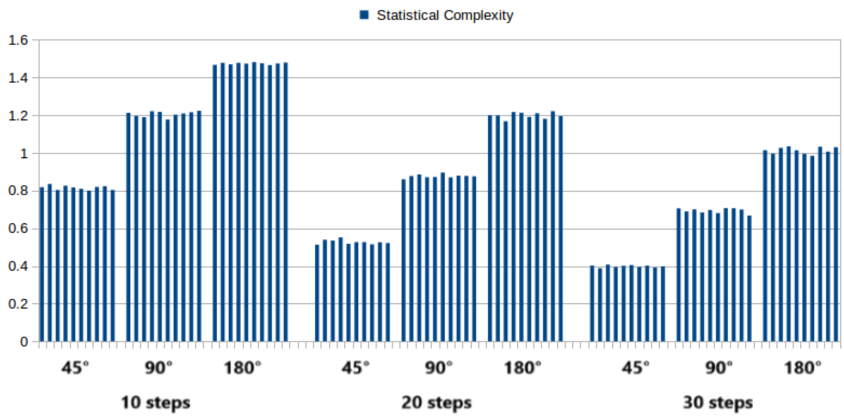


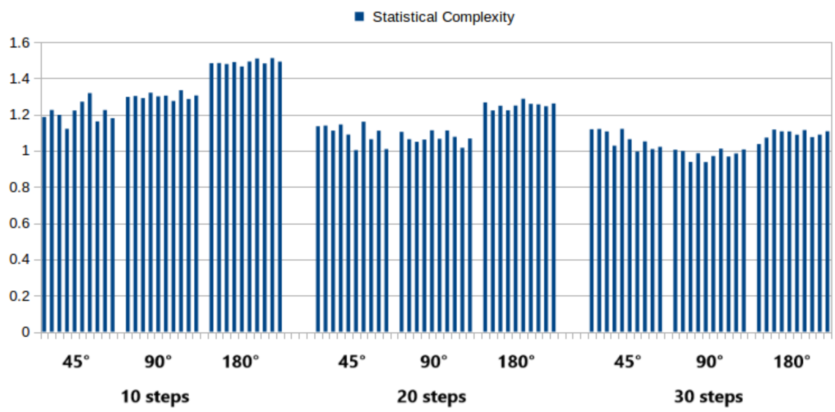
Figure 4.1: ϵ-machine reconstructed from one robot random walk in a)finite arena and b)infinite arena.

Based on the model, seems that the robot never does "inversions" (R→L or L→R actions) in the infinite arena. That is, after turning movement, it always continues straight. This is certainly feasible, but it is a different behaviour than the robot in the finite arena. Obstacle avoidance interferes with direction perhaps because a turn leads to another obstacle or wall or robot, which implies another turn. All series are represented by these two models, only differs for probability values of causal state and transitions. A feature has been found that makes it possible to distinguish them.

Moreover, results of applied measures were analyzed. The statistical complexity for these series does not help. This similarity is also shown in the results of the comparison between healthy and sick subjects, in chapter 4 (Figure 5.1 and Figure 5.2). The resulting signal is oscillating and has no particular phases. Nevertheless in the case of eegs, when it is divided by the number of causal states (or by the number of causal states weighted by disequilibrium) correlation with hypnograms rises. Hence, unlike in the latter case, statistical complexity was dismissed here to focus more on its derived measures.



(a)



(b)

Figure 4.2: Statistical complexity of one robot aggregation behavior in a)infinite arena and b)dodecagonal arena

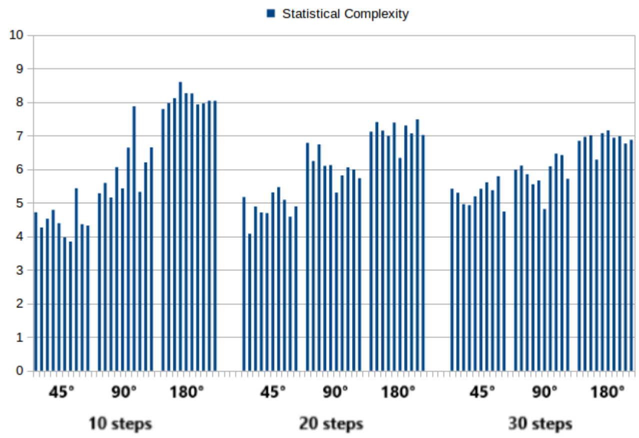
Looking at Figure 4.2, it can be observed that statistical complexity (like entropy rate) is higher when there are few straight steps. The same for the angle of rotation. In accordance with what we expected. A smaller number of straight steps results in fewer steps needed to undertake a rotation, so robots vary their movement frequently. At the same manner, the rotation angle makes rotations more effective also causing a more unexpected trajectory of robots. Namely, these factors affect the complexity of the random walk. However, if the case of the infinite arena confirms this behaviour, in the finite one, this correlation seems almost to disappear. If for one robot in the finite arena (Figure 4.2, b) this ratio is less marked but maintained, when we consider more robots this situation changes.

Robot ensemble

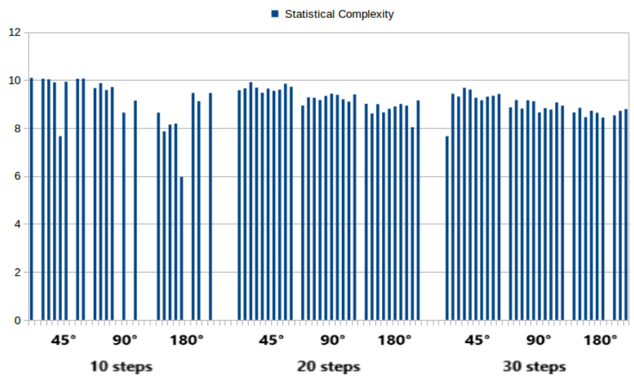
The analysis per single robot has produced interesting outcomes. Given the simplicity of the controllers that direct them, this was a result we could expect. But what if instead of considering the individual robot, the group is considered as a system? What we require to analyze is the complexity of the swarm, its behaviour. Not only the single robot complexity but that of the robots as ensemble. So, instead of reading files per column, a row of the file is considered as a symbol (or state) of the series (or process). Obviously, states exponentially grow with the number of robot (the number of symbols, indeed). Thus reconstructed, the models have a number of states just less than the total. Also this behavior was expected.

In the original series (those with 12.000 occurences), tests on computational time are been conducted[1]: with 5 robots the algorithm takes less than a second, with 7 robots from 1 to 5 seconds, with 10 robots from 10 to 30 seconds..With 20 robots after one hour the algorithm still running. Computation time grows as the number of causal states. Although reducing the number of occurences leads to results, it leads also to the same old problems (memory length and number of symbol).

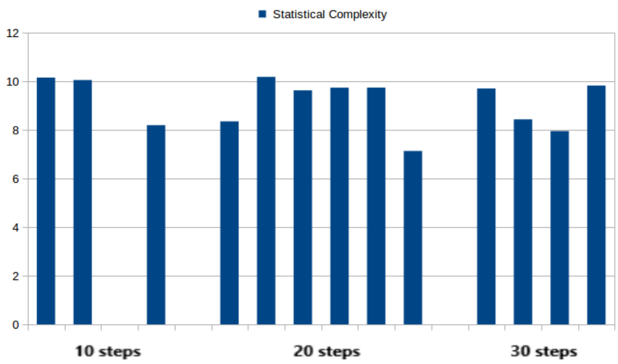
[1]With a portable computer with common performances



(a)



(b)



(c)

Figure 4.3: Statistical complexity of a)10-robot, b)20-robot and c)30-robot simulations, in a dodecagonal arena

Figure 4.3 shows statistical complexites of all the multi-robot series. In this way we are analyzing the behaviour of the swarm, not the individual behaviour. The complexity, as we could have expected, grows and things become inoperable. With 10 robots the behavior tends to vanish but share something with that of 1-robot. With 20 robots it seems almost like a saturated signal, with 40 robots very few models (13 of 90, Figure 4.3, c) are reconstructed; the complexity drops to zero. Various block lengths were tested; typically, the trend is that of those shown. With such a higher number of symbols, the series may be too short. Or simply the system is chaotic? It is the problem of memory length? An examination of these cases would be of interest.

Analysing the series by means of sliding windows did not result in significant cases; narrower segments have shown the same trend as the complete series.

4.3 Aggregation

This task concerns the ability of a swarm to aggregate. As random walk, the aggregation is one of the fundamental behaviors in nature, indeed studied from different disciplines. In robotics both represent computational method inspired by the natural evolution. As already mentioned, unlike the random walk it is not stationary; finally robots stop in the center of the arena.

The cases examined are characterized by the fact that the black spot is not wide enough to contain all the robots, therefore at least one robot will ramain outside the spot, trying to enter the black zone. This situation can be said to be the final attractor of the system.

4.3.1 Data series

For aggregation task, we have data collected on simulations taken on the dodecagonal arena with a full-black circle drawn in the center. Simulations with twenty robots are recorded. Thus, data are organized in twenty file containing series with twenty symbols per row. Also in this case the data analyzed are a mapping of simulated data. As the random walk, each robot can take three behaviours: aggregate (A = aggregation) stop (H=halt) and move away (R = repulsion). So the series are of the type:

$$\ldots$$
$$AAAAAAAAAAAAAAAAAAAA$$
$$ARAAAAARAAAHAARAAAAR$$
$$AARAAAAAAAARAAAAARAA$$
$$\ldots$$

From the first column, the first robot (as the fourth or the fifth) has maintained the aggregation behaviour for all three steps. The same approach of the random walk is used. Thus, each column represents a robot dynamic and rows are considered also as twenty-robot ensemble (or less). Differences with the random walk are evident even just looking at the whole data file; unlike the random walk data which show its homogeneity, data of aggregation show distinct phases. Initially almost all robots move (many A are visible), then some robots reach the black zone and halt (and likely they do not move anymore). Finally almost all robots stop, except the one that continues performing aggregation (A) and repulsion (R) moves indefinitely.

4.3.2 Results

The reconstructed ϵ-machines are almost as complex as the series simply because there are not intrinsic causal states in this case. Those the model captures are local correlations that emerge because of the low probability that other states have too. The repetition of a few sequences of symbols (i.e., in the final phase) is sufficient to reveal a state in the model. And therefore the algorithm yields nothing more than a track equivalent to the observed series; capturing also the final sequence before all robots stop. Approaching the end, moves decrease. The more robots reach the center and stop, the less rotations are performed. This induces these "silent" states to emerge. This case shows a limit of the ϵ-machines. Nonetheless, the reconstructed models represent this peculiarity (of non stationarity). This possibility of capturing also transient phases can be exploited.

Table 4.1 shows result of series analyzed as ensemble of 20 robots. All simulations show low statistical complexity but a (relative) high number of states. Even this case underlines the need for a model (or structure) in addition to the measure; differences with the random walk are not so clear observing only the values on the table. Additionally, envisioning also the comparison with other processes, dividing the complexity by the number of causal states may be results in a more considerable metric then the statistical one.

	file 1	**file 2**	**file 3**	**file 4**	**file 5**	**file 6**	**file 7**	**file 8**	**file 9**	**file 10**
Statistical Complexity	2.87655	2.1266	0	2.81613	2.86984	2.90691	3.0226	1.9702	0.969055	3.03094
# Causal State	23	24	19	20	19	23	23	21	22	23
Complexity / #CS	0.125068	0.0886085	0	0.140807	0.151044	0.126387	0.131417	0.093819	0.0440479	0.13178

	file 11	**file 12**	**file 13**	**file 14**	**file 15**	**file 16**	**file 17**	**file 18**	**file 19**	**file 20**
Statistical Complexity	3.06821	0.773707	2.49757	2.19103	2.52661	0	2.96153	3.13838	1.81111	2.32213
# Causal State	23	20	24	24	18	23	19	25	21	21
Complexity / #CS	0.1334	0.0386853	0.104065	0.0912931	0.140367	0	0.15587	0.125535	0.0862434	0.110578

Table 4.1: Results of aggregation

Things change when these series are analyzed per blocks (narrower segments). Three phases are identified; an initial phase, when robots start to move in search of the point of aggregation, a chaotic phase, almost all robot are moving (the resulted complexity is zero) and a final phase, when robots begin to stop in the centre. Particularly, the third phase in not always present. A peculiarity of the final phase is the "attractor" that sometimes emerge. At this stage one robot may be situated near the central circle without being able to enter it.

Chapter 5

An application: electroencephalograms

Electroencephalography (eeg) is a method to record electrical activity of the brain. Signals are extracted from polysomnographic recordings and are used especially for sleep disorder diagnoses. Especially in recent years, with the advent of complexity measures, interest rises in applying these metrics to these contexts. In addition, with a glance towards automatic design, through tools such as ϵ-machines, it might be interesting and very useful to automatically distinguish between healthy and cases, for example. Or be able to diagnose a disease by determining an anomaly in the model representative of patients. In addition, it is an interdisciplinary context that finds a vast number of uses and applications, especially in the medical field.

Brain dynamics are similar to the kinds of process analyzed in this dissertation: spindles indicate a transfer of information between the hippocampus and the neocortex. K-complexes are essentially the patterns found in such brain dynamics[Amzica and Steriade, 2002]; their occurrence and their periodicity align with the changes in sleep *architecture* that people experience as they age [De Gennaro and Ferrara, 2003]. They are linked to intelligence too [Fogel and Smith, 2011]. A number of studies involve the analysis of these brain dynamics to evaluate, for example, the disorders of consciousness [de Biase et al., 2014].

Another factor influencing this choice was the data availability. This also thanks to the work in [Devuyst et al., 2011], who made these datasets freely available on the web. Data come from the DREAMS project because are published electroencephalograms of both healthy and sick subjects related to the same channel, so comparable (although these datasets are accounted in distinct works). Moreover, the existing systems are badly adapted to pathological subjects and involve several corrections on the analysis of the eeg signals. The inefficiency of those systems is due to the sleep signals instability, the abundance of artifacts, the difficulty to apply classification criteria. The insufficiency of the sleep's macrostructure analysis (sleep stage analysis), which needs to be assisted by a microstructure analysis [Lab, 2018]. This seems fertile terrain for computational mechanics.

Finally, this context is not at all well known by those who have written this thesis, contrary to how it was in the case of experiments. Thus, this case was also chosen to show accessibility to research that this method offers. Further, having a tool to employ these theories would also withdraw the issue from informatic context.

As for the experiments of the previous chapter, electroencephalograms are analyzed applying sliding windows on series: in this case, because of the volume of data, tests are conducted with blocks of series but not as sliding windows but as consecutive samples of the series. Unlike previous case, a block length of 1000 is used. So each series is divided into blocks of such dimensions. Reconstructing the ϵ-machine in this way makes results perfectly comparable with their hypnograms (sampled every 5 seconds, as the time covered by 1000 observations at 200Hz).

Thus, focus is mainly on two aspects: i)correlations between hypnograms and the statistical mechanics measures and ii)relations between health controls and patients.

Datasets

The first dataset consist of eight excerpts of 30 minutes of the central eeg channel. At the sampling frequency of 200Hz, 100Hz or 50Hz. A segment of 30 minutes of the central eeg channel was extracted from each whole-night recording for spindles scoring. Recordings come from patient with various pathologies: dyssomnia, restless legs syndrome, insomnia and apnoea/hypopnoea syndrome [Devuyst, 2018b]. Each excerpt is a textual file (a series of measurements) containing one column with the values (amplitudes in microvolt) of the central eeg samples. The number of samples depends on the sampling rate corresponding to the excerpt. So we have excerpts of 90.000, 180.000 and 360.000 measurement for 50Hz, 100Hz and 200Hz respectively. Except for 1 and 3, the others will be used for comparison with excerpts of healthy subjects; they have the same sampling rate of the same channel (CZ-A1), at 200Hz.

The second dataset consist of 10 excerpts, all of the same channel (CZ-A1), but coming from healthy subjects [Devuyst, 2018a]. Sample frequency is 200Hz for all excerpts. This database was discovered later. The idea of comparing healthy subjects and patients was born thanks to the discovery of its availability.

For each excerpt, we also have the related hypnogram available, in both databases. In other words, the various patient states (sleep, wake, REM stage) were annotated during the recordings. Hence the first idea of trying to infer these states (that of the subject's hypnogram) from the statistical complexity (and related transformations) of the excerpt.

In this case, the algorithm was modified to read series of integers. This because we could thus have the opportunity to change its accuracy (in reading the values). This permits us to quickly change the mapping done on the values and therefore to make various attempts. The series of healthy subjects have, unfortunately, an accuracy up to the first decimal number, hence much less than the measurements of patients. To make the results comparable, once again, the same precision (of healthy subject's series) is used for the patients of the first database.

5.1 Correlation with the hypnogram

Since these datasets also provide hypnograms, a first attempt was to infer, from the resulting models, the status of the subject (wake, sleep and so on). These hypnograms are integers series. These integers correspond to the sleep stage annotated by the expert:

- 5 correspond to the wake stage,
- 4 is the REM stage,
- 3 ,2, 1, 0 are respectively, sleep stage S1, S2, S3, S4,
- -1 represent sleep stage movement,
- -2 or -3 are unknow sleep stage.

These kinds of measurements have permitted us to easily compare these sleep stages with the related metrics derived from the ϵ-machine reconstruction applied on excerpts.

In this case, graphs did not show great correlations, so it was thought to aggregate the data in averages to have more immediate results. Working with R[1] we utilized it to calculate correlations. The command is cor(x,y) where x and y are series of values. In R this command calculates the Pearson correlation formula (as default). Excerpt 1 and 3 were not considered. Due to the sampling frequency of the measurements they contain less value; respectively 180.000 (100Hz) and 90.000 (50Hz) values.

In Figure 5.1 the correlation between statistical complexity measures and hypnograms of sick subjects is shown. Some excerpts like the fourth, the fifth, and the eighth have no significance. The second and the sixth show

[1] A tool for statistical computing and graphics: `https://www.r-project.org/about.html`

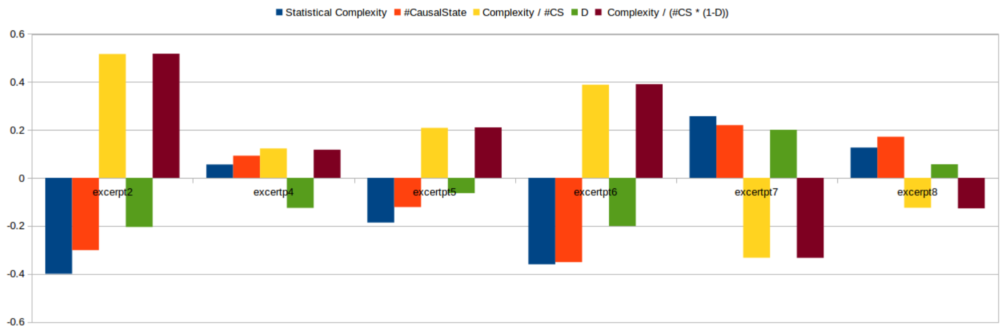


Figure 5.1: Correlations between statistical complexity measures and hypnograms of electroencephalograms of patients

a higher correlation of the measures derived (both have generally the same trend). On the contrary, the seventh show a negative correlation for the same metrics.

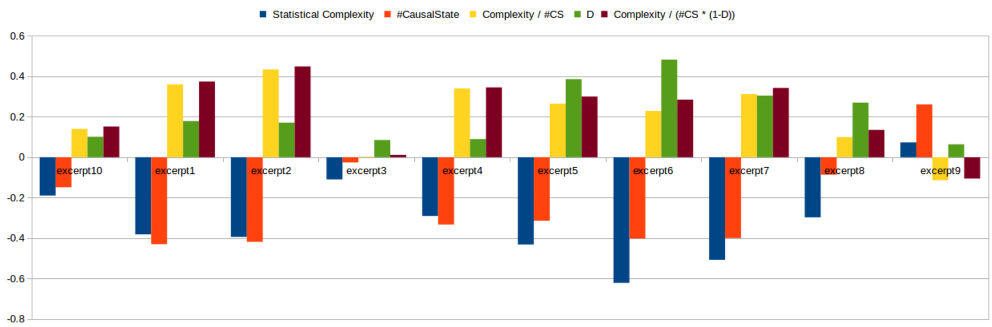


Figure 5.2: Correlations between statistical complexity measures and hypnograms of electroencephalograms of healty controls

In Figure 5.2 are shown correlation (Pearson Formula) between statistical complexity measures and hypnograms of eeg of the healthy subjects. Also here some case are not significant: excerpt 10 and excerpt 3 and excerpt 9. Anyway in all other cases, correlation is more regular respect to previous case. Moreover, it is interesting to note that statistical complexity, generally has a negative correlation even higher than the correlation of the other measures. Negative or not, in this case the trend of correlation seems more stable.

5.2 Patients and healty controls

From the first analysis, we realized that the signal of healthy patients was on average more stable. The statistical complexity was lower as well as the number of the causal states of the reconstructed model. Less evident in models reconstructed from whole series (which also confirms this relationship), in this case (360 blocks for each series) the number of causal states is enough to get the distinction clear.

Values of series from the first database (that of the patients) are truncated at the first decimal digit. This may be a severe discretization. Actually, experiments with various precisions were conducted. Measure trends do not display particular modifications. Nonetheless, on average statistical complexity grows with precision, like the number of causal states. Given the results that are being shown, this would only make clear the difference between patients and healthy controls. We cannot say anything regards precision adopted on healthy subject's series; "raw" measurements had this precision.

File	Statistical Complexity	#CausalState	Complexity / #CS	Disequilbrium	Complexity / (#CS * (1-D))
excerpt1	9.3711	928	0.0100982	0.000762681	0.0101059
excerpt2	10.0658	1609	0.00625593	0.00055217	0.00625939
excerpt3	9.2208	732	0.0125967	0.000551949	0.0126037
excerpt4	9.06425	983	0.009221	0.00141658	0.00923408
excerpt5	9.42968	1144	0.00824273	0.000967739	0.00825071
excerpt6	9.61761	1292	0.00744397	0.000864461	0.00745041
excerpt7	9.70664	1397	0.00694821	0.000816813	0.00695389
excerpt8	9.49774	1135	0.00836806	0.000828859	0.008375
AVG	**9.4967025**	**1152.5**	**0.00864685**	**0.0008451565**	**0.008654135**

Table 5.1: Averages of eeg of patients

In Table 5.1 averages of the measures taken over block of series, from the first database, are shown. These are result from sick subjects. Sich people have a more "noisy" dynamic signal, reflecting in a higher number of symbols per block. Indeed a higher number of states in the model. Thus, complexity grows. However, these results have a linear trend. Averages show that typically 1000 states are reconstructed, approximately; the excerpt 2 slightly differs from other series in the number of causal states. This is confirmed by the statistical complexity too. This is one of the cases where derived measures would be able to say something more; the number of states, given the same statistical complexity, lowers the value of the measure. When many more states are reconstructed, compared to those that the process typically have, they will tend to have a more uniform probability distribution. This is the case with few occurrences and too many symbols. Perhaps due to the particular transient phase of the process.The probabilities of actual causal states do not emerge (or simply they are not in this phase). The model ends up capturing many more states than it normally captures. In these cases, dividing by the number of causal states (consider also disequilibrium), lowers the value of complexity for this particular contingency.

These differences are also well captured by the metrics with the disequilibrium (being really low, it causes few deviations from the formula with the number of causal states). The use of these derived formulae should be better investigated.

File	Statistical Complexity	#CausalState	Complexity / #CS	Disequilibrium	Complexity / (#CS * (1-D))
excerpt1	7.42997	381	0.0195012	0.00509487	0.0196011
excerpt2	7.24375	367	0.0197377	0.00590381	0.0198549
excerpt3	6.67477	243	0.0274682	0.0084673	0.0277028
excerpt4	7.79509	504	0.0154664	0.00413224	0.0155306
excerpt5	7.31137	460	0.0158943	0.00664632	0.0160006
excerpt6	7.30276	302	0.0241813	0.00488707	0.0243001
excerpt7	7.31925	275	0.0266155	0.00456205	0.0267374
excerpt8	6.379	247	0.0258259	0.0284164	0.0265812
excerpt9	6.76451	228	0.0296689	0.00723727	0.0298852
excerpt10	7.17431	289	0.0248246	0.00507237	0.0249512
AVG	**7.139478**	**329.6**	**0.0229184**	**0.00804197**	**0.02311451**

Table 5.2: Averages eeg of healty controls

When the results of healthy subjects were averaged (Table 5.2), the outcome became evident. From the analyses which have been conducted, statistical complexity, as well as the number of causal states of the reconstructed model, seem to have a more stationary trend. So (and this is the most interesting thing) they have averages not comparable with those of sick subjects. In this case, this values are much lower. This means that they are perfectly distinguishable. We have found a satisfactory result.

Although not so scentifically important, it is a result that confirms the usefulness of ϵ-machine and the statistical complexity. In this case derived measures are less significative.

Conclusion

In this thesis, practical aspects of the application of computational mechanics were examined. Tests are conducted to discuss properties and practical limits of the application of this approach and to validate the algorithm chosen for these evaluations. With the experiments on robot simulations, details of the model and the ways it can be exploited are shown. Finally, this method was applied on electroencephalograms as applicative example.

The aim of this dissertation was to explore the possibilities (and contingencies) of applying this approach. It is not important which algorithm has been chosen for these assessments, however, this choice has influenced the results. The Brodu's algorithm is faster, compared to the CSSR. This is due to the fact that the former does not consider all pasts up to the length of memory. This also causes a lower growth rate of the number of causal states when memory length increases (because of lack of occurrences). Nevertheless, results of the two reconstructions are comparable and no substantial differences have been found. Consequently, the fastest was selected.

A first issue raised from the preliminary tests is about the ratio of the number of observations and the size of the alphabet (used to represent the process state space). Usually, when the number of occurrences is adequate with respect to the number of symbols (recalling Shalizi rule) the right model can be reconstructed also with limited data. Conversely, when observations are not enough, the reconstructed model contains causal states with negligible probability or contains sequences of states with a single transition. This is the same situation observed in the case of aggregation; when the process in not stationary. This represents a limit case for such algorithms. However, at the same time these peculiarities of the model can be exploited; that is, the non-stationarity of aggregation is represented in the model and transient states can be analyzed to detect particular *"sub-dynamics"*, as anomalies or errors (e.g., an incorrect procedure performed by a robot). Tests on the logistic map have confirmed that just said above; the generation of these anomalous states, in the reconstructed model, causes the statistical complexity to be non-zero even when the process is chaotic. This is in contrast with the ordered-chaotic boundary conditions of vanishing. However, it can be easily overcome by setting a threshold in the reconstruction algorithm or, for example, combining statistical complexity with other metrics (like the number of causal states or the disequilibrium of the model). More investigations are needed to best evaluate these contingencies.

Experiments have shown advantages of exploiting the models (ϵ-machines) rather than the metrics (statistical complexity). As inferential models, that are reconstructed by means of an algorithm, ϵ-machines can be helpful in autonomous robots; for anomaly detection, image filtering, previsions and so on. In this case, differences were found between the infinite arena and the finite arena (with obstacle avoidance). However, the model reconstruction can be exploited in various ways. Signals from actuators (e.g., wheel motors) and sensors, of a robot, could also be taken into account. For example, encoding the movement in the system's state characterization; mapping the values of the motors and the values of the sensors, means obtaining a model that encodes within its causal states also the sensor-actuator dependencies. Such a model can be used as a new controller because it encodes the intrinsic sense-plan-act cycle of the robot. Being reconstructed by an algorithm, it is done in a totally autonomous way. One of the most important results concerns this question is that mapping (or symbolization) of input data which becomes fundamental; by acting on it, models of different types, capturing different dynamics, can be reconstructed from the same process. As in the example described above. That is, the characterization of symbols from system states becomes a crucial aspect (more than before). This should be analyzed much further.

The attempt on electroencephalograms has shown that results can be obtained, through the application of these models, even in less known contexts. In this case, statistical entropy (and its derived measures) was more useful; these signals are noisy, indeed the reconstructed models are quite complex and not easy to analyze. Nevertheless, more deeper investigations (and experiments) are needed; for example, since eeg signal is noisy, it would be interesting to apply the BSI (Bayesian Structural Inference method, see the section on recent developments). This method avoids many of the problems because de facto the reconstructed models are not sensitive to perturbations (and perhaps noise). It would be interesting to apply it also on data as that of aggregation, that are not stationary. So future works should include that of analyzing these new reconstructions (i.e., BSI) to understand the benefits they would bring and the new properties that can be exploited.

These scenarios show that there is a wide range of possibilities for exploiting ϵ-machines. Almost all research fields, which involve the study of complex system dynamics, can take advantage from computational mechanics (e.g., neurosciences, biology, economics and so on). Given the potential of ϵ-machines and given the attractiveness of computational mechanics, more practical experiments would be undertaken; they would facilitate the use of this method in all contexts, thus encouraging its dissemination.

Bibliography

[Amigoni and Schiaffonati, 2016] Amigoni, F. and Schiaffonati, V. (2016). Explorative experiments in autonomous robotics. In *Model-Based Reasoning in Science and Technology*, pages 585–599. Springer.

[Amzica and Steriade, 2002] Amzica, F. and Steriade, M. (2002). The functional significance of k-complexes. *Sleep medicine reviews*, 6(2):139–149.

[Barnett and Crutchfield, 2015] Barnett, N. and Crutchfield, J. P. (2015). Computational mechanics of input–output processes: Structured transformations and the\epsilon-transducer. *Journal of Statistical Physics*, 161(2):404–451.

[Brodu, 2011] Brodu, N. (2011). Reconstruction of epsilon-machines in predictive frameworks and decisional states. *Advances in Complex Systems*, 14(05):761–794.

[Chomsky, 1956] Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124.

[Crutchfield, 2018] Crutchfield, J. (2018). Computational mechanics archive. `http://csc.ucdavis.edu/~cmg/index.htm`. Online; accessed 02 February 2018.

[Crutchfield, 1994] Crutchfield, J. P. (1994). The calculi of emergence: computation, dynamics and induction. *Physica D: Nonlinear Phenomena*, 75(1):11–54.

[Crutchfield, 2012] Crutchfield, J. P. (2012). Between order and chaos. *Nature Physics*, 8(1):17–24.

[Crutchfield, 2017] Crutchfield, J. P. (2017). The origins of computational mechanics: A brief intellectual history and several clarifications. *arXiv preprint arXiv:1710.06832*.

[Crutchfield and McNamara, 1987] Crutchfield, J. P. and McNamara, B. S. (1987). Equations of motion from a data series. *Complex systems*, 1(417-452):121.

[Crutchfield and Young, 1988] Crutchfield, J. P. and Young, K. (1988). Computation at the onset of chaos. In *The Santa Fe Institute, Westview*. Citeseer.

[Crutchfield and Young, 1989] Crutchfield, J. P. and Young, K. (1989). Inferring statistical complexity. *Physical Review Letters*, 63(2):105.

[Daw et al., 2003] Daw, C. S., Finney, C. E. A., and Tracy, E. R. (2003). A review of symbolic analysis of experimental data. *Review of Scientific instruments*, 74(2):915–930.

[de Biase et al., 2014] de Biase, S., Gigli, G. L., Lorenzut, S., Bianconi, C., Sfreddo, P., Rossato, G., Basaldella, F., Fuccaro, M., Corica, A., Tonon, D., et al. (2014). The importance of polysomnography in the evaluation of prolonged disorders of consciousness: sleep recordings more adequately correlate than stimulus-related evoked potentials with patients' clinical status. *Sleep medicine*, 15(4):393–400.

[De Gennaro and Ferrara, 2003] De Gennaro, L. and Ferrara, M. (2003). Sleep spindles: an overview. *Sleep medicine reviews*, 7(5):423–440.

[Devuyst, 2018a] Devuyst, S. (2018a). The dreams k-complexes database. `http://www.tcts.fpms.ac.be/~devuyst/Databases/DatabaseKcomplexes/`. Online; accessed 02 February 2018.

[Devuyst, 2018b] Devuyst, S. (2018b). The dreams sleep spindles database. `http://www.tcts.fpms.ac.be/~devuyst/Databases/DatabaseSpindles/`. Online; accessed 02 February 2018.

[Devuyst et al., 2011] Devuyst, S., Dutoit, T., Stenuit, P., and Kerkhofs, M. (2011). Automatic sleep spindles detection—overview and development of a standard proposal assessment method. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 1713–1716. IEEE.

[Feldman, 1998] Feldman, D. (1998). Information theory, excess entropy.

[Feldman and Crutchfield, 1998] Feldman, D. P. and Crutchfield, J. P. (1998). Measures of statistical complexity: Why? *Physics Letters A*, 238(4-5):244–252.

[Feldman and Crutchfield, 2003] Feldman, D. P. and Crutchfield, J. P. (2003). Structural information in two-dimensional patterns: Entropy convergence and excess entropy. *Physical Review E*, 67(5):051104.

[Fogel and Smith, 2011] Fogel, S. M. and Smith, C. T. (2011). The function of the sleep spindle: a physiological index of intelligence and a mechanism for sleep-dependent memory consolidation. *Neuroscience & Biobehavioral Reviews*, 35(5):1154–1165.

[Francesca et al., 2015] Francesca, G., Brambilla, M., Brutschy, A., Garattoni, L., Miletitch, R., Podevijn, G., Reina, A., Soleymani, T., Salvaro, M., Pinciroli, C., et al. (2015). Automode-chocolate: automatic design of control software for robot swarms. *Swarm Intelligence*, 9(2-3):125–152.

[Grassberger, 1984] Grassberger, P. (1984). Chaos and diffusion in deterministic cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):52–58.

[Grassberger, 1986] Grassberger, P. (1986). Toward a quantitative theory of self-generated complexity. *International Journal of Theoretical Physics*, 25(9):907–938.

[Grassberger, 2017a] Grassberger, P. (2017a). Comment on" inferring statistical complexity". *arXiv preprint arXiv:1708.04190*.

[Grassberger, 2017b] Grassberger, P. (2017b). Some comments on computational mechanics, complexity measures, and all that. *arXiv preprint arXiv:1708.03197*.

[Grunwald and Vitányi, 2004] Grunwald, P. and Vitányi, P. (2004). Shannon information and kolmogorov complexity. *arXiv preprint cs/0410002*.

[Kolmogorov, 1968] Kolmogorov, A. (1968). Logical basis for information theory and probability theory. *IEEE Transactions on Information Theory*, 14(5):662–664.

[Lab, 2018] Lab, T. (2018). The dreams project. `http://www.tcts.fpms.ac.be/projects/dreams/`. Online; accessed 02 February 2018.

[Liu et al., 2014] Liu, L. F., Hu, H. P., Deng, Y. S., and Ding, N. D. (2014). An entropy measure of non-stationary processes. *Entropy*, 16(3):1493–1500.

[Lizier, 2010] Lizier, J. T. (2010). *The local information dynamics of distributed computation in complex systems*. PhD thesis, School of Information Technologies, The University of Sydney.

[Lopez-Ruiz et al., 1995] Lopez-Ruiz, R., Mancini, H. L., and Calbet, X. (1995). A statistical measure of complexity. *Physics Letters A*, 209(5-6):321–326.

[MacKay, 2003] MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

[Mitchell et al., 1993] Mitchell, M., Hraber, P., and Crutchfield, J. P. (1993). Revisiting the edge of chaos: Evolving cellular automata to perform computations. *arXiv preprint adap-org/9303003*.

[Nerukh et al., 2002] Nerukh, D., Karvounis, G., and Glen, R. C. (2002). Complexity of classical dynamics of molecular systems. *The Journal of chemical physics*, 117(21):9611–9622.

[Packard et al., 1980] Packard, N. H., Crutchfield, J. P., Farmer, J. D., and Shaw, R. S. (1980). Geometry from a time series. *Physical review letters*, 45(9):712.

[Roli et al., 2018] Roli, A., Ligot, A., and Birattari, M. (2018). Complexity measures in automatic design of robot swarms: an exploratory study. In M. Pelillo, I. Poli, A. R. R. S. D. S. M. V., editor, *Italian Workshop on Artificial Life and Evolutionary Computation (to appear)*. Springer.

[Shalizi and Crutchfield, 2001] Shalizi, C. R. and Crutchfield, J. P. (2001). Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of statistical physics*, 104(3):817–879.

[Shalizi et al., 2001] Shalizi, C. R. et al. (2001). *Causal architecture, complexity and self-organization in the time series and cellular automata*. PhD thesis, University of Wisconsin–Madison.

[Shalizi and Shalizi, 2004] Shalizi, C. R. and Shalizi, K. L. (2004). Blind construction of optimal nonlinear recursive predictors for discrete sequences. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 504–511. AUAI Press.

[Shalizi et al., 2002] Shalizi, C. R., Shalizi, K. L., and Crutchfield, J. P. (2002). An algorithm for pattern discovery in time series. *arXiv preprint cs/0210025*.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication, part i, part ii. *Bell Syst. Tech. J.*, 27:623–656.

[Still, 2009] Still, S. (2009). Information-theoretic approach to interactive learning. *EPL (Europhysics Letters)*, 85(2):28005.

[Strelioff and Crutchfield, 2014] Strelioff, C. C. and Crutchfield, J. P. (2014). Bayesian structural inference for hidden processes. *Physical Review E*, 89(4):042119.

[Varn and Crutchfield, 2004] Varn, D. P. and Crutchfield, J. P. (2004). From finite to infinite range order via annealing: The causal architecture of deformation faulting in annealed close-packed crystals. *Physics Letters A*, 324(4):299–307.

[Whalen, 2010] Whalen, S. H. (2010). *Security Applications of the [epsilon]-machine*. University of California, Davis.

[Wolfram, 1983] Wolfram, S. (1983). Statistical mechanics of cellular automata. *Reviews of modern physics*, 55(3):601.

[Xiang et al., 2008] Xiang, K., Zhou, X., and Jiang, J. (2008). Anomaly detection in complex system based on epsilon machine. *International Journal of Systems Science*, 39(10):1007–1016.