



UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR MATHEMATIK



UNIVERSITÄT ZU LÜBECK
GRADUATE SCHOOL FOR COMPUTING
IN MEDICINE AND LIFE SCIENCES

From the Institute of Mathematics
of the University of Lübeck
Director: Prof. Dr. Prestin

Investigating measures of complexity for dynamical systems and for time series

Dissertation
for Fulfillment of Requirements
for the Doctoral Degree of the University of Lübeck
- from the Department of Computer Science -

Submitted by
Valentina A. Unakafova
from Cherkasy, Ukraine

Lübeck 2015



UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR MATHEMATIK



UNIVERSITÄT ZU LÜBECK
GRADUATE SCHOOL FOR COMPUTING
IN MEDICINE AND LIFE SCIENCES

First referee: Prof. Dr. Karsten Keller

Second referee: PD Dr. Bernd Pompe

Chairman: Prof. Dr. Thomas Martinetz

Date of oral examination: 6 of March, 2015

Approved for printing. Lübeck, 10 of March, 2015

Zusammenfassung

Diese Arbeit ist der Untersuchung von Komplexitätsmaßen für dynamische Systeme und für Zeitreihen gewidmet. Die zentralen Gegenstände der gesamten Arbeit sind die Permutationsentropie und ihre Schätzung, die empirische Permutationsentropie, die beide von Bandt und Pompe im Jahr 2002 eingeführt wurden. Die Permutationsentropie wird aus den Verteilungen ordinalen Muster berechnet, und jedes ordinale Muster mit Ordnung d beschreibt die Relationen zwischen den Komponenten eines $(d + 1)$ -dimensionalen Vektors.

Auf der einen Seite untersuchen wir die theoretischen Eigenschaften der Permutationsentropie und einiger anderer Komplexitätsmaße für dynamische Systeme. Auf der anderen Seite studieren wir die Eigenschaften der empirischen Permutationsentropie und vergleichen sie mit anderen praktischen Komplexitätsmaßen bezüglich der Anwendung auf reale Daten, vor allem auf EEG-Daten mit dem Ziel der Detektion von epileptischen Anfällen.

Die wichtigsten Ergebnisse dieser Arbeit sind die folgenden.

- Wir geben hinreichende Bedingungen für die Gleichheit von Permutationsentropie und Kolmogorov-Sinai-Entropie für den allgemeinen Fall und insbesondere für den eindimensionalen Fall.
- Wir entwickeln effiziente Algorithmen zur Berechnung von ordinalen Mustern, empirischer Permutationsentropie und damit verbundenen ordinale-Muster-basierten Komplexitätsmaßen. Die vorgeschlagenen Algorithmen sind schneller als die bekannten Algorithmen und können in Echtzeit angewendet werden.
- Wir vergleichen die Eigenschaften der empirischen Permutationsentropie und der weit verbreiteten Approximate-Entropie und Sample-Entropie. Insbesondere zeigen wir mögliche Probleme und geben Hinweise für die Anwendung dieser Entropien auf reale Daten.
- Wir schlagen ein neues Komplexitätsmaß, die robuste empirische Permutationsentropie, vor, die im Gegensatz zur empirischen Permutationsentropie auch metrische Information gebraucht. Dieses Komplexitätsmaß hat bezüglich der Detektion von epileptischen Anfällen in EEG-Daten während eines Wachzustands bessere Ergebnisse als die empirische Permutationsentropie gezeigt.

Die Arbeit gliedert sich in fünf Kapitel.

Kapitel 1 gibt eine kurze Einführung in die Arbeit.

In Kapitel 2 untersuchen wir, wann die Kolmogorov-Sinai-Entropie und die Permutationsentropie gleich sind. Wir haben keine bejahende Antwort erhalten, ob die Entropien gleich sind, sondern präsentieren interessante und neue Ergebnisse, die als Grundlage für weitere Untersuchungen in dieser Richtung dienen könnten.

Kapitel 3 ist dem Vergleich der empirischen Permutationsentropie mit der Approximate-Entropie und Sample-Entropie gewidmet. Wir diskutieren hier die theoretischen Grundlagen für die drei Entropien. Dann vergleichen wir die praktischen Eigenschaften der Entropien. Außerdem führen wir die (gegen Rauschen) robuste empirische Permutationsentropie ein.

In Kapitel 4 stellen wir effiziente Algorithmen zur Berechnung von ordinalen Mustern, der empirischen Permutationsentropie und von zwei weiteren ordinale-Muster-basierten Komplexitätsmaßen vor. Insbesondere wird eine effiziente Nummerierung für die schnelle Berechnung von ordinalen Mustern mit Bindungen (die dem Fall des häufigen Erscheinens von gleichen Werten in einer Zeitreihe angepasst werden) eingeführt.

Schließlich wenden wir in Kapitel 5 die empirische Permutationsentropie, die robuste empirische Permutationsentropie, die Approximate-Entropie und die Sample-Entropie auf EEG-Daten mit dem Ziel der Detektion von epileptischen Anfällen an. Wir erhalten gute Detektionsergebnisse für die robuste empirische Permutationsentropie. Außerdem untersuchen wir die Wahl der Parameter der empirische Permutationsentropie. Schließlich zeigen wir, dass die kombinierte Verwendung von mehreren praktischen Komplexitätsmaßen ein vielversprechender Ansatz zur Unterscheidung zwischen verschiedenen Zuständen eines System auf der Basis von Messdaten ist.

Abstract

This thesis is devoted to the investigation of complexity measures for dynamical systems and for time series. The central concepts throughout the thesis are the permutation entropy and an estimate of it, the empirical permutation entropy, both introduced by Bandt and Pompe in 2002. The permutation entropy is computed from the distributions of ordinal patterns, each ordinal pattern of order d describes the order relations between the components of $(d + 1)$ -dimensional vector.

On the one hand, we study theoretical properties of the permutation entropy and some other measures of dynamical system complexity. On the other hand, we study the properties of empirical permutation entropy and compare it with other practical complexity measures, when applying to real-world data, and, especially, to EEG data with the aim of epileptic seizure detection.

The main results of this thesis are the following.

- We provide sufficient conditions for equality of permutation entropy and Kolmogorov-Sinai entropy for the general and, especially, for the one-dimensional cases.
- We develop efficient algorithms for computing ordinal patterns, empirical permutation entropy and related ordinal-patterns-based measures. The proposed algorithms are faster than the known methods and can be applied in real-time.
- We compare properties of the empirical permutation entropy with the widely-used approximate entropy and sample entropy. In particular, we indicate possible problems and provide hints for applying the considered entropies to real-world data.
- We propose a new quantity, the robust empirical permutation entropy, which in contrast to the empirical permutation entropy uses also a metric information. This quantity has shown better results than the empirical permutation entropy for epileptic seizures detection in EEG data during the awake state.

The thesis is organized as follows.

Chapter 1 provides a brief introduction to the topic of this thesis.

In Chapter 2 we investigate when the Kolmogorov-Sinai entropy and the permutation entropy coincide. We have not obtained an affirmative answer when the entropies coincide, but we present interesting and new results which could be the basis for further research in this direction.

Chapter 3 is devoted to the comparison of the empirical permutation entropy with the approximate entropy and the sample entropy. We discuss here the theoretical underpinnings for the three entropies. Then we compare practical properties of the considered entropies. We also introduce the robust (to noise) empirical permutation entropy.

In Chapter 4 we present efficient algorithms of computing ordinal patterns, the empirical permutation entropy and two other ordinal-patterns-based measures. In particular, for fast computing of ordinal patterns with tied ranks (adapted to the case of high occurrence of equal values in a time series) the efficient enumeration is introduced.

Finally, in Chapter 5 we apply empirical permutation entropy, robust empirical permutation entropy, approximate entropy and sample entropy to EEG data with the aim of epileptic seizure detection. We demonstrate good detection results for the robust empirical permutation entropy. Besides we study the choice of the parameters for empirical permutation entropy. Finally, we demonstrate that a combined use of several practical complexity measures is a promising approach for discriminating between different states of a system on the basis of real-world data.

Acknowledgement

First of all, I am very grateful to my supervisor Prof. Karsten Keller who gave me an opportunity to work on this thesis. I would like to thank him for his guidance, enthusiasm, advice and patience during all stages of my research work.

I sincerely thank all the staff of the Institute of Mathematics and of the Graduate School for Computing in Medicine and Life Sciences at the University of Luebeck for their support and cooperation. I would like to thank particularly Katja Dau and Olga Schachmatova from the management team of the Graduate School for their encouragement and help with documentary work.

Many thanks to my Romanian friends Rada Popescu, Alexandru Barbu and Irina Burciu for their friendship and support. Finally, I would to thank my family and my husband Anton for their constant understanding and encouragement.

I dedicate this work to my parents, Anna A. Kulikova and Andrei D. Zavizion.

Contents

1	Introduction	1
1.1	Dynamical systems, time series and observables	1
1.2	The model of measuring complexity from real-world data	2
1.2.1	Birkhoff’s ergodic theorem	5
1.2.2	Takens’ reconstruction theorem	5
1.3	Outline of the thesis	7
2	Relationship of permutation entropy and Kolmogorov-Sinai entropy	9
2.1	Preliminaries	11
2.1.1	Kolmogorov-Sinai entropy	11
2.1.2	Lyapunov exponents	12
2.1.3	Ordinal patterns and ordinal partitions	13
2.1.4	Permutation entropy and Kolmogorov-Sinai entropy	14
2.2	Equality of permutation entropy and Kolmogorov-Sinai entropy	15
2.2.1	Piecewise strictly monotone interval maps	15
2.2.2	A criterion for the coincidence in the general case	16
2.3	Comparing ordinal partitions $\mathcal{P}^{\mathbf{X}}(d + n - 1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$	17
2.3.1	The general case	17
2.3.2	The one-dimensional case	19
2.3.3	Conclusions	22
2.4	Comparing ordinal partitions for mixing interval maps	22
2.4.1	Ordinal patterns of order $(d + n - 1)$ and (n, d) -words	24
2.4.2	The “neighboring” partitions $\mathcal{P}(d + 1)_{n-1}$ and $\mathcal{P}(d)_n$	25
2.4.3	The partitions $\mathcal{P}(d)_n$ and $\mathcal{P}(d + n - 1)$	26
2.4.4	Conclusions and open questions	28
2.5	Conclusions	28
2.6	Proofs	29
2.6.1	Proof of Theorem 8	29
2.6.2	Proof of Proposition 11	31
2.6.3	Proof of Theorem 13	32
3	Comparing practical complexity measures	35
3.1	Theoretical underpinnings of approximate and sample entropy	36
3.1.1	Correlation entropy and $H_2(T)$ -entropy	36
3.1.2	Relationship between $H_2(T)$ -entropy, correlation entropy and KS entropy	38

3.1.3	Eckmann-Ruelle’s approximation of KS entropy	39
3.1.4	Estimation of the $H_2(T)$ -entropy	40
3.2	Definitions of the entropies	41
3.2.1	Approximate entropy and sample entropy	41
3.2.2	Empirical permutation entropy	42
3.3	Comparing the practical properties of the entropies	45
3.3.1	Preliminaries	45
3.3.2	Robustness with respect to strictly monotone transformations . .	46
3.3.3	Dependence on the length of a time series	48
3.3.4	Estimation of large complexity	50
3.3.5	Robustness to noise	52
3.3.6	Robust empirical permutation entropy	55
3.3.7	Computational efficiency	57
3.4	Summary	59
3.4.1	The choice between approximate entropy, sample entropy and empirical permutation entropy	59
3.4.2	Hints for using empirical permutation entropy	60
4	Efficient computing of ordinal-patterns-based characteristics	61
4.1	Computing ordinal patterns and the empirical permutation entropy . . .	62
4.1.1	Ordinal patterns	62
4.1.2	The empirical permutation entropy	64
4.2	Efficiently computing the numbers of ordinal patterns	65
4.2.1	Precomputed numbers of ordinal patterns	65
4.2.2	Storage requirements	66
4.3	Efficiently computing the empirical permutation entropy	66
4.3.1	Precomputed values of empirical permutation entropy	67
4.3.2	Round-off error	68
4.3.3	Scheme of the method	69
4.4	Efficiently computing numbers of ordinal patterns with tied ranks	69
4.4.1	Ordinal patterns with tied ranks	70
4.4.2	Precomputed numbers of ordinal patterns with tied ranks	71
4.4.3	Storage requirements	73
4.5	Efficiently computing empirical conditional entropy of ordinal patterns .	73
4.5.1	Efficiently computing $(2, d)$ -words	74
4.5.2	Efficiently computing empirical conditional entropy of ordinal patterns	74
4.6	Comparing efficiency of methods	75
4.7	Conclusions	77
4.8	Supplementary materials	78
4.8.1	Number representation of ordinal patterns with tied ranks	78
4.8.2	Amount of ordinal patterns with tied ranks	79

5	Measuring complexity of EEG	81
5.1	Applying empirical permutation entropy for analysis of EEG data . . .	81
5.1.1	Description of EEG data from The European Epilepsy Database	81
5.1.2	Choice of delay, order, and EEG channel	82
5.2	Detecting epileptic seizures in EEG data by empirical permutation entropy, robust empirical permutation entropy, approximate entropy and sample entropy	86
5.2.1	Detecting epileptic seizures in dependence on vigilance state . . .	87
5.2.2	Detecting epileptic seizures in short-term EEG data	89
5.2.3	Detecting epileptic seizures in long-term EEG data	91
5.3	Discrimination between different complexities of EEG data	94
5.3.1	Description of EEG data from the Bonn EEG Database	94
5.3.2	Discriminating recordings by empirical permutation entropy, approximate entropy and sample entropy	95
5.3.3	Discriminating recordings by empirical permutation entropy computed for different delays	95
5.4	Conclusion	97
A	MATLAB code	99
A.1	Computing empirical permutation entropy by the new method	99
A.2	Computing empirical permutation entropy by the old method	100
A.3	Computing empirical permutation entropy for ordinal patterns with tied ranks	101
A.4	Computing empirical conditional entropy of ordinal patterns	103
A.5	Computing robust empirical permutation entropy	105
A.6	Computing empirical permutation entropy of long-term EEG recording .	106
	Bibliography	111
	Curriculum Vitae	120

Chapter 1

Introduction

The problem of measuring the complexity of an underlying system from observed data arises in various fields of research and in many applications. For example, distinguishing between brain states on the basis of EEG (electroencephalogram) data is an important problem nowadays [MAEL07, Leh08]. A widely-used concept for modeling and measuring complexity of real-world data, in particular, of EEG data, is a *dynamical system* [Sta05, Leh08]. A dynamical system is defined by a pair (Ω, T) , where Ω is a *state space* describing all possible states of the system, and T is a *dynamics* that describes how a point $\omega \in \Omega$ is evolving. *Complexity* of a dynamical system is characterized by unpredictability of its dynamics, i.e. more *complex* dynamical systems are less predictable. In this thesis we investigate quantifiers of dynamical systems complexity and their derived counterparts adapted to real-world data.

First, in Section 1.1 we recall the basic notions from dynamical systems theory. Then in Section 1.2 we present the model, used throughout the thesis, for measuring complexity from real-world data. Finally, in Section 1.3 we outline the content of the thesis.

1.1 Dynamical systems, time series and observables

Given a dynamical system (Ω, T) , in order to introduce a measure on Ω , a sigma-algebra $\mathcal{A}(\Omega)$ is defined, which is a collection of subsets from Ω satisfying the following conditions

- (i) $\emptyset \in \mathcal{A}(\Omega)$,
- (ii) if $A \in \mathcal{A}(\Omega)$, then $\Omega \setminus A \in \mathcal{A}(\Omega)$, and
- (iii) if $A_1, A_2, A_3, \dots \in \mathcal{A}(\Omega)$, then $\bigcup_{n=1}^{\infty} A_n \in \mathcal{A}(\Omega)$.

A pair $(\Omega, \mathcal{A}(\Omega))$ is called a *measurable space*, the sets $A \in \mathcal{A}(\Omega)$ are called *measurable sets*. Given a topological space Ω , the smallest sigma-algebra on it containing all the open sets of Ω is called the *Borel sigma-algebra* $\mathbb{B}(\Omega)$. A *measure* μ is a map that assigns a non-negative real number to each measurable set, i.e. $\mu : \mathbb{B}(\Omega) \rightarrow [0, +\infty)$,

and satisfies the following conditions:

- (i) $\mu(B) \geq 0$ for all $B \in \mathbb{B}(\Omega)$,
- (ii) $\mu(\emptyset) = 0$,
- (iii) for all pairwise disjoint measurable sets $B_1, B_2, B_3, \dots \in \mathbb{B}(\Omega)$,

$$\text{it holds } \mu\left(\bigcup_{n=1}^{\infty} B_n\right) = \sum_{n=1}^{\infty} \mu(B_n).$$

We assume further that the measure μ is a probability measure, i.e. $\mu(\Omega) = 1$, which is just a normalization for a finite μ . A triple $(\Omega, \mathbb{B}(\Omega), \mu)$ is called a *measure space* (or *probability space* in the case of a probability measure).

A map T is called *measure-preserving* or μ -*preserving* if it holds $\mu(T^{-1}(B)) = \mu(B)$ for all $B \in \mathbb{B}(\Omega)$. Throughout the thesis we consider a measure-preserving dynamical system $(\Omega, \mathbb{B}(\Omega), \mu, T)$. This means that Ω is a state space, $\mathbb{B}(\Omega)$ is a Borel sigma-algebra on it, $T : \Omega \leftrightarrow \Omega$ is a $\mathbb{B}(\Omega)$ - $\mathbb{B}(\Omega)$ -measurable μ -preserving map, and $\mu : \mathbb{B}(\Omega) \rightarrow [0, 1]$ is a probability measure. An intrinsic property of a measure-preserving dynamical system is *Poincaré recurrence* (see [ELW11] for a good description), roughly speaking, it means that for any measurable set $B \in \mathbb{B}(\Omega)$ almost every $\omega \in B$ returns to the set B infinitely often.

Given a dynamical system (Ω, T) , the dynamics T provides an *orbit* $\omega, T(\omega), T^2(\omega), \dots$ for each $\omega \in \Omega$. Further we refer to $T^i(\omega)$ with $i \in \mathbb{N}_0$ as i -th *iterate* of ω . In reality, one usually observes only a time series without knowing either the state space of a dynamical system or the dynamics. A dynamical system is related to the observed *time series* by an *observable* $X : \Omega \rightarrow \mathbb{R}$, a function that assigns to each possible state $\omega \in \Omega$ a value from \mathbb{R} . From a mathematical point of view, an observable is a *Borel measurable* map. Given an observable X , one observes from a dynamical system the time series

$$(x_i)_{i \in \mathbb{N}_0} = (X(T^i(\omega)))_{i \in \mathbb{N}_0}$$

from the orbit $(T^i(\omega))_{i \in \mathbb{N}_0}$. In many cases it is reasonable to consider more than one observable in order to obtain more information from the dynamical system, for example, in Chapter 2 we consider a vector $\mathbf{X} = (X_1, X_2, \dots, X_N)$ for $N \in \mathbb{N}$.

1.2 The model of measuring complexity from real-world data

Throughout the thesis we use the following scheme of measuring complexity from real-world data, see Figure 1.1.

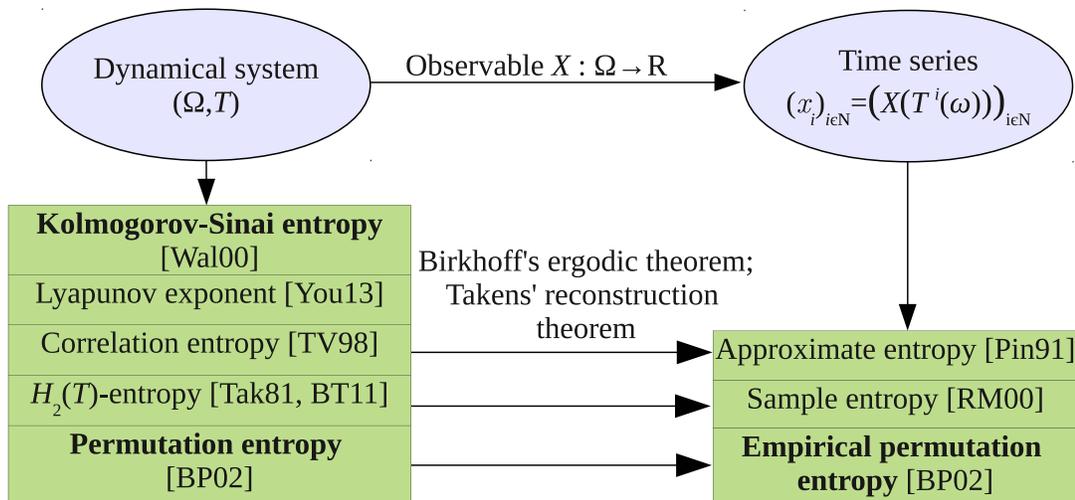


Figure 1.1: A scheme of measuring complexity from time series

Complexity of a dynamical system (Ω, T) is characterized by the following theoretical complexity measures, among which the most important is Kolmogorov-Sinai (KS) entropy.

- The KS entropy¹ measures, roughly speaking, the unpredictability of a dynamics [Wal00]. The KS entropy is an invariant of a measure-preserving dynamical system.
- The Lyapunov exponent measures how two infinitely close to each other points from the state space can diverge under the action of the dynamics. Due to Pesin's theorem [You13] and Ruelle's inequality [Rue78] in certain cases one can compute or bound the KS entropy by the Lyapunov exponent.
- The $H_2(T)$ -entropy and the correlation entropy are introduced in [TV98] and [Tak81, BT11] on the basis of the ideas from [HP83, GP84, ER85].
- The permutation entropy, introduced in [BP02], is a simple concept with a strong relation to the KS entropy [BKP02, Kel12, KUU12, AKM14]. The permutation entropy will be the central concept throughout the thesis.

It is often not easy to estimate these and similar quantities from finite real-world time series. Therefore practical complexity measures adapted for real-world time series, like the approximate entropy [Pin91], the sample entropy [RM00], and the empirical permutation entropy [BP02] are introduced.

Example 1. To motivate the practical complexity measures, in Figure 1.2 we present the values of approximate entropy, sample entropy and empirical permutation entropy computed from EEG data with an epileptic seizure (in gray in the upper plot). The

¹Kolmogorov-Sinai entropy is sometimes also called *metric entropy* and *measure-theoretic entropy*.

EEG has been recorded from channel F4 from a male patient of age 36, the complex partial seizure has been recorded during awake state, EEG data are provided by [Epi14].

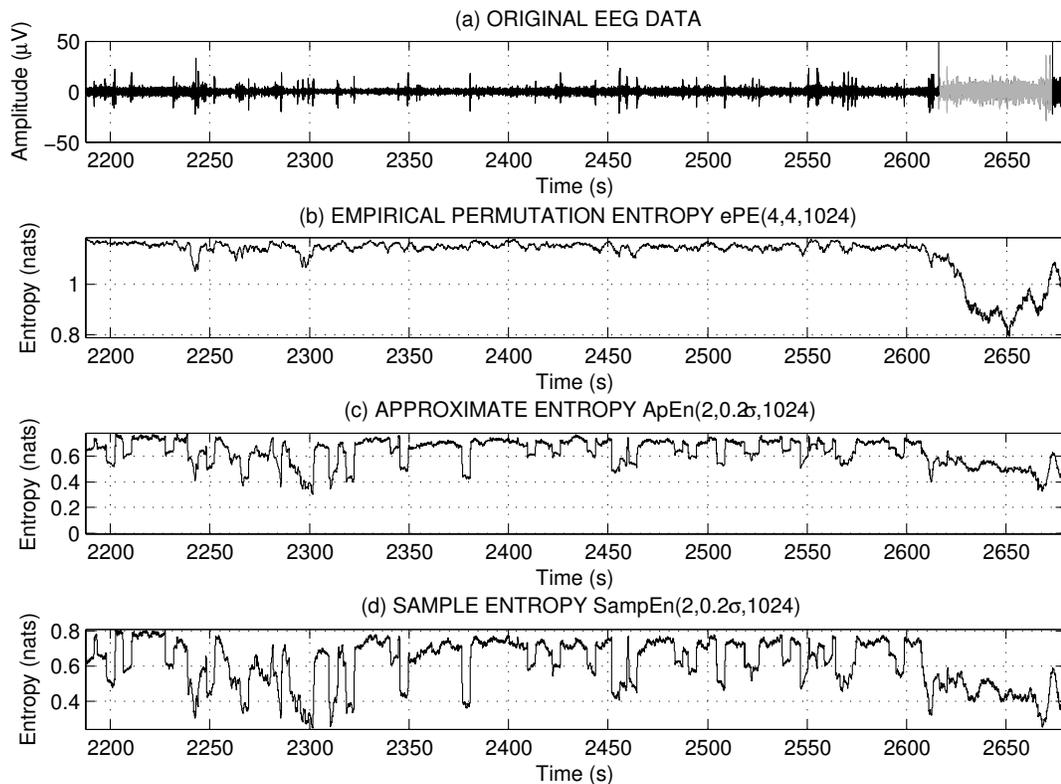


Figure 1.2: The values of approximate entropy, sample entropy and empirical permutation entropy, computed from EEG data, decrease during the epileptic seizure (in gray, upper plot)

The entropies are computed in maximally overlapping sliding windows of 4 s size. One can see that all the entropies reflect the seizure by a decrease of their values (for more details of this experiment we refer to Example 23, p. 88). This is important since detecting epileptic seizures is a relevant problem in biomedical research [LE98, MAEL07, Leh08]. We continue measuring complexity of EEG data by approximate entropy, sample entropy and empirical permutation entropy in Chapter 5.

Roughly speaking, practical complexity measures are related to the theoretical complexity measures by two theorems. Birkhoff’s theorem allows to recover properties of a dynamical system just from one orbit of almost every $\omega \in \Omega$, and Takens’ theorem enables to reconstruct the original dynamics T from observed time series. We consider both theorems in more details in Subsections 1.2.1 and 1.2.2.

1.2.1 Birkhoff's ergodic theorem

We consider here an ergodic map T that provides the possibility to recover properties of the measure-preserving dynamical system just from an orbit of a single ω for almost all $\omega \in \Omega$.

Definition 1. A map T is said to be *ergodic* if for every $B \in \mathbb{B}(\Omega)$ with $T^{-1}(B) = B$ it holds either $\mu(B) = 0$ or $\mu(B) = 1$.

A dynamical system $(\Omega, \mathbb{B}(\Omega), T, \mu)$ with ergodic T is also called an *ergodic dynamical system*. Roughly speaking, an ergodic dynamical system does not split up into “independently” developing systems, i.e. the orbits of almost all points $\omega \in \Omega$ reflect approximately the same properties of the dynamics T .

Now we present Birkhoff's ergodic theorem which is one of the central results in ergodic theory (see [Cho00] for details and proof).

Theorem 1 (Birkhoff's ergodic theorem, [Cho00]). *Given a measure-preserving dynamical system $(\Omega, \mathbb{B}(\Omega), T, \mu)$ and an integrable function f , there exists a function f^* such that $\int_{\Omega} |f^*| d\mu(\omega) < \infty$, $f^*(T(\omega)) = f^*(\omega)$ for all $\omega \in \Omega$ and for almost every $\omega \in \Omega$ it holds*

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} f(T^i(\omega)) = f^*(\omega). \quad (1.1)$$

Furthermore, if T is ergodic, then f^* is constant and for almost every $\omega \in \Omega$ it holds

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} f(T^i(\omega)) = \int_{\Omega} f d\mu(\omega). \quad (1.2)$$

Roughly speaking, according to Birkhoff's ergodic theorem, ergodicity provides a possibility to obtain all information of a distribution of a measure μ from an orbit of almost every point $\omega \in \Omega$. Another important property of dynamical systems, which is stronger than ergodicity, is mixing [Cho00].

Definition 2. T is said to be *mixing* or *strong-mixing* if for every $A, B \in \mathbb{B}(\Omega)$

$$\lim_{n \rightarrow \infty} \mu(T^{-n}A \cap B) = \mu(A)\mu(B).$$

Roughly speaking, mixing implies that future iterates $T^i(\omega)$ of ω become “almost independent” from ω as i increases. We use the mixing property for proofs in Chapter 2.

1.2.2 Takens' reconstruction theorem

Takens' theorem is closely related to the so-called *state space reconstruction*. In order to explain the idea of the state space reconstruction, we consider four different time series in Figure 1.3. (We have taken the idea of such a representation from the instructive and well explained dissertation of S.A. Borovkova [Bor98].) These time series are:

- (1) EEG data recorded from a healthy subject with open eyes [Bon14];
- (2) a sequence of random real numbers;
- (3) a time series $(x_i)_{i \in \mathbb{N}_0}$ observed with $X = \text{id}$ from an orbit of the logistic map $T_{\text{LM}} : [0, 1] \leftrightarrow$ given by $T_{\text{LM}}(\omega) = 4\omega(1 - \omega)$, i.e. $(x_i)_{i \in \mathbb{N}_0} = (T_{\text{LM}}^i(\omega))_{i \in \mathbb{N}_0}$;
- (4) a time series $(x_i)_{i \in \mathbb{N}_0}$ observed with $X = \text{id}$ from an orbit of the tent map $T_{\text{TM}} : [0, 1] \leftrightarrow$ given by $T_{\text{TM}}(\omega) = 2 \min\{\omega, 1 - \omega\}$, i.e. $(x_i)_{i \in \mathbb{N}_0} = (T_{\text{TM}}^i(\omega))_{i \in \mathbb{N}_0}$.

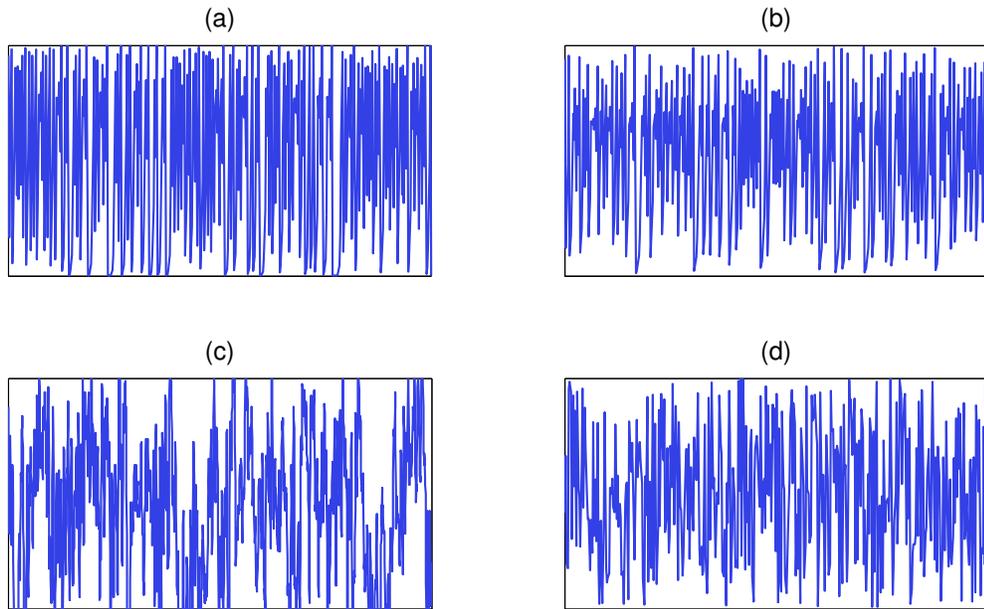


Figure 1.3: What is the correspondence between the plots (a), (b), (c), (d) and the time series (1), (2), (3), (4)?

Can you guess what the correspondence between the plots (a), (b), (c), (d) and the time series (1), (2), (3), (4) is? Indeed, it is complicated to answer the question just looking at Figure 1.3.

Now we plot the values x_{i+1} versus the values x_i (so-called *delay plot*) for each time series, and the situation becomes clear (see Figure 1.4). Indeed, delay plots reveal the deterministic dynamics for the time series observed from the logistic (a) and the tent map (b), complex dynamics for the EEG data (c) and for a sequence of random numbers (d).

Now we come back to the *state space reconstruction*; its basic idea is to consider a sequence of *reconstruction vectors* $(x_i, x_{i+1}, \dots, x_{i+k-1})_{i \in \mathbb{N}_0}$ instead of the original time series $(x_i)_{i \in \mathbb{N}_0}$, in order to reveal the properties of the underlying dynamics. For example, the reconstruction vectors with $k = 2$ reveal the underlying dynamics in Figure 1.4. The considered k is called an *embedding dimension*. By the Takens' theorem if k is high

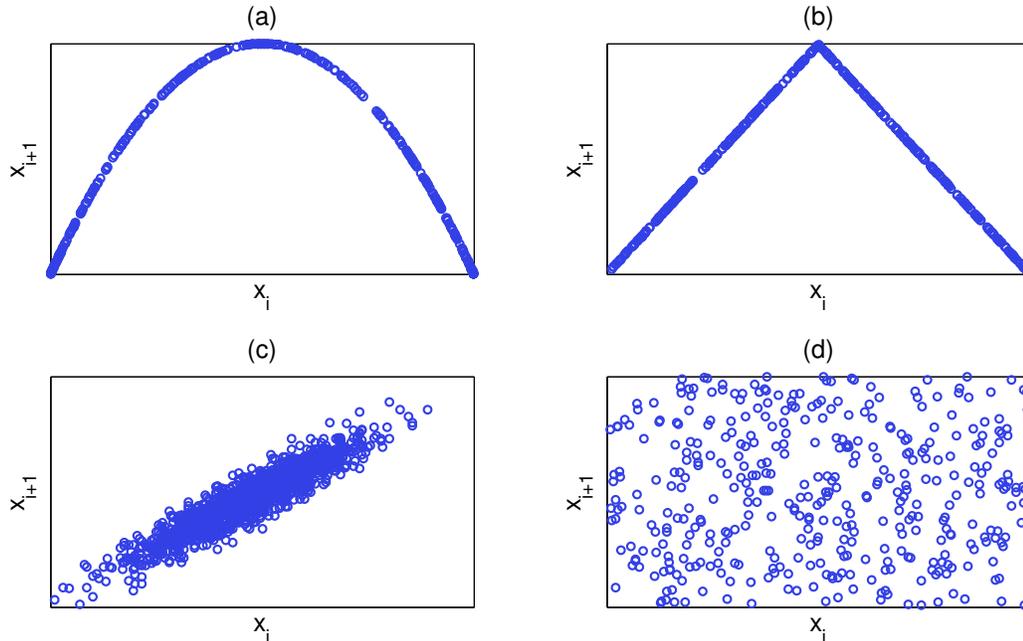


Figure 1.4: Delay plots of time series observed from the logistic map (a), time series observed from the tent map (b), EEG data from [Bon14] (c), and a sequence of random numbers (d)

enough then the properties of the original dynamics can be recovered from the behavior of the reconstruction vectors [Bor98]. We refer for the formulation of Takens' theorem to [Tak81] (see a good and simple explanation of Takens' theorem in [Bor98]). Further we consider reconstruction vectors in Chapter 3 when considering approximate entropy and sample entropy.

1.3 Outline of the thesis

Throughout the thesis we consider the theoretical and practical complexity measures shown in Figure 1.1, we investigate their interrelationship (Chapters 2, 3), compare properties of the practical complexity measures (Chapter 3), propose a fast algorithm for computing the empirical permutation entropy and related measures (Chapter 4), and apply practical complexity measures for analyzing EEG data (Chapter 5). We specify now the main points considered in each chapter.

- In Chapter 2 we investigate when the KS entropy and the permutation entropy coincide. This investigation is motivated by the result of entropies equality for piecewise strictly monotone interval maps established by Bandt, Keller and Pompe [BKP02]. In this chapter we compare the permutation entropy and the KS entropy on the basis of the recent approach from [KUU12]; we have not obtained an affirmative answer whether the KS entropy and the permutation entropy coincide,

but we describe the main problems occurring for the considered approach and present interesting and new results which could be the basis for further research in this direction. In particular, we provide sufficient conditions for coincidence of both entropies for the general and, especially, for the one-dimensional cases.

- In Chapter 3 we compare the practical complexity measures: the widely-used approximate and sample entropy with the relatively new empirical permutation entropy. First, we explain the theoretical underpinnings for the approximate entropy, the sample entropy and the empirical permutation entropy. Then we compare the important for applications to real-world data properties of the entropies, such as computational and storage requirements, sensitivity to the length of a time series, robustness with respect to noise, ability to correctly estimate a large complexity of a time series and robustness with respect to strictly monotone transformations. In particular, we introduce a robust (to noise) empirical permutation entropy, which is further applied to EEG data for epileptic seizure detection in Chapter 5.
- In Chapter 4 we present efficient algorithms for computing ordinal patterns, the empirical permutation entropy, which is based on calculating distributions of ordinal patterns, robust empirical permutation entropy and other ordinal-patterns-based measures (e.g. recently introduced conditional entropy of ordinal patterns [UK14]). In particular, for fast computing of ordinal patterns with tied ranks (adapted to the case of high occurrence of equal values in a time series) the efficient enumeration is introduced. The proposed algorithms are faster than the known methods, they provide a possibility to compute the ordinal-patterns-based measures of large time series in real time.
- Finally, in Chapter 5 we apply the practical complexity measures for analyzing EEG data with the aim of epileptic seizure detection. We discuss the choice of the parameters when applying empirical permutation entropy to EEG data. Then we illustrate how a vigilance state influences the values of empirical permutation entropy, approximate entropy and sample entropy. We demonstrate that in many cases the robust empirical permutation entropy, introduced in Chapter 3, shows better results than the empirical permutation entropy, the approximate entropy and the sample entropy for epileptic seizure detection in EEG recordings during the awake state. We demonstrate also that a combined use of empirical permutation entropy with approximate and sample entropy or a combined use of empirical permutation entropy computed for different values of parameters is a promising approach for measuring complexity and discriminating between different states of a system on the basis of real-world data.

Chapter 2

Relationship of permutation entropy and Kolmogorov-Sinai entropy

In this chapter we discuss the relationship between the permutation entropy [BP02] and the well-known Kolmogorov-Sinai (KS) entropy [Wal00]. This relationship is of interest because, on the one hand, the permutation entropy is a conceptually simple and well-interpretable measure of dynamical systems complexity, and, on the other hand, the empirical permutation entropy is a successfully applied measure of time series complexity (see Chapters 3, 4 for details).

A significant result, given by Bandt, Keller, and Pompe, is equality of KS and permutation entropy for piecewise strictly monotone interval maps [BKP02]. This result gave rise to the question of coincidence of the entropies in the general case. Amigó et al. have shown the equality of the entropies for a slightly different concept of permutation entropy [AKK05, Ami10, Ami12]. The representation of KS entropy on the basis of ordinal partitions given in [KS09, KS10, Kel12] for many cases allows to compare KS entropy and permutation entropy (see also [AKM14] for a recent result). However, apart from the result in [BKP02], to our knowledge, nothing is known about the equality of permutation entropy and KS entropy for other dynamical systems.

Throughout the chapter we use the following result from [KUU12] as a basic approach to comparing the permutation entropy and the KS entropy (see the exact formulation in Theorem 8, p. 16).

Theorem 2. *The following statements are equivalent.*

- (i) *The ordinal partitions $\mathcal{P}^{\mathbf{X}}(d+n-1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$ do not differ too much.*
- (ii) *The permutation entropy and the KS entropy coincide.*

Now we briefly explain the idea behind Theorem 2. Recall that an ordinal pattern of

order d describes order relations between the components of $(d + 1)$ -dimensional vector. Given some observables \mathbf{X} from the dynamical system, the partition $\mathcal{P}^{\mathbf{X}}(d + n - 1)$ is provided by ordinal patterns of order $(d + n - 1)$ and the partition $\mathcal{P}^{\mathbf{X}}(d)_n$ is provided by n successive ordinal patterns of order d .

Example 2. In Figure 2.1 the partitions $\mathcal{P}^{\mathbf{X}}(d)_n$ and $\mathcal{P}^{\mathbf{X}}(d + n - 1)$ are presented for $d = 1$ and $n = 2$. One can see that $\mathcal{P}^{\mathbf{X}}(2)$ refines $\mathcal{P}^{\mathbf{X}}(1)_2$, i.e. the elements of the partition $\mathcal{P}^{\mathbf{X}}(2)$ are proper subsets of the elements of the partition $\mathcal{P}^{\mathbf{X}}(1)_2$. Some elements of the partition $\mathcal{P}^{\mathbf{X}}(1)_2$ contain one element of the partition $\mathcal{P}^{\mathbf{X}}(2)$ and some elements contain two, it depends on the order relations between the iterates. Indeed, some of the elements of the partition $\mathcal{P}^{\mathbf{X}}(1)_2$ determine the relation between ω and $T^2(\omega)$ like $\omega < T(\omega)$ and $T(\omega) < T^2(\omega)$ determine $\omega < T^2(\omega)$, whereas $\omega < T(\omega)$ and $T(\omega) > T^2(\omega)$ do not determine the order relation between ω and $T^2(\omega)$. Many proofs in this chapter are based on playing with the order relations in ordinal patterns of order $d + n - 1$ and in n successive ordinal patterns of order d .

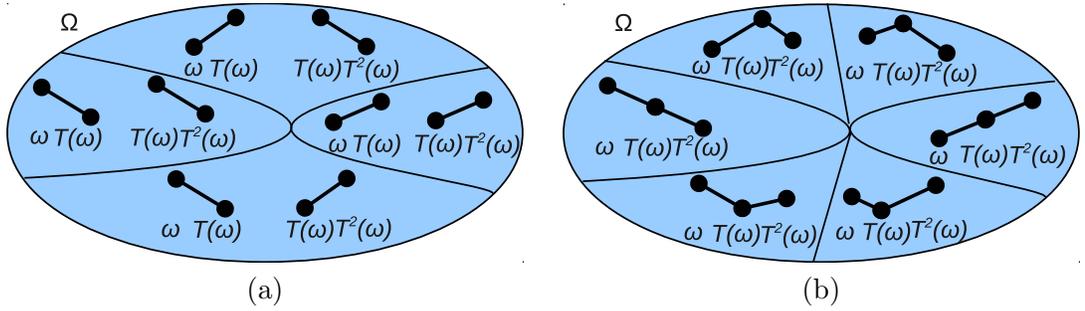


Figure 2.1: The partition $\mathcal{P}^{\mathbf{X}}(1)_2$ by 2 successive ordinal patterns of order 1 (a) and the finer partition $\mathcal{P}^{\mathbf{X}}(2)$ by ordinal patterns of order 2 (b) of the state space Ω

The rest of the chapter is organized as follows. In Section 2.1 the KS entropy, ordinal patterns, ordinal partitions and the permutation entropy are introduced. In Section 2.2 we consider the main approaches to comparing the permutation entropy and the KS entropy. The considerations in Sections 2.3 and 2.4 are based on comparing the partitions $\mathcal{P}^{\mathbf{X}}(d + n - 1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$ (see Theorem 2).

- In Section 2.3 we compare the partitions $\mathcal{P}^{\mathbf{X}}(d + n - 1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$ for the general case on the basis of pure combinatorics and we provide a sufficient condition for the coincidence of the permutation entropy and the KS entropy. However we show that the pure combinatorial relation between the partitions $\mathcal{P}^{\mathbf{X}}(d + n - 1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$ is rather complicated already for $\Omega \subset \mathbb{R}$ with $\mathbf{X} = \mathbf{id}$.
- In Section 2.4 we compare the partitions $\mathcal{P}^{\mathbf{X}}(d + n - 1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$ for mixing interval maps [UUK13]. We formulate for this case a sufficient condition for the equality of the permutation entropy and the KS entropy.

Finally, we discuss possible future directions of work in Section 2.5.

2.1 Preliminaries

2.1.1 Kolmogorov-Sinai entropy

Throughout the chapter, we consider a measure-preserving dynamical system $(\Omega, \mathbb{B}(\Omega), \mu, T)$, see Subsection 1.1 for a background. The (*Shannon*) *entropy* of a finite partition $\mathcal{P} = \{P_1, P_2, \dots, P_l\} \subset \mathbb{B}(\Omega)$ of Ω with respect to μ is defined by

$$H(\mathcal{P}) = - \sum_{P \in \mathcal{P}} \mu(P) \ln \mu(P)$$

(with $0 \ln 0 := 0$).

Let us associate the elements of a finite partition $\mathcal{P} = \{P_1, P_2, \dots, P_l\}$ with letters of the alphabet $A = \{1, 2, \dots, l\}$. One can make a word $a_1 a_2 \dots a_n$ of given length n from the letters $a \in A$. The set A^n of all such words $a_1 a_2 \dots a_n$ provides a partition \mathcal{P}_n of Ω into the sets

$$P_{a_1 a_2 \dots a_n} = \{\omega \mid \omega \in P_{a_1}, T(\omega) \in P_{a_2}, \dots, T^{n-1}(\omega) \in P_{a_n}\}. \quad (2.1)$$

The *entropy rate* of the map T with respect to the measure μ and the partition \mathcal{P} is given by

$$h_\mu(T, \mathcal{P}) = \lim_{n \rightarrow \infty} \frac{H(\mathcal{P}_n)}{n}. \quad (2.2)$$

It is well known that the limit in (2.2) exists (see, for example, [Wal00]). One can see that different partitions of the state space Ω provide different entropy rates. Therefore, in order to have a uniquely defined measure of complexity, one takes the supremum over all possible finite partitions in the following definition of the KS entropy. The *Kolmogorov-Sinai (KS) entropy* is defined by

$$h_\mu(T) = \sup_{\mathcal{P} \text{ finite partition}} \lim_{n \rightarrow \infty} \frac{H(\mathcal{P}_n)}{n}. \quad (2.3)$$

Roughly speaking, the KS entropy $h_\mu(T)$ measures the unpredictability of the dynamics T .

Remark 1. Note that in some books (e.g. [Dow11]) one takes a supremum over all infinite partitions in the definition of KS entropy, whereas in (2.3) it is taken over all finite partitions. These definitions are equivalent since by Lemma 1.19 from [ELW11] the KS entropy can be computed using finite partitions only.

In the general case it is not easy to determine the KS entropy by (2.3), because one has to consider an infinite number of finite partitions \mathcal{P} of the state space Ω . The exception is the case of existence of a *generating partition*.

Definition 3. A finite partition $\mathcal{G} = \{G_1, G_2, \dots, G_l\} \subset \mathbb{B}(\Omega)$ of Ω is said to be *generating* (under T) if, given the sigma-algebra \mathcal{A} generated by the sets $T^{-n}(G_i)$ with $i = 1, 2, \dots, l$ and $n \in \mathbb{N}$, for every $B \in \mathbb{B}(\Omega)$ there exists a set $A \in \mathcal{A}$ such that $\mu(A \triangle B) = 0$.

Due to the Kolmogorov-Sinai theorem, the KS entropy and the entropy rate of a *generating partition* \mathcal{G} coincide: $h_\mu(T) = h_\mu(T, \mathcal{G})$ [Wal00].

2.1.2 Lyapunov exponents

In certain cases the KS entropy can be computed or at least bounded by Lyapunov exponents due to Pesin's formula [Pes97, You03, You13]. Roughly speaking, Lyapunov exponents measure how two infinitely close to each other points from the state space diverge under the action of the dynamics T . Throughout the thesis we use only a particular case of Pesin's formula for differentiable interval maps on the space $([0, 1], \mathbb{B}([0, 1]), \mu)$.

Definition 4 ([Cho00]). Given a piecewise continuously differentiable map $T : [0, 1] \leftarrow$ the *Lyapunov exponent* is defined by

$$\lambda(T) = \int_0^1 \ln |T'(\omega)| d\mu(\omega).$$

For the following discussion we recall the definition of a Sinai-Ruelle-Bowen (SRB) measure for the case $\Omega \subseteq \mathbb{R}^N$ [MN00].

Definition 5. A measure μ is said to be *Sinai-Ruelle-Bowen (SRB)* on $(\Omega, \mathbb{B}(\Omega))$ with respect to T if for Lebesgue-almost all $\omega \in \Omega$ it holds

$$\mu(B) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} 1_B(T^i(\omega)), \quad (2.4)$$

where 1_B is the characteristic function of the set B .

The SRB measure is the most natural measure for the system.

Theorem 3. (Pesin's formula for the one-dimensional case.) *If μ is the ergodic T -invariant SRB measure on $([0, 1], \mathbb{B}([0, 1]))$ then for $\lambda(T) > 0$ it holds*

$$h_\mu(T) = \lambda(T) = \int_0^1 \ln |T'(\omega)| d\mu(\omega). \quad (2.5)$$

Moreover, in this case the KS entropy can be estimated from the orbit of μ -almost every point ω :

$$h_\mu(T) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |T'(T^i(\omega))|. \quad (2.6)$$

Proof. Equality (2.5) is a direct consequence of Pesin's formula [You13, Theorem 1], and equation (2.6) follows from (2.5) by Birkhoff's Ergodic Theorem [Cho00]. \square

Example 3. Throughout the thesis we often use in examples the logistic map $T_{\text{LM}} : [0, 1] \leftarrow$ given by $T_{\text{LM}}(\omega) = A\omega(1 - \omega)$ for $A \in [3.5, 4]$. Note that for the family of logistic maps it has been shown that for almost all $A \in [3.5, 4]$ there exists the SRB measure μ_A [MN00, Lyu02]. By Pesin's formula, for almost all $A \in [3.5, 4]$,

the KS entropy of the dynamical system $([0, 1], \mathbb{B}([0, 1]), \mu_A, T_{LM})$, is given by $h_{\mu_A}(T_{LM}) = \max\{\lambda(T_{LM}), 0\}$. The Lyapunov exponent for the logistic maps can be estimated rather accurately [Spr03], therefore we use further this family for comparing practical complexity measures with the KS entropy.

2.1.3 Ordinal patterns and ordinal partitions

We define here ordinal patterns and ordinal partitions which we need for defining the permutation entropy in the subsequent subsection.

Definition 6. Let Π_d be the set of permutations of the set $\{0, 1, \dots, d\}$ for $d \in \mathbb{N}$. Then the real vector $(x_0, x_1, \dots, x_d) \in \mathbb{R}^{d+1}$ has the *ordinal pattern* $\pi = (r_0, r_1, \dots, r_d) \in \Pi_d$ of order d if

$$\begin{aligned} x_{r_0} &\geq x_{r_1} \geq \dots \geq x_{r_d}, \text{ where} \\ r_{l-1} &> r_l \text{ in the case } x_{r_{l-1}} = x_{r_l}. \end{aligned}$$

Definition 7. We say that a real vector $(x_0, x_1, \dots, x_{d+n-1}) \in \mathbb{R}^{d+n}$ has the (n, d) -word $\pi_1 \pi_2 \dots \pi_n$ if

$$(x_i, x_{i+1}, \dots, x_{i+d}) \text{ has the ordinal pattern } \pi_{i+1} \in \Pi_d \text{ for } i = 0, 1, \dots, n-1.$$

We divide now the state space $\Omega \subset \mathbb{R}$ into the sets of points having the same dynamics from the ordinal viewpoint.

Definition 8. For $d \in \mathbb{N}$, the partition $\mathcal{P}(d) = \{P_\pi \mid \pi \in \Pi_d\}$ with

$$P_\pi = \left\{ \omega \in \Omega \mid \left(T^d(\omega), T^{d-1}(\omega), \dots, T(\omega), \omega \right) \text{ has the ordinal pattern } \pi \right\}$$

is called the *ordinal partition of order d* with respect to T .

Remark 2. Note that in contrast to the traditional definition of a partition, $\mathcal{P}(d)$ may contain some empty sets $P_\pi \in \mathcal{P}(d)$, corresponding to the unrealized ordinal patterns $\pi \in \Pi_d$. However, this distinction does not cause any problems and $\mathcal{P}(d)$ still has all properties of the partition.

The partition $\mathcal{P}(d)_n$ associated with the collection of (n, d) -words consists of the sets (note, analogously, Remark 2)

$$P_{\pi_1 \pi_2 \dots \pi_n} = \left\{ \omega \mid \omega \in P_{\pi_1}, T(\omega) \in P_{\pi_2}, \dots, T^{n-1}(\omega) \in P_{\pi_n} \right\}, \pi_1, \pi_2, \dots, \pi_n \in \Pi_d.$$

In order to define the ordinal partition in a general way we consider observables $\mathbf{X} = (X_1, X_2, \dots, X_N)$ for $N \in \mathbb{N}$ from the dynamical system such that $X_i : \Omega \rightarrow \mathbb{R}$ for $i = 1, 2, \dots, N$.

Definition 9. For $N \in \mathbb{N}$, let $\mathbf{X} = (X_1, X_2, \dots, X_N)$ be an \mathbb{R} -valued random vector on $(\Omega, \mathbb{B}(\Omega))$. Then, the partition $\mathcal{P}^{\mathbf{X}}(d) = \left\{ P_{(\pi^1, \pi^2, \dots, \pi^N)} \mid \pi^j \in \Pi_d \text{ for } j = 1, 2, \dots, N \right\}$ for $d \in \mathbb{N}$ with (note, analogously, Remark 2)

$$P_{(\pi^1, \pi^2, \dots, \pi^N)} = \{ \omega \in \Omega \mid (X_j(T^d(\omega)), X_j(T^{d-1}(\omega)), \dots, X_j(\omega)) \text{ has the ordinal pattern } \pi^j \text{ for } j = 1, 2, \dots, N \}$$

is called the *ordinal partition of order d with respect to T and \mathbf{X}* .

Consider an n -tuple $a = (\pi^1, \pi^2, \dots, \pi^N)$ as a letter from the alphabet $\Pi_d^N = (\Pi_d)^N$. One obtains the partition $\mathcal{P}^{\mathbf{X}}(d)_n$ generated by the words $a_1 a_2 \dots a_n$ with the letters $a_i \in \Pi_d^N$ for $i = 1, 2, \dots, n$ (compare with (2.1), note, analogously, Remark 2):

$$\begin{aligned} \mathcal{P}^{\mathbf{X}}(d)_n &= \{ P_{a_1 a_2 \dots a_n} \mid a_i = (\pi_i^1, \pi_i^2, \dots, \pi_i^N) \in \Pi_d^N \text{ for } i = 1, 2, \dots, n \}, \text{ where} \\ P_{a_1 a_2 \dots a_n} &= \{ \omega \in \Omega \mid \omega \in P_{a_1}, T(\omega) \in P_{a_2}, \dots, T^{n-1}(\omega) \in P_{a_n} \}. \end{aligned}$$

2.1.4 Permutation entropy and Kolmogorov-Sinai entropy

Definition 10. Given a random vector $\mathbf{X} = (X_1, X_2, \dots, X_N)$ on $(\Omega, \mathbb{B}(\Omega))$ with $X_i : \Omega \rightarrow \mathbb{R}$ for $i = 1, 2, \dots, N$, the *permutation entropy* with respect to \mathbf{X} is defined by

$$h_{\mu}^{\mathbf{X}}(T) = \overline{\lim}_{d \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(d))}{d}. \quad (2.7)$$

For natural choices of observables \mathbf{X} the KS entropy can be represented on the basis of ordinal partitions [Kel12]. This provides a possibility to compare the permutation entropy and the KS entropy. In particular, for these choices of observables it was shown that the KS entropy is not greater than the permutation entropy [Kel12].

Theorem 4. For $N \in \mathbb{N}$, let $\mathbf{X} = (X_1, X_2, \dots, X_N)$ be a random vector on $(\Omega, \mathbb{B}(\Omega))$. Then for \mathbf{X} satisfying the conditions from Theorems 5-7 in [KUU12] it holds

$$h_{\mu}(T) = \lim_{d \rightarrow \infty} h_{\mu}(T, \mathcal{P}^{\mathbf{X}}(d)) = \lim_{d \rightarrow \infty} \lim_{n \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{n}, \text{ and} \quad (2.8)$$

$$h_{\mu}(T) \leq h_{\mu}^{\mathbf{X}}(T). \quad (2.9)$$

The following choice of observables \mathbf{X} provides (2.8) as it was shown in [Kel12].

Theorem 5. Let $N \in \mathbb{N}$, Ω be a Borel subset of \mathbb{R}^N and $\mathbf{X} = (X_1, X_2, \dots, X_N)$ be a random vector on $(\Omega, \mathbb{B}(\Omega))$ such that X_i is the i -th coordinate projection for $i = 1, 2, \dots, N$, i.e.

$$X_i((\omega_1, \omega_2, \dots, \omega_N)) = \omega_i \text{ for } (\omega_1, \omega_2, \dots, \omega_N) \in \Omega. \quad (2.10)$$

Then (2.8) and (2.9) are valid.

(See [Kel12, KUU12] for other possible choices of observables \mathbf{X} .) In Sections 2.2 and 2.3 we use observables \mathbf{X} given by (2.10), in Section 2.4 we consider \mathbf{X} as an identity map $\mathbf{X} = \mathbf{id}$. According to Remark 2 the following lemma holds.

Lemma 6. *For $d, n, N \in \mathbb{N}$, $\mathbf{X} = (X_1, X_2, \dots, X_N)$ given by (2.10), it holds*

$$\begin{aligned} |\mathcal{P}^{\mathbf{X}}(d)_n| &= ((d+1)!(d+1)^{n-1})^N, \\ |\mathcal{P}^{\mathbf{X}}(d+n-1)| &= ((d+n)!)^N. \end{aligned}$$

2.2 Equality of permutation entropy and Kolmogorov-Sinai entropy

2.2.1 Piecewise strictly monotone interval maps

In this subsection we discuss the equality of permutation entropy and KS entropy for piecewise strictly monotone interval map T and for an identity map $\mathbf{X} = \mathbf{id}$, shown in [BKP02].

Definition 11. Given an interval $[a, b] \subset \mathbb{R}$, a map $T : [a, b] \leftrightarrow$ is said to be *piecewise strictly monotone* if there is a finite partition of $[a, b]$ into intervals such that T is continuous and strictly monotone on each of those intervals.

Theorem 7. [BKP02] *Given an interval $[a, b] \subset \mathbb{R}$ and a piecewise strictly monotone map $T : [a, b] \leftrightarrow$, for an identity map $\mathbf{X} = \mathbf{id}$ it holds*

$$h_\mu(T) = h_\mu^{\mathbf{X}}(T).$$

Remark 3. It was also shown in [BKP02] that the *topological permutation entropy* and the *topological entropy* (see definitions in [BKP02, Mis03] and [Mis03, You03], correspondingly) coincide for piecewise strictly monotone interval maps. However, Misiurewicz has shown that topological permutation entropy does not coincide with the topological entropy for arbitrary continuous interval maps [Mis03]. To our knowledge, one still has not found an example of a dynamical system $(\Omega, \mathbb{B}(\Omega), T, \mu)$ with $h_\mu(T) \neq h_\mu^{\mathbf{X}}(T)$.

Example 4. In order to illustrate the relationship of permutation entropy and KS entropy for piecewise strictly monotone interval maps, we introduce the *permutation entropy of order d* with respect to observables \mathbf{X} in the following way:

$$h_\mu^{\mathbf{X}}(T, d) = \frac{H(\mathcal{P}^{\mathbf{X}}(d))}{d}.$$

Despite of equality of the entropies for piecewise strictly monotone interval maps, the permutation entropy of order d converges to the KS entropy with increasing d rather slowly [BKP02]. In Figure 2.2 we present the values of the permutation entropy of order

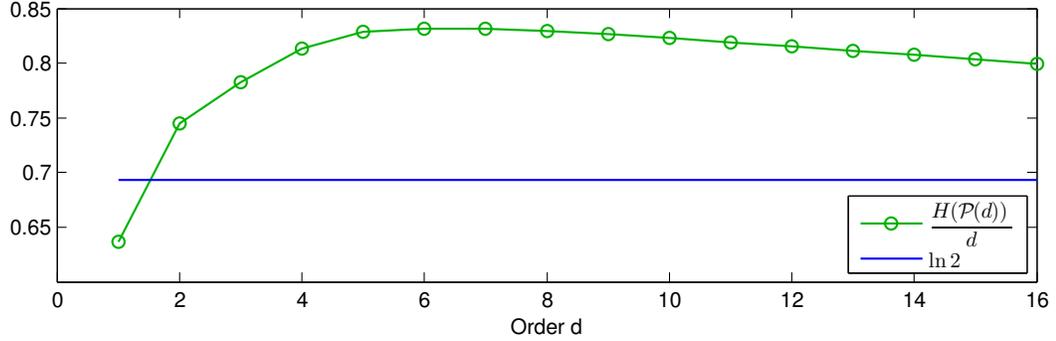


Figure 2.2: The values of the permutation entropy of order d computed from an orbit of the logistic map T_{LM}

d computed from an orbit of the logistic map T_{LM} for $A = 4$, the length of the orbit is 10^8 . The KS entropy of the logistic map for $A = 4$ is equal to $\ln 2$ [Cho00]. Note that there is a modification of permutation entropy of order d , which is called *conditional entropy of ordinal patterns of order d* and provides better convergence to the KS entropy for several cases (see [UK14, Una15] for details).

2.2.2 A criterion for the coincidence in the general case

Note that it is difficult to generalize Theorem 7 directly because its proof is essentially based on the piecewise monotonicity. Therefore we present a criterion for the coincidence of permutation entropy and KS entropy from [KUU12], which is the basic approach to comparing the entropies in this chapter.

Theorem 8. *The following statements are equivalent for $h_\mu(T)$ satisfying (2.8):*

(i) $h_\mu(T) = h_\mu^{\mathbf{X}}(T)$.

(ii) For each $\varepsilon > 0$ there exists some $d_\varepsilon \in \mathbb{N}$ such that for all $d \geq d_\varepsilon$ there is some $n_d \in \mathbb{N}$ with

$$H(\mathcal{P}^{\mathbf{X}}(d+n-1)) - H(\mathcal{P}^{\mathbf{X}}(d)_n) < (n-1)\varepsilon \text{ for all } n \geq n_d. \quad (2.11)$$

(The proof is given in Subsection 2.6.1.) Roughly speaking, Theorem 8 says that the permutation entropy and the KS entropy coincide when the partitions $\mathcal{P}^{\mathbf{X}}(d+n-1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$ do not differ “too much”. Note that the partition $\mathcal{P}^{\mathbf{X}}(d+n-1)$ is finer than the partition $\mathcal{P}^{\mathbf{X}}(d)_n$, therefore it holds

$$H(\mathcal{P}^{\mathbf{X}}(d+n-1)) \geq H(\mathcal{P}^{\mathbf{X}}(d)_n). \quad (2.12)$$

Remark 4. Note that we do not consider here the approach to comparing the permutation entropy and the KS entropy proposed in [Ami10, Ami12]. The author shows there that the modified permutation entropy, qualitatively different from Definition 10, coincides with the KS entropy. See also [KUU12, Subsection 3.4] and [HN11] for more details.

2.3 Comparing ordinal partitions $\mathcal{P}^{\mathbf{X}}(d+n-1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$

In Subsections 2.3.1 and 2.3.2 we describe the relation between the partitions $\mathcal{P}^{\mathbf{X}}(d+n-1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$ for the general case (Corollary 9, Proposition 10) and for $\Omega \subset \mathbb{R}$ with $\mathbf{X} = \text{id}$ (Proposition 11), correspondingly.

2.3.1 The general case

We recall that the partition $\mathcal{P}^{\mathbf{X}}(d+n-1)$ is a refinement of the partition $\mathcal{P}^{\mathbf{X}}(d)_n$, i.e. each element of the partition $\mathcal{P}^{\mathbf{X}}(d+n-1)$ is a subset of some element of the partition $\mathcal{P}^{\mathbf{X}}(d)_n$. In order to compare the partitions $\mathcal{P}^{\mathbf{X}}(d+n-1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$, we introduce a set $Q^{\mathbf{X}}(k, n, d)$ for $k, n, d \in \mathbb{N}$ as the union of those elements $P \in \mathcal{P}^{\mathbf{X}}(d)_n$ that are unions of exactly k elements from the partition $\mathcal{P}^{\mathbf{X}}(d+n-1)$:

$$Q^{\mathbf{X}}(k, n, d) = \bigcup_{P \in \mathcal{P}^{\mathbf{X}}(k, n, d)} P, \text{ where} \\ \mathcal{P}^{\mathbf{X}}(k, n, d) = \left\{ P \in \mathcal{P}^{\mathbf{X}}(d)_n \mid P = \bigcup_{i=1}^k P_i, \text{ with } P_i \in \mathcal{P}^{\mathbf{X}}(d+n-1) \right\}. \quad (2.13)$$

Using the introduced sets and the entropy properties it holds¹

$$H(\mathcal{P}^{\mathbf{X}}(d)_n) = - \sum_{k=1}^{(n!)^N} \sum_{P \in \mathcal{P}^{\mathbf{X}}(k, n, d)} \mu(P) \ln \mu(P), \text{ whereas} \quad (2.14) \\ H(\mathcal{P}^{\mathbf{X}}(d+n-1)) = - \sum_{P' \in \mathcal{P}^{\mathbf{X}}(d+n-1)} \mu(P') \ln \mu(P') \\ = - \sum_{k=1}^{(n!)^N} \sum_{P \in \mathcal{P}^{\mathbf{X}}(k, n, d)} \sum_{P' \subset P \mid P' \in \mathcal{P}^{\mathbf{X}}(d+n-1)} \mu(P') \ln \mu(P') \\ \leq - \sum_{k=1}^{(n!)^N} \sum_{P \in \mathcal{P}^{\mathbf{X}}(k, n, d)} k \frac{\mu(P)}{k} \ln \frac{\mu(P)}{k}. \quad (2.15)$$

¹The summation in (2.14) is going up to $(n!)^N$. Indeed, we show further in Proposition 12 that for given $n, d \in \mathbb{N}$ and for $\Omega \subset \mathbb{R}$ it holds

$$\max_{P' \subset P \mid P \in \mathcal{P}(d)_n} \#\{P' \in \mathcal{P}(d+n-1)\} \leq n!$$

Then by (2.14) and (2.15) one can represent the difference of the entropies in the following way

$$\begin{aligned}
H(\mathcal{P}^{\mathbf{X}}(d+n-1)) - H(\mathcal{P}^{\mathbf{X}}(d)_n) &\leq \sum_{k=1}^{(n!)^N} \sum_{P \in \mathcal{P}^{\mathbf{X}}(k,n,d)} \left(\mu(P) \ln \mu(P) - k \frac{\mu(P)}{k} \ln \frac{\mu(P)}{k} \right) \\
&= \sum_{k=1}^{(n!)^N} \ln k \sum_{P \in \mathcal{P}^{\mathbf{X}}(k,n,d)} \mu(P) \\
&= \sum_{k=1}^{(n!)^N} \mu(Q^{\mathbf{X}}(k,n,d)) \ln k. \tag{2.16}
\end{aligned}$$

By (2.16) we obtain the following corollary from Theorem 8, which provides a sufficient condition for the coincidence of permutation entropy and KS entropy.

Corollary 9. *For $h_\mu(T)$ satisfying (2.8), for the following statements (i) implies (ii).*

(i) *For each $\varepsilon > 0$ there exists some $d_\varepsilon \in \mathbb{N}$ such that for all $d \geq d_\varepsilon$ there is some $n_d \in \mathbb{N}$ such that for all $n \geq n_d$ it holds*

$$\sum_{k=1}^{(n!)^N} \mu(Q^{\mathbf{X}}(k,n,d)) \ln k < (n-1)\varepsilon.$$

(ii) $h_\mu(T) = h_\mu^{\mathbf{X}}(T)$.

In order to illustrate the difficulties of the combinatorial approach, we consider in Example 5 the “worst case” for Corollary 9, namely, the equidistributed measure between the elements of the partition $\mathcal{P}^{\mathbf{X}}(d+n-1)$.

Example 5. We assume that for all $P \in \mathcal{P}^{\mathbf{X}}(d+n-1)$ it holds $\mu(P) = \frac{1}{((d+n)!)^N}$. (We are not sure whether such dynamical system exists.) Then (see (2.13) for the definition of $\mathcal{P}^{\mathbf{X}}(k,n,d)$)

$$\mu(Q^{\mathbf{X}}(k,n,d)) = \frac{k}{((d+n)!)^N} |P^{\mathbf{X}}(k,n,d)|.$$

In this case (2.16) holds with equality:

$$\begin{aligned}
H(\mathcal{P}^{\mathbf{X}}(d+n-1)) - H(\mathcal{P}^{\mathbf{X}}(d)_n) &= \sum_{k=1}^{(n!)^N} \mu(Q^{\mathbf{X}}(k,n,d)) \ln k \\
&= \frac{1}{((d+n)!)^N} \sum_{k=1}^{(n!)^N} |P^{\mathbf{X}}(k,n,d)| k \ln k. \tag{2.17}
\end{aligned}$$

Determining $|P^{\mathbf{X}}(k, n, d)|$ or bounding it by $q_1(k, n, d) < |P^{\mathbf{X}}(k, n, d)| < q_2(k, n, d)$ for some $q_1(k, n, d)$ and $q_2(k, n, d)$ for $k = 1, 2, \dots, (n!)^N$ can help understanding when statement (ii) from Theorem 8 holds. However, determining $|P^{\mathbf{X}}(k, n, d)|$ leads to a complicated combinatorics even for $\Omega \subset \mathbb{R}$ with $\mathbf{X} = \mathbf{id}$ as we show in Subsection 2.3.2.

Let us compare now the partitions $\mathcal{P}^{\mathbf{X}}(d + n - 1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$ by comparing the “neighboring” partitions $\mathcal{P}^{\mathbf{X}}(d + 1)_{n-1}$ and $\mathcal{P}^{\mathbf{X}}(d)_n$. For this we define a set $V_{\Pi_d} \subset \Pi_d$:

$$V_{\Pi_d} = \{\pi = (r_0, r_1, \dots, r_d) \in \Pi_d : r_a = 0, r_b = d \text{ for some } a, b : |a - b| = 1\}, \quad (2.18)$$

i.e. V_{Π_d} is the set of all ordinal patterns of order d with the numbers 0 and d staying nearby. It holds

$$|V_{\Pi_d}| = 2d!, \quad (2.19)$$

whereas $|\Pi_d| = (d + 1)!$.

Example 6. We present all ordinal patterns of order $d = 3$, one can see that the numbers 0 and 3 in ordinal patterns from V_{Π_3} (in gray) are staying nearby.

(0, 1, 2, 3)	(0, 3, 1, 2)	(1, 2, 0, 3)	(2, 0, 1, 3)	(2, 3, 0, 1)	(3, 1, 0, 2)
(0, 1, 3, 2)	(0, 3, 2, 1)	(1, 2, 3, 0)	(2, 0, 3, 1)	(2, 3, 1, 0)	(3, 1, 2, 0)
(0, 2, 1, 3)	(1, 0, 2, 3)	(1, 3, 0, 2)	(2, 1, 0, 3)	(3, 0, 1, 2)	(3, 2, 0, 1)
(0, 2, 3, 1)	(1, 0, 3, 2)	(1, 3, 2, 0)	(2, 1, 3, 0)	(3, 0, 2, 1)	(3, 2, 1, 0)

The following proposition allows to compare the partitions $\mathcal{P}^{\mathbf{X}}(d + 1)_{n-1}$ and $\mathcal{P}^{\mathbf{X}}(d)_n$ by means of the set $V_{\Pi_{d+1}}$.

Proposition 10. *Let $d, n, N \in \mathbb{N}$, $\mathbf{X} = (X_1, X_2, \dots, X_N) \in \mathbb{R}^N$. Then for all $P_{a_1 a_2 \dots a_n} \in \mathcal{P}^{\mathbf{X}}(d)_n$ with $a_i = (\pi_i^1, \pi_i^2, \dots, \pi_i^N)$ it holds*

$$\#\{P' \in \mathcal{P}^{\mathbf{X}}(d + 1)_{n-1} \mid P' \subset P_{a_1 a_2 \dots a_n}\} = 2^{m_1} 2^{m_2} \dots 2^{m_N}, \text{ where}$$

$$m_k = \#\{i = 1, 2, \dots, n - 1 \mid \pi_i^k \in V_{\Pi_{d+1}}\} \text{ for } k = 1, 2, \dots, N.$$

(Proposition 10 generalizes Proposition 16, the proofs are analogous, see the proof of Proposition 16 in Section 2.4.) However, it is difficult to obtain a similar result for the “distant” partitions $\mathcal{P}^{\mathbf{X}}(d + n - 1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$ due to the complicated combinatorics already for $\Omega \subset \mathbb{R}$ with $\mathbf{X} = \mathbf{id}$ as we show in Subsection 2.3.2.

2.3.2 The one-dimensional case

In this subsection we show the difficulties of a pure combinatorial approach to comparing the partitions $\mathcal{P}(d + n - 1)$ and $\mathcal{P}(d)_n$ for $\Omega \subset \mathbb{R}$ and $\mathbf{X} = \mathbf{id}$. We start from comparing

the “neighboring” partitions $\mathcal{P}(d)_n$ and $\mathcal{P}(d+1)_{n-1}$ in the following proposition ensuing from Proposition 10.

Proposition 11. *Given $d, n \in \mathbb{N}$ for all $m = 0, 1, \dots, n-1$ it holds*

$$\#\left\{P \in \mathcal{P}(d)_n \mid \bigcup_{i=1}^{2^m} P_i = P \text{ with } P_i \in \mathcal{P}(d+1)_{n-1}\right\} = 2(d+1)!d^{n-m-1} \binom{n-2}{m}. \quad (2.20)$$

(The proof is given in Subsection 2.6.2). One can see that (2.20) looks complicated already for the relation between the “neighboring” partitions $\mathcal{P}(d+1)_{n-1}$ and $\mathcal{P}(d)_n$ for $\Omega \subset \mathbb{R}$. It is also not clear how to move from the combinatorial relation between the “neighboring” partitions $\mathcal{P}(d+1)_{n-1}$ and $\mathcal{P}(d)_n$ to the relation between the partitions $\mathcal{P}(d+n-1)$ and $\mathcal{P}(d)_n$ since the relation between the partitions $\mathcal{P}(d+n-1)$ and $\mathcal{P}(d)_n$ is more complicated that we illustrate in the following example.

Example 7. Given an element $P \in \mathcal{P}(d)_n$ we determine how many elements from the partition $\mathcal{P}(d+n-1)$ it contains. For example, we consider $P_{\pi_1\pi_2\pi_3\pi_4} \in \mathcal{P}(2)_4$ with $\pi_1 = (1, 0, 2)$, $\pi_2 = (2, 1, 0)$, $\pi_3 = (0, 2, 1)$ and $\pi_4 = (1, 0, 2)$ and we want to determine

$$\#\{P' \in \mathcal{P}(5) \mid P' \subset P_{\pi_1\pi_2\pi_3\pi_4}\}. \quad (2.21)$$

For that we represent $(\pi_i)_{i=1}^4$ by the following rule, which does not change ordering:

$$\pi'_i = \pi_i + (n-i)(1, 1, \dots, 1), \quad (2.22)$$

i.e. for $n = 4$ it holds $\pi'_4 = (1, 0, 2)$, $\pi'_3 = (1, 3, 2)$, $\pi'_2 = (4, 3, 2)$ and $\pi'_1 = (4, 3, 5)$. In Figure 2.3 we present now $P_{\pi_1\pi_2\pi_3\pi_4}$ by drawing the tree from left to right moving from π'_4 to π'_1 , on each level of the tree the next point i is included according to π'_{n-i} .

One can see that $\pi'_4 = (1, 0, 2)$ is split into $(\mathbf{1}, \mathbf{3}, 0, \mathbf{2})$ and $(\mathbf{1}, 0, \mathbf{3}, \mathbf{2})$, because from $\pi'_4 = (\mathbf{1}, \mathbf{0}, \mathbf{2})$ and $\pi'_3 = (\mathbf{1}, \mathbf{3}, \mathbf{2})$ do not determine the order relation between the points 0 and 3. On the second level there are two possibilities to include the point 4 in the upper branch $(1, 3, 0, 2)$ (on the left or on the right from 1) and three possibilities to include the point 4 in the bottom branch $(1, 0, 3, 2)$ (left from 1, 0, in the middle between 1, 0, and on the right from 1, 0).

Finally, we obtain nine elements $P' \in \mathcal{P}(5)$ satisfying (2.21):

$$\begin{aligned} P_{\pi_1\pi_2\pi_3\pi_4} = & P_{(4,1,3,5,0,2)} \cup P_{(4,1,3,0,5,2)} \cup P_{(4,1,3,0,2,5)} \cup P_{(1,4,3,5,0,2)} \cup P_{(1,4,3,0,5,2)} \\ & \cup P_{(1,4,3,0,2,5)} \cup P_{(4,1,0,3,5,2)} \cup P_{(4,1,0,3,2,5)} \cup P_{(1,4,0,3,5,2)} \cup P_{(1,4,0,3,2,5)} \\ & \cup P_{(1,0,4,3,5,2)} \cup P_{(1,0,4,3,2,5)}. \end{aligned}$$

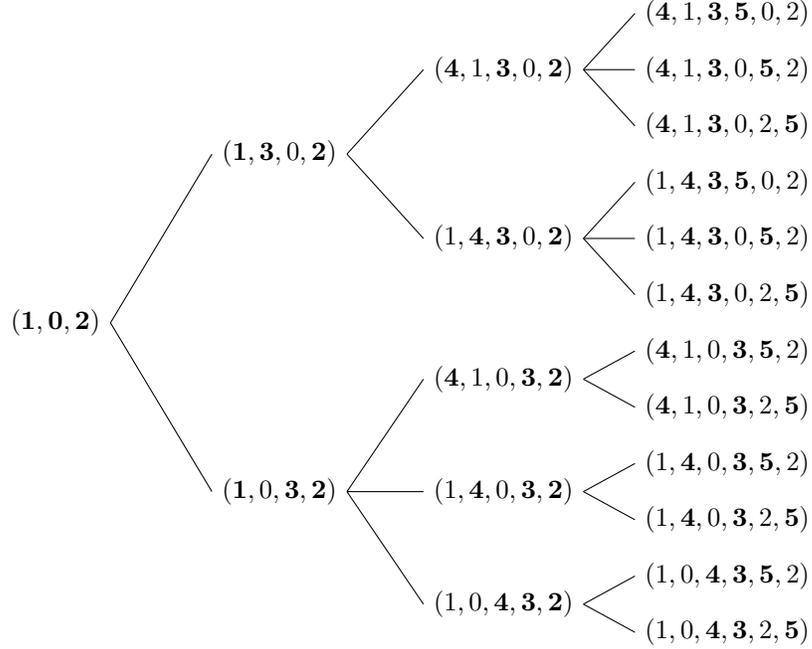


Figure 2.3: The elements from the partition $\mathcal{P}(5)$ are obtained from one element $P_{\pi_1\pi_2\pi_3\pi_4} \in \mathcal{P}(2)_4$ according to the order relations $\pi_1 = (1, 0, 2)$, $\pi_2 = (2, 1, 0)$, $\pi_3 = (0, 2, 1)$ and $\pi_4 = (1, 0, 2)$

In the same way, one can determine $\#\{P' \in \mathcal{P}(d+n-1) \mid P' \subset P\}$ for every given $P \in \mathcal{P}(d)_n$, but it seems complicated to obtain the general formula. However we bound $\#\{P' \in \mathcal{P}(d+n-1) \mid P' \subset P\}$ from above in Proposition 12.

Proposition 12. *For all $n, d \in \mathbb{N}$ it holds*

$$\max_{P \in \mathcal{P}(d)_n} \#\{P' \in \mathcal{P}(d+n-1) \mid P' \subset P\} \leq n!. \quad (2.23)$$

Proof. The statement becomes clear if we look at Figure 2.3. We consider $P_{\pi_1\pi_2\dots\pi_n} \in \mathcal{P}(d)_n$ and we represent $\pi'_i \in \Pi_d$ by (2.22) for all $i = 1, 2, \dots, n$. Then we enumerate the levels of the tree from left to right from 1 to n . When moving from the level i to the level $i+1$ the relation between the points $i, i+1, \dots, d+i$ is known due to π'_{i+1} , but the order relation between $d+i$ and $0, 1, \dots, i-1$ is not known. Hence each “node” of the level i has maximum $i+1$ “children”, because there are $i+1$ possibilities to place each number. From that (2.23) follows. \square

Note that the bound $n!$ is not sharp (see, for instance, Example 8).

Example 8. In this example we illustrate that it does not seem simple to obtain the general formula for $|P^{\mathbf{X}}(k, n, d)|$. For $\mathbf{X} = \mathbf{id}$ and $\Omega \subset \mathbb{R}$ we present in Table 2.1 the values of $|P^{\mathbf{X}}(k, n, d)|$ for $k = 1, 2, \dots, n!$ for $n = 4$ and $d = 1, 2, \dots, 5$.

k	1	2	3	4	5	6	7	8
$d = 1$	2	0	0	4	0	2	0	0
$d = 2$	34	14	24	30	6	18	12	0
$d = 3$	428	256	304	208	60	140	56	0
$d = 4$	61872	39936	26496	13392	3312	7056	1440	144
$d = 5$	773040	474000	254400	122640	26640	55920	9120	1440

k	9	10	11	12	13	14	15	16	17	...	24
$d = 1$	4	0	2	0	0	0	0	2	0	0	0
$d = 2$	12	6	0	4	0	2	0	0	0	0	0
$d = 3$	48	24	0	8	0	4	0	0	0	0	0
$d = 4$	1152	576	0	96	0	48	0	0	0	0	0
$d = 5$	7200	3600	0	480	0	240	0	0	0	0	0

Table 2.1: The values of $|P^{\text{id}}(k, 4, d)|$ show how many elements $P \in \mathcal{P}(d)_4$ are unions of exactly k elements from the partitions $\mathcal{P}(d+3)$ for $k = 1, 2, \dots, 4!$ and $d = 1, 2, \dots, 5$.

2.3.3 Conclusions

To answer when the permutation entropy and the KS entropy coincide (see Theorem 8) we compared the partitions $\mathcal{P}^{\mathbf{X}}(d+n-1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$ for the general and for the one-dimensional case by using pure combinatorics. We conclude that pure combinatorics does not provide a direct answer to our question, but the formulated results (see Corollary 9, Propositions 10-12, Examples 7-8) are interesting and could be further investigated. In particular, solving the following open problems is of interest.

Problem 1. Find a general formula for $|P^{\mathbf{X}}(k, n, d)|$ (see (2.13)) or find functions $q_1(k, n, d)$ and $q_2(k, n, d)$ such that $q_1(k, n, d) \leq |P^{\mathbf{X}}(k, n, d)| \leq q_2(k, n, d)$ for $k = 1, 2, \dots, n!$, for $n, d \in \mathbb{N}$.

Problem 2. For what dynamical systems does statement (i) from Corollary 9 hold?

2.4 Comparing ordinal partitions for mixing interval maps

In this section we consider and discuss an approach from [UUK13] to comparing the partitions $\mathcal{P}(d+n-1)$ and $\mathcal{P}(d)_n$ for Ω being an interval in \mathbb{R} . We obtain an upper bound of the difference $H(\mathcal{P}(d+n-1)) - H(\mathcal{P}(d)_n)$ for mixing T , which still does not provide (2.11), but sheds some new light on the general problem of equality between the KS and the permutation entropy in the one-dimensional case. However, we are not sure that mixing is an important property for (2.11), because mixing is not a necessary condition for (2.11). For instance, it holds $H(\mathcal{P}(d+n-1)) - H(\mathcal{P}(d)_n) = 0$ for ergodic but non-mixing irrational rotation.

For the following discussion we adapt the definition (2.18) of the set V_{Π_d} , which is

“in charge” of the difference $H(\mathcal{P}(d+1)_{n-1}) - H(\mathcal{P}(d)_n)$, to the case $\Omega \subset \mathbb{R}$:

$$V_d = \left\{ \omega \in \mathbb{R} \mid \omega < T^d(\omega), T^l(\omega) \notin (\omega, T^d(\omega)) \text{ for all } l = 1, \dots, d-1 \right\} \\ \cup \left\{ \omega \in \mathbb{R} \mid \omega \geq T^d(\omega), T^l(\omega) \notin [T^d(\omega), \omega] \text{ for all } l = 1, \dots, d-1 \right\}. \quad (2.24)$$

We present now the main results of this section, Theorems 13, 14, and Corollary 15.

Theorem 13. *If T is mixing, then for all $\varepsilon > 0$ there exists some $d_\varepsilon \in \mathbb{N}$ such that for all $d > d_\varepsilon$*

$$\mu(V_d) < \varepsilon. \quad (2.25)$$

(The proof is given in Subsection 2.6.3.) Note that the ergodicity of the map T is not enough for (2.25). For instance, Theorem 13 does not hold for an ergodic but non-mixing irrational rotation. Theorem 14 provides a tool for comparing the “successive” partitions $\mathcal{P}(d+1)_{n-1}$ and $\mathcal{P}(d)_n$.

Theorem 14. *For all $n \in \mathbb{N} \setminus \{1\}$ and $d \in \mathbb{N}$ it holds*

$$H(\mathcal{P}(d+1)_{n-1}) - H(\mathcal{P}(d)_n) \leq \ln 2(n-1)\mu(V_{d+1}). \quad (2.26)$$

(The proof is given in Subsection 2.4.2.) Putting together Theorem 13 and 14, one obtains a more explicit variant of (2.26):

Corollary 15. *If T is mixing, then for all $\varepsilon > 0$ there exists some $d_\varepsilon \in \mathbb{N}$ such that for all $d \geq d_\varepsilon, n \in \mathbb{N} \setminus \{1\}$ it holds*

$$H(\mathcal{P}(d+1)_{n-1}) - H(\mathcal{P}(d)_n) < (n-1)\varepsilon.$$

Coming back to the partitions $\mathcal{P}(d+n-1)$ and $\mathcal{P}(d)_n$, in Subsection 2.4.3 we obtain the following upper bound for $H(\mathcal{P}(d+n-1)) - H(\mathcal{P}(d)_n)$, compare with (2.11):

$$H(\mathcal{P}(d+n-1)) - H(\mathcal{P}(d)_n) \leq \ln 2 \sum_{i=1}^{n-1} (n-i)\mu(V_{d+i}). \quad (2.27)$$

Subsection 2.4.1 illustrates the relation between the (n, d) -words, ordinal patterns and the set V_d . In Subsection 2.4.2 we focus on the partitions $\mathcal{P}(d+1)_{n-1}$ and $\mathcal{P}(d)_n$ and prove Theorem 14. Subsection 2.4.3 is devoted to the relation of the partitions $\mathcal{P}(d+n-1)$ and $\mathcal{P}(d)_n$ and provides (2.27). We make conclusions and state the open questions in Subsection 2.4.4.

2.4.1 Ordinal patterns of order $(d + n - 1)$ and (n, d) -words

Figure 2.4 illustrates a segment $(\omega, T(\omega), \dots, T^5(\omega))$ of some orbit (a) and the corresponding (5, 1)-, (4, 2)-, (3, 3)-, (2, 4)- and (1, 5)-words (b).

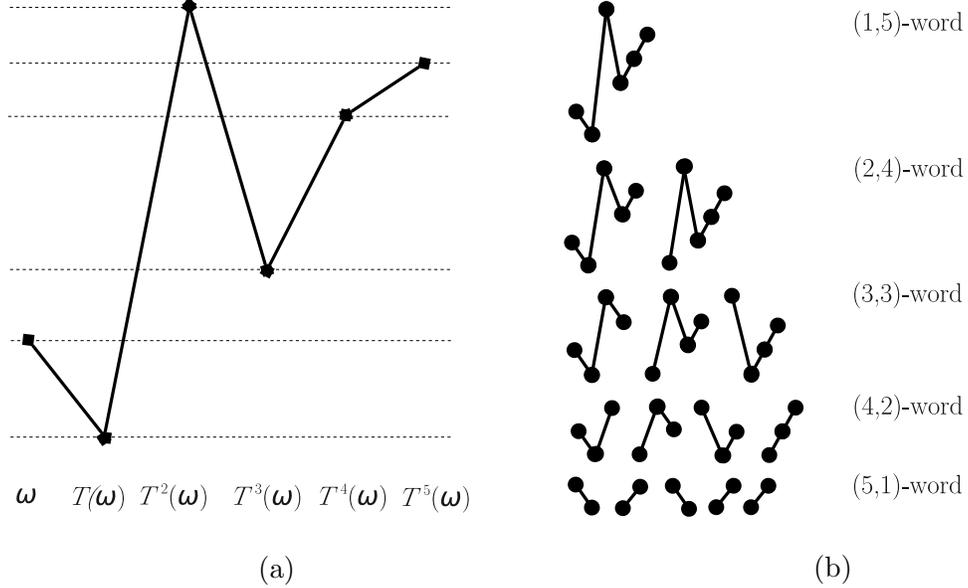


Figure 2.4: Representation of the segment of the orbit (a) by (n, d) -words (b)

Upon moving from (1, 5)- to (5, 1)-words one loses some information about the ordering of the iterates of T . For example, the (3, 3)-word determines the relation

$$\omega < T^3(\omega),$$

but in the (4, 2)-word this relation is already lost. It either holds $\omega \geq T^3(\omega)$ or $\omega < T^3(\omega)$. On the other hand, one does not lose the relation

$$\omega < T^4(\omega)$$

when moving from the (2, 4)-word to the (3, 3)-word, although ω and $T^4(\omega)$ are in different patterns of the (3, 3)-word. The reason for this is the existence of the intermediate iterate $T^3(\omega)$ with

$$\omega < T^3(\omega) < T^4(\omega).$$

More generally, if there is some intermediate iterate $T^l(\omega)$ with $\omega < T^l(\omega) < T^{d+1}(\omega)$ or $T^{d+1}(\omega) \leq T^l(\omega) \leq \omega$, the relation between ω and $T^{d+1}(\omega)$ is not lost upon moving from $(1, d + 1)$ - to $(2, d)$ -words, and is lost otherwise. Therefore, the set V_{d+1} (see (2.24)) consists of all ω , for which the relation between ω and $T^{d+1}(\omega)$ is lost upon moving from $(1, d + 1)$ - to $(2, d)$ -words. More precisely, the set V_{d+1} is a union of the sets of the partition $\mathcal{P}(d + 1)$ that are proper subsets of some sets of the partition $\mathcal{P}(d)_2$.

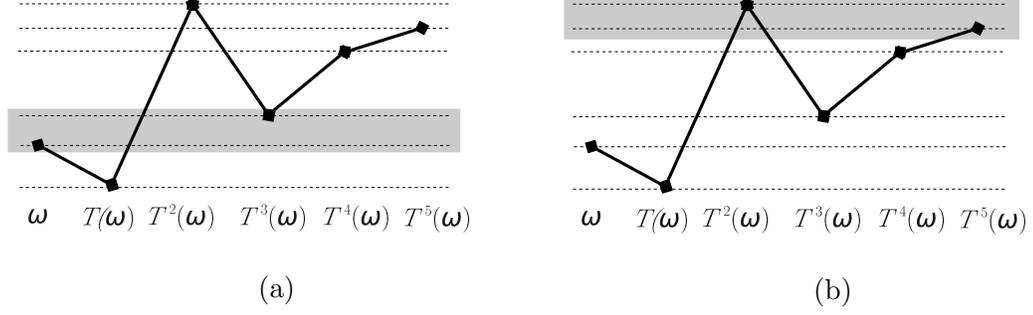


Figure 2.5: $\omega \in V_3$ (a), $T^2(\omega) \in V_3$ (b)

Figure 2.5 illustrates $\omega, T^2(\omega) \in V_3$ for our example.

In the following subsection we compare the partitions $\mathcal{P}(d)_n$ and $\mathcal{P}(d+1)_{n-1}$ by means of the set V_{d+1} .

2.4.2 The “neighboring” partitions $\mathcal{P}(d+1)_{n-1}$ and $\mathcal{P}(d)_n$

Upon moving from $(n-1, d+1)$ - to (n, d) -words, for $i = 0, 1, \dots, n-2$ the relation between $T^i(\omega)$ and $T^{d+i+1}(\omega)$ is lost iff $T^i(\omega) \in V_{d+1}$. Therefore, if $V_{d+1} \neq \emptyset$, then the partition $\mathcal{P}(d+1)_{n-1}$ is properly finer than the partition $\mathcal{P}(d)_n$. The following holds:

Proposition 16. *Given non-empty $P \in \mathcal{P}(d)_n$, let*

$$m(P) = \#\{0 \leq l \leq n-2 \mid P \subset T^{-l}(V_{d+1})\}. \quad (2.28)$$

Then there exist $2^{m(P)}$ sets $P_1, P_2, \dots, P_{2^{m(P)}} \in \mathcal{P}(d+1)_{n-1}$ with

$$P_1 \cup P_2 \cup \dots \cup P_{2^{m(P)}} = P.$$

Proof. Given $P \in \mathcal{P}(d)_n$, the corresponding (n, d) -word determines the same dynamics for all $\omega \in P$, and for $l = 0, 1, \dots, n-2$ it holds either

$$P \subset T^{-l}(V_{d+1}) \quad \text{or} \quad (2.29)$$

$$P \cap T^{-l}(V_{d+1}) = \emptyset. \quad (2.30)$$

(Note that $V_{d+1} \neq \emptyset$ since $P \in \mathcal{P}(d)_n$ is non-empty by assumption.) If $V_{d+1} = \emptyset$ then $m(P) = 0$. For each l with (2.29) and all $\omega \in P$, it holds either $T^l(\omega) < T^{d+l+1}(\omega)$ or $T^{d+l+1}(\omega) \leq T^l(\omega)$ providing a partitioning of P into two subsets. We are done since there are exactly m such partitionings. \square

Example 9. Figure 2.6 illustrates Proposition 16.

For $\omega \notin V_2 \cup T^{-1}(V_2)$ the obtained $(3, 1)$ -word is not split and contains the same information about the ordering as $2^0 = 1$ $(2, 2)$ -word (a), for $\omega \in V_2$ the $(3, 1)$ -word is split into $2^1 = 2$ $(2, 2)$ -words (b) and for $\omega \in V_2 \cap T^{-1}(V_2)$ the $(3, 1)$ -word is split into $2^2 = 4$ $(2, 2)$ -words (c).

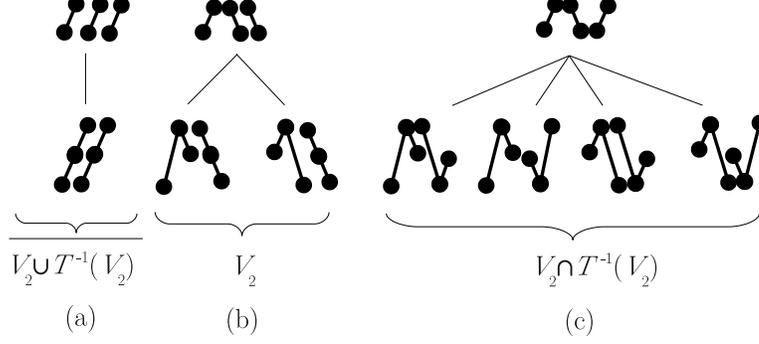


Figure 2.6: From (3, 1)- to (2, 2)-words. $\overline{V_2 \cup T^{-1}(V_2)}$ in (a) stands for the complement of $V_2 \cup T^{-1}(V_2)$

Let $m(P)$ be determined as in Proposition 16 for each $P \in \mathcal{P}(d)_n$. Since if $V_{d+1} \neq \emptyset$ then for each P it holds either (2.29) or (2.30), it follows

$$\sum_{j=0}^{n-2} \mu(T^{-j}(V_{d+1})) = \sum_{j=0}^{n-2} \sum_{P \in \mathcal{P}(d)_n} \mu(T^{-j}(V_{d+1}) \cap P) = \sum_{P \in \mathcal{P}(d)_n} m(P) \mu(P). \quad (2.31)$$

Therefore, by Proposition 16 and (2.31) one obtains an upper bound for $H(\mathcal{P}(d+1)_{n-1}) - H(\mathcal{P}(d)_n)$ in the following way:

$$\begin{aligned} H(\mathcal{P}(d+1)_{n-1}) - H(\mathcal{P}(d)_n) &= - \sum_{P' \in \mathcal{P}(d+1)_{n-1}} \mu(P') \ln \mu(P') + \sum_{P \in \mathcal{P}(d)_n} \mu(P) \ln \mu(P) \\ &\leq \sum_{P \in \mathcal{P}(d)_n} \left(\mu(P) \ln \mu(P) - 2^{m(P)} \frac{\mu(P)}{2^{m(P)}} \ln \frac{\mu(P)}{2^{m(P)}} \right) \\ &= \ln 2 \sum_{P \in \mathcal{P}(d)_n} m(P) \mu(P) \\ &= \ln 2 \sum_{j=0}^{n-2} \mu(T^{-j}(V_{d+1})) = \ln 2(n-1) \mu(V_{d+1}). \end{aligned} \quad (2.32)$$

Inequality (2.32) provides the proof of Theorem 14.

2.4.3 The partitions $\mathcal{P}(d)_n$ and $\mathcal{P}(d+n-1)$

Here we move from (n, d) -words to $(1, d+n-1)$ -words (i.e. ordinal patterns of order $(d+n-1)$). At this point we cannot definitely say into how many (n, d) -words a $(1, d+n-1)$ -word is split in dependence on the sets $V_{d+1}, \dots, V_{d+n-1}$ (see also Example 7).

Example 10. Figure 2.7 illustrates a (3, 1)-word with the same information as in the (1, 3)-word (a), other two (3, 1)-words are split into three and five (1, 3)-words ((b) and (c), respectively).

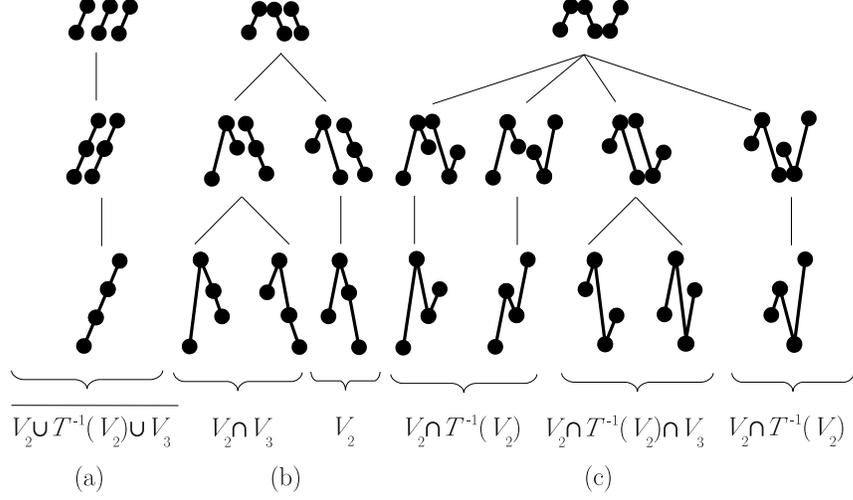


Figure 2.7: From (3,1)- to (1,3)-words. $\overline{V_2 \cup T^{-1}(V_2) \cup V_3}$ in (a) stands for the complement of $V_2 \cup T^{-1}(V_2) \cup V_3$

One obtains an upper bound for $H(\mathcal{P}(d+n-1)) - H(\mathcal{P}(d)_n)$ by successive application of (2.32):

$$\begin{aligned}
H(\mathcal{P}(d+n-1)) - H(\mathcal{P}(d)_n) &= \sum_{i=1}^{n-1} (H(\mathcal{P}(d+n-i)_i) - H(\mathcal{P}(d+n-i-1)_{i+1})) \\
&\leq \ln 2 \sum_{i=1}^{n-1} i \mu(V_{d+n-i}) = \ln 2 \sum_{i=1}^{n-1} (n-i) \mu(V_{d+i}). \quad (2.33)
\end{aligned}$$

However, the upper bound given by (2.33) is a weak bound and does not provide directly the required condition (ii) for Theorem 8 as we will see in the following example.

Example 11. Assume that we have the equidistributed measure $\mu(P) = \frac{1}{(d+1)!}$ for all $P \in \mathcal{P}(d)$. Then $\mu(V_d) = \frac{2}{(d+1)}$ since there are $2d!$ ordinal patterns of order d in V_{Π_d} (see (2.19)). Therefore one can represent the upper bound from (2.33) as

$$\begin{aligned}
\ln 2 \sum_{i=1}^{n-1} (n-i) \mu(V_{d+i}) &= \ln 2 \sum_{i=1}^{n-1} \frac{2(n-i)}{d+i+1} \\
&= 2 \ln 2 \left(\sum_{i=1}^{\lfloor \frac{n-d-1}{2} \rfloor} \frac{n-i}{d+i+1} + \sum_{i=\lfloor \frac{n-d-1}{2} \rfloor + 1}^{n-1} \frac{n-i}{d+i+1} \right) \\
&\geq 2 \ln 2 \lfloor \frac{n-d-1}{2} \rfloor = \ln 2(n-d-1) > (n-1)\varepsilon
\end{aligned}$$

for all $\varepsilon > 0$ starting from some d and $n \gg d$. (Note that $\frac{n-i}{d+i+1} \geq 1$ for $i \leq \lfloor \frac{n-d-1}{2} \rfloor$.) This means that the upper bound from (2.33) is larger than required by (2.11).

In the following proposition we describe how the measure $\mu(V_d)$ should decrease with increasing d in order to provide the coincidence of the permutation entropy and the KS entropy for $\Omega \subset \mathbb{R}$.

Proposition 17. *Let Ω be an interval in \mathbb{R} , $\mathbf{X} = \mathbf{id}$. Then for the following statements (i) implies (ii).*

(i) *For all $\varepsilon > 0$ there exists some $d_0 \in \mathbb{N}$ such that for all $d > d_0$ there exists some $n_0 \in \mathbb{N}$ such that for all $n > n_0$ it holds*

$$\sum_{i=1}^{n-1} \frac{n-i}{n-1} \mu(V_{d+i}) \leq \varepsilon. \quad (2.34)$$

(ii) $h_\mu(T) = h_\mu^{\mathbf{X}}(T)$.

2.4.4 Conclusions and open questions

In this section we have bounded the difference of the entropies of the neighboring partitions $\mathcal{P}(d+1)_{n-1}$ and $\mathcal{P}(d)_n$ for $\Omega \subset (a, b)$ and a mixing map T (see Theorems 13, 14 and Corollary 15). However, the transition from the partitions $\mathcal{P}(d+1)_{n-1}$ and $\mathcal{P}(d)_n$ to the partitions of interest $\mathcal{P}(d+n-1)$ and $\mathcal{P}(d)_n$ (see Theorem 8) is combinatorially complicated as it is shown in Section 2.3.

The approach considered in this section and the condition formulated for coincidence of the permutation entropy and the KS entropy (Proposition 17) are interesting results for further investigation of the question when the entropies coincide. We think that solving the following problems could provide one step more in this direction.

Problem 3. For what dynamical systems does statement (i) from Proposition 17 hold?

Problem 4. Is the mixing property of the map T important for coincidence of Kolmogorov-Sinai entropy and permutation entropy?

Problem 5. Is there a better bound than (2.32) of the difference $H(\mathcal{P}(d+1)_{n-1}) - H(\mathcal{P}(d)_n)$ or a better bound than (2.33) of the difference $H(\mathcal{P}(d+n-1)) - H(\mathcal{P}(d)_n)$?

Note that the upper bound (2.33) is a weak bound and, probably, can be improved, since it is obtained by rough summing up (2.32).

2.5 Conclusions

In this chapter we have investigated when the KS entropy and the permutation entropy coincide. The basic approach to doing this, provided by Theorem 8, is comparing the partitions $\mathcal{P}^{\mathbf{X}}(d+n-1)$ and $\mathcal{P}^{\mathbf{X}}(d)_n$. We have formulated sufficient conditions for coincidence of permutation entropy and KS entropy in Propositions 9 and 17 (p. 18

and 28); and we have described the combinatorial relations between the partitions $\mathcal{P}^{\mathbf{X}}(d+1)_{n-1}$ and $\mathcal{P}^{\mathbf{X}}(d)_n$ (Propositions 10, 11). These results could be used for further comparing the permutation entropy and the KS entropy.

In particular, solving the following problems could give an insight to the relationship between permutation entropy and KS entropy.

- Problems 1-5 (see pages 22, 28).
- Is it possible to simplify the proof of Theorem 7 for piecewise strictly monotone maps using Theorem 8?
- Is there a map for which the KS entropy does not coincide with the permutation entropy?
 - Is it possible to find such a map in a similar way as in [Mis03]?
 - Is there such a mixing (one-dimensional) map?

2.6 Proofs

2.6.1 Proof of Theorem 8

We provide here the proof of Theorem 8 [KUU12]. First, we show two auxiliary lemmas.

Lemma 18. *For $h_\mu^{\mathbf{X}}(T)$ and $h_\mu(T)$ defined by (2.7) and (2.8) it holds*

$$h_\mu^{\mathbf{X}}(T) \geq h_\mu(T).$$

Proof. For $h_\mu(T) = 0$ clearly $h_\mu(T) \leq h_\mu^{\mathbf{X}}(T)$ holds. Then we consider $h_\mu(T) > \alpha > 0$ and show that $h_\mu^{\mathbf{X}}(T) \geq \alpha$. Since α can be chosen arbitrarily near to $h_\mu(T)$ this implies $h_\mu^{\mathbf{X}}(T) \geq h_\mu(T)$.

Given $\beta > 1$ with $h_\mu(T) > \beta\alpha$, by (2.8) there exists some $d \in \mathbb{N}$ and some $n_d \in \mathbb{N}$ with $\frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{n} > \beta\alpha$ for all $n \geq n_d$. Thus for all $n \geq \max\left\{n_d, \frac{d}{\beta-1}\right\}$ by (2.12) we obtain

$$\frac{H(\mathcal{P}^{\mathbf{X}}(d+n-1))}{d+n-1} \geq \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{d+n-1} \geq \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{(\beta-1)n+n-1} > \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{\beta n} > \alpha,$$

implying

$$h_\mu^{\mathbf{X}}(T) = \overline{\lim}_{n \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(d+n-1))}{d+n-1} \geq \alpha.$$

□

Now we show that $h_\mu(T) = h_\mu^{\mathbf{X}}(T)$ implies existence of the limit in the definition of the permutation entropy (2.7).

Lemma 19. Let $h_\mu(T) = h_\mu^{\mathbf{X}}(T)$. Then

$$h_\mu^{\mathbf{X}}(T) = \lim_{d \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(d))}{d}. \quad (2.35)$$

Proof. Given some $k \in \mathbb{N}$ for all $n \in \mathbb{N}$ it holds using (2.12)

$$\lim_{d \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(d))}{d} = \lim_{n \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(k+n-1))}{k+n-1} \geq \lim_{n \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(k)_n)}{k+n-1} = \lim_{n \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(k)_n)}{n}.$$

Therefore, we have

$$\lim_{d \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(d))}{d} \geq \lim_{k \rightarrow \infty} \lim_{n \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(k)_n)}{n} = h_\mu(T) = h_\mu^{\mathbf{X}}(T) = \overline{\lim}_{d \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(d))}{d},$$

which shows (2.35). \square

Proof of Theorem 8

In order to show equivalence of (i) and (ii), we consider statement (ii'), which is equivalent to (ii):

(ii') For each $\varepsilon > 0$ there exists some $d_\varepsilon \in \mathbb{N}$ such that for all $d \geq d_\varepsilon$ there is some $n_d \in \mathbb{N}$ with

$$H(\mathcal{P}^{\mathbf{X}}(d+n-1)) - H(\mathcal{P}^{\mathbf{X}}(d)_n) < (d+n-1)\varepsilon \text{ for all } n \geq n_d.$$

Proof. (ii) \Rightarrow (ii') is obvious. Let us show now (ii') \Rightarrow (ii). For $\varepsilon > 0$ there exists $d_\varepsilon \in \mathbb{N}$ such that for all $d > d_\varepsilon$ there exists $n_d > d$ with

$$H(\mathcal{P}^{\mathbf{X}}(d+n-1)) - H(\mathcal{P}^{\mathbf{X}}(d)_n) < (d+n-1)\frac{\varepsilon}{2} < (n-1+n-1)\frac{\varepsilon}{2} = (n-1)\varepsilon$$

for all $n \geq n_d$.

\square

(i) \Rightarrow (ii'). We are going to prove now that for all $\varepsilon > 0$ there exists some $M_\varepsilon \in \mathbb{R}$, $d_\varepsilon \in \mathbb{N}$ such that for all $d > d_\varepsilon$ there exists some n_d with

$$M_\varepsilon - \varepsilon < \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{d+n-1} \leq \frac{H(\mathcal{P}^{\mathbf{X}}(d+n-1))}{d+n-1} < M_\varepsilon \text{ for all } n > n_d, \quad (2.36)$$

which shows (ii'). Consider some $\varepsilon > 0$ and $M_\varepsilon := h_\mu(T) + \varepsilon/2$. By Lemma 19 and (2.8) there exists some $d_\varepsilon \in \mathbb{N}$ such that for all $d \geq d_\varepsilon$ it holds

$$\frac{H(\mathcal{P}^{\mathbf{X}}(d))}{d} < M_\varepsilon \quad (2.37)$$

and

$$h_\mu(T) - \frac{\varepsilon}{4} < \lim_{n \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{n}. \quad (2.38)$$

From (2.37) and (2.12) it follows for all $d \geq d_\varepsilon$

$$\frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{d+n-1} \leq \frac{H(\mathcal{P}^{\mathbf{X}}(d+n-1))}{d+n-1} < M_\varepsilon \text{ for all } n \in \mathbb{N},$$

which shows the right-hand side of (2.36).

Given $d \geq d_\varepsilon$, (2.37) and (2.38) imply existence of some $n_d \geq \frac{4M_\varepsilon(d-1)}{\varepsilon}$ with

$$\begin{aligned} h_\mu(T) - \frac{\varepsilon}{4} &< \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{n} = \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{d+n-1} + \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{d+n-1} \frac{d-1}{n} \\ &\leq \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{d+n-1} + M_\varepsilon \frac{d-1}{n} \\ &\leq \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{d+n-1} + \frac{\varepsilon}{4} \end{aligned}$$

for all $n \geq n_d$. Hence we have

$$M_\varepsilon - \varepsilon = h_\mu(T) - \frac{\varepsilon}{2} < \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{d+n-1}. \quad (2.39)$$

for all $n \geq n_d$. Putting (2.37) and (2.39) together we have (2.36) and we are done.

(ii') \Rightarrow (i). By (ii') for each $\varepsilon > 0$ there exists some $d_\varepsilon \in \mathbb{N}$ such that for all $d \geq d_\varepsilon$ there is some $n_d \in \mathbb{N}$ with

$$\frac{H(\mathcal{P}^{\mathbf{X}}(d+n-1))}{d+n-1} \leq \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{d+n-1} + \varepsilon$$

for all $n \geq n_d$. For $d \geq d_\varepsilon$ this implies

$$h_\mu^{\mathbf{X}}(T) = \overline{\lim}_{n \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(d+n-1))}{d+n-1} \leq \overline{\lim}_{n \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{d+n-1} + \varepsilon = \lim_{n \rightarrow \infty} \frac{H(\mathcal{P}^{\mathbf{X}}(d)_n)}{n} + \varepsilon,$$

hence by (2.8) we have $h_\mu^{\mathbf{X}}(T) \leq h_\mu(T) + \varepsilon$. This provides $h_\mu^{\mathbf{X}}(T) \leq h_\mu(T)$ for $\varepsilon \rightarrow 0$. Now (i) follows by Lemma 18.

2.6.2 Proof of Proposition 11

Proof. First, we fix some $m, d, n \in \mathbb{N}$ and we determine

$$\#\left\{ P' \in \mathcal{P}(d+1)_{n-1} \mid \bigcup_{i=1}^{2^m} P_i = P \text{ where } P' \subset P \text{ and } P \in \mathcal{P}(d)_n \right\}. \quad (2.40)$$

Then we divide (2.40) by 2^m in order to obtain the required amount (2.20) of elements $P \in \mathcal{P}(d)_n$ that are unions of exactly 2^m elements $P' \in \mathcal{P}(d+1)_{n-1}$. To count all $P_{\pi_1 \pi_2 \dots \pi_{n-1}} \in \mathcal{P}(d+1)_{n-1}$ satisfying (2.40), we consider two cases, namely, $\pi_1 \notin V_{\Pi_{d+1}}$ and $\pi_1 \in V_{\Pi_{d+1}}$.

The first case, $\pi_1 \notin V_{\Pi_{d+1}}$. There are $(d+2)! - 2(d+1)! = (d+1)!d$ different $\pi_1 \notin V_{\Pi_{d+1}}$. By Proposition 10 exactly m permutations from $\pi_1, \pi_2, \dots, \pi_{n-1}$ are in $V_{\Pi_{d+1}}$. Given π_i , there are 2 possibilities to choose $\pi_{i+1} \in V_{\Pi_{d+1}}$, because π_i

determines $(d + 1)$ from $(d + 2)$ entries of π_{i+1} and $\pi_{i+1} \in V_{\Pi_{d+1}}$ assumes either $\pi_{i+1} = (\dots, 0, d + 1, \dots)$ or $\pi_{i+1} = (\dots, d + 1, 0, \dots)$. Therefore there are 2^m choices of the m permutations from $V_{\Pi_{d+1}}$. Given π_i , there are d possibilities to choose $\pi_{i+1} \notin V_{\Pi_{d+1}}$, because $\pi_{i+1} \notin V_{\Pi_{d+1}}$ forbids both $\pi_{i+1} = (\dots, 0, d + 1, \dots)$ and $\pi_{i+1} = (\dots, d + 1, 0, \dots)$, i.e. there are d^{n-m-2} choices of $n - m - 2$ permutations from $\Pi_{d+1} \setminus V_{\Pi_{d+1}}$. Note that one can place m from $n - 2$ permutations $\pi_i \in V_{\Pi_{d+1}}$ in $\binom{n-2}{m}$ ways. So for $\pi_1 \notin V_{\Pi_{d+1}}$ there are

$$(d + 1)! d 2^m d^{n-m-2} \frac{(n-2)!}{m!(n-m-2)!} = (d + 1)! d^{n-m-1} 2^m \binom{n-2}{m} \quad (2.41)$$

elements $P_{\pi_1 \pi_2 \dots \pi_{n-1}}$ satisfying (2.40).

The second case, $\pi_1 \in V_{\Pi_{d+1}}$. It holds $\#\{\pi_1 \in V_{\Pi_{d+1}} \mid \pi_1 \in \Pi_{d+1}\} = 2(d + 1)!$. Analogously to the first case, there are 2^{m-1} choices of $m - 1$ permutations from $V_{\Pi_{d+1}}$ (we do not count π_1); and there are d^{n-m-1} choices of $n - m - 1$ permutations from $\Pi_{d+1} \setminus V_{\Pi_{d+1}}$. So, for $\pi_1 \in \Pi_{d+1}$ the number of elements $P_{\pi_1 \pi_2 \dots \pi_{n-1}}$ satisfying (2.40) is given by

$$(d + 1)! d^{n-m-1} 2^m \binom{n-2}{m}. \quad (2.42)$$

Summing up (2.41) and (2.42) and dividing it by 2^m we are done. \square

2.6.3 Proof of Theorem 13

Lemma 20. ([UUK13]) *Let T be ergodic. Given an interval $A \subset \Omega$ and $d \in \mathbb{N} \setminus \{1\}$, let $\tilde{V}_d = \tilde{V}_d(A)$ be the set of points $\omega \in A$ for which at least one of two following conditions holds:*

$$T^l(\omega) \notin \{a \in A \mid a < \omega\} \text{ for all } l = 1, \dots, d - 1, \quad (2.43)$$

$$T^l(\omega) \notin \{a \in A \mid a > \omega\} \text{ for all } l = 1, \dots, d - 1. \quad (2.44)$$

Then for all $\varepsilon > 0$ there exists some $d_\varepsilon \in \mathbb{N}$ such that $\mu(\tilde{V}_d) < \varepsilon$ for all $d > d_\varepsilon$.

Proof. Let \tilde{V}_d^L be a set of points ω satisfying (2.43). Then it is sufficient to show $\mu(\tilde{V}_d^L) < \frac{\varepsilon}{2}$ for the corresponding d since for points satisfying (2.44) the proof is completely resembling.

Consider a partition $\{B_i\}_{i=1}^\infty$ of A into intervals B_i with the following properties:

$$(i) \mu(B_i) = \frac{\mu(A)}{2^i} \text{ for all } i \in \mathbb{N},$$

$$(ii) \text{ for all } i < j, \text{ and for all } \omega_1 \in B_i, \omega_2 \in B_j \text{ it holds } \omega_1 > \omega_2.$$

Since $\mu(\{\omega\}) = 0$ for all $\omega \in \Omega$ (if $\mu(\{\omega\}) > 0$ then $\mu(V_d) = 0$, see Remark 5), such partition always exists.

Define $D_{i,d} = \left\{ \omega \in B_i \mid T^l(\omega) \notin \bigcup_{j=i}^\infty B_j \text{ for all } l = 1, \dots, d - 1 \right\}$. It holds

$$\bigcup_{l=1}^{d-1} \left(D_{i,d} \cap T^{-l} \left(\bigcup_{j=i}^\infty B_j \right) \right) = \emptyset. \quad (2.45)$$

For all $d \in \mathbb{N}$, (2.45) provides $\tilde{V}_d^L \subseteq \bigcup_{i=1}^{\infty} D_{i,d}$ and, since $D_{i,d} \subseteq B_i$, it holds

$$\begin{aligned} \mu(\tilde{V}_d^L) &\leq \mu\left(\bigcup_{i=1}^{\infty} D_{i,d}\right) = \sum_{i=1}^{\infty} \mu(D_{i,d}) \leq \sum_{i=1}^k \mu(D_{i,d}) + \sum_{i=k+1}^{\infty} \mu(B_i) \\ &\leq \sum_{i=1}^k \mu(D_{i,d}) + \sum_{i=k+1}^{\infty} \frac{\mu(A)}{2^i} \leq \sum_{i=1}^k \mu(D_{i,d}) + \frac{\mu(A)}{2^k} \end{aligned} \quad (2.46)$$

for all $k \in \mathbb{N}$. On the other hand, by the ergodicity of T and by (2.45) we have

$$\mu\left(\bigcap_{d=1}^{\infty} D_{i,d}\right) \mu\left(\bigcup_{j=i}^{\infty} B_j\right) = \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{l=1}^{m-1} \mu\left(\bigcap_{d=1}^{\infty} D_{i,d} \cap T^{-l}\left(\bigcup_{j=i}^{\infty} B_j\right)\right) = 0.$$

Therefore, $\mu\left(\bigcup_{j=i}^{\infty} B_j\right) > 0$ implies $\mu\left(\bigcap_{d=1}^{\infty} D_{i,d}\right) = 0$ and, since $D_{i,1} \supseteq D_{i,2} \supseteq \dots$, it holds

$$\lim_{d \rightarrow \infty} \mu(D_{i,d}) = \mu\left(\bigcap_{d=1}^{\infty} D_{i,d}\right) = 0$$

for all $i \in \mathbb{N}$.

Now let $\varepsilon > 0$. Fix some $k \in \mathbb{N}$ with $k > \log_2 \frac{4}{\varepsilon}$ and d_ε with $\mu(D_{i,d}) < \frac{\varepsilon}{4k}$ for all $i = 1, 2, \dots, k$ and $d > d_\varepsilon$. Then, owing to (2.46), for $d > d_\varepsilon$ it holds

$$\mu(\tilde{V}_d^L) < k \frac{\varepsilon}{4k} + \frac{\varepsilon}{4} = \frac{\varepsilon}{2}$$

completing the proof. \square

Now we are coming to the proof of Theorem 13. Given $\varepsilon > 0$, let $r > \frac{3}{\varepsilon}$ and let $\{A_i\}_{i=1}^r$ be a partition of Ω into intervals A_i with $\mu(A_i) = \frac{1}{r}$. Furthermore, fix some $d_\varepsilon \in \mathbb{N}$ with

$$\mu\left(A_i \cap T^{-d}(A_i)\right) \leq \mu^2(A_i) + \frac{\varepsilon}{3r} = \frac{1}{r^2} + \frac{\varepsilon}{3r} \quad (2.47)$$

and

$$\mu(\tilde{V}_d) \leq \frac{\varepsilon}{3r} \quad (2.48)$$

for all $i = 1, 2, \dots, r$ and all $d > d_\varepsilon$, which is possible by the strong-mixing of T and by Lemma 20, respectively.

For $\omega \in V_d \cap A_i$ it is impossible that both $T^d(\omega) \notin A_i$ and $\omega \notin \tilde{V}_d(A_i)$, implying

$$\begin{aligned} V_d &= \bigcup_{i=1}^r (V_d \cap A_i) \subset \bigcup_{i=1}^r \left((A_i \cap T^{-d}(A_i)) \cup \tilde{V}_d(A_i) \right) \\ &= \bigcup_{i=1}^r (A_i \cap T^{-d}(A_i)) \cup \bigcup_{i=1}^r \tilde{V}_d(A_i). \end{aligned}$$

From this, (2.47), and (2.48), one obtains

$$\begin{aligned} \mu(V_d) &\leq \sum_{i=1}^r \mu\left(A_i \cap T^{-d}(A_i)\right) + \sum_{i=1}^r \mu\left(\tilde{V}_d(A_i)\right) \\ &\leq r \left(\frac{1}{r^2} + \frac{\varepsilon}{3r} \right) + \frac{\varepsilon}{3} < \varepsilon. \end{aligned}$$

Remark 5. Note that $\mu(\{\omega\}) > 0$ implies that ω is a periodic point, i.e. $\mu(\{\omega\}) = 1$. However, such $\omega \notin V_d$ for all $d \in \mathbb{N}$ since $\omega \in P_{(0,1,\dots,d)}$, i.e. $\mu(V_d) = 0$.

Chapter 3

Comparing practical complexity measures

In this chapter we compare the empirical permutation entropy (ePE) [BP02] with two widely-used practical measures of complexity, the approximate entropy (ApEn) [Pin91] and the sample entropy (SampEn) [RM00], in order to point out their advantages and drawbacks when applying to real-world data.

In Figure 3.1 we present the values of ePE, ApEn, and SampEn computed from the orbits of the logistic map $T_{LM} : [0, 1] \leftrightarrow$, given by $T_{LM}(x) = Ax(1 - x)$, in dependence on the parameter $A \in [3.5, 4]$ in comparison with the Lyapunov exponent¹ (LE). (The length of the orbit is 10^4 and the step between the values of $A \in [3.5, 4]$ is $5 \cdot 10^{-4}$. See Section 3.2 for description of the parameters for the entropies.)

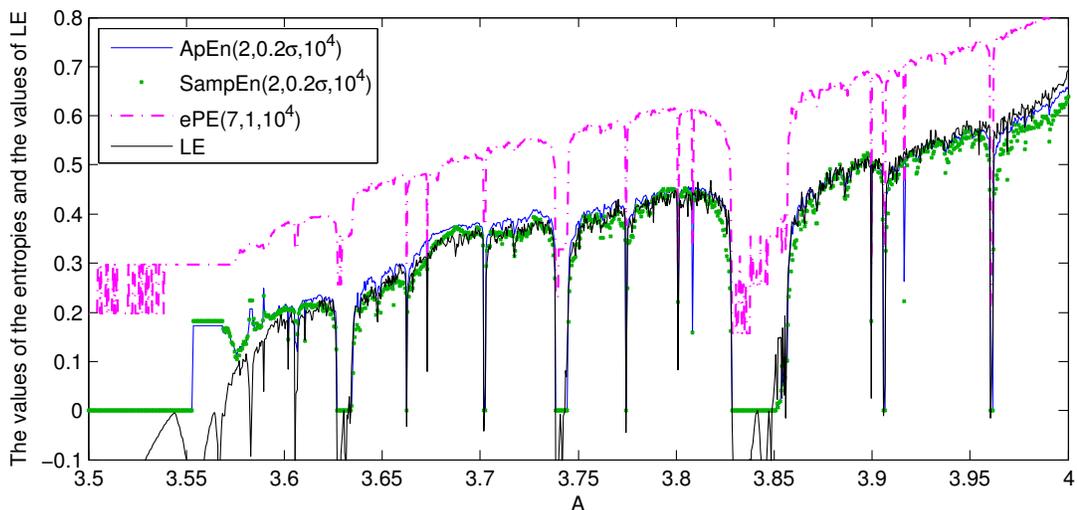


Figure 3.1: The values of approximate entropy (ApEn), sample entropy (SampEn) and empirical permutation entropy (ePE), computed from the orbits of the logistic map, in comparison with the values of Lyapunov exponent (LE)

¹See Subsection 2.1.2 for the relationship between the KS entropy and the Lyapunov exponent.

One can see that the entropies behave very similar to the values of LE (though the ePE values are constantly higher than the LE values). This motivates us to study the entropies in more details, especially for real-world data.

The rest of the chapter is organized as follows. We consider theoretical underpinnings for ApEn and SampEn in Section 3.1. We define the entropies and discuss their parameters in Section 3.2. Further in Section 3.3 we investigate practical properties of the complexity measures, namely, robustness to monotone transformations (Subsection 3.3.2), sensitivity to the length of a time series (Subsection 3.3.3), ability to correctly estimate high complexity of a time series (Subsection 3.3.4), robustness with respect to noise (Subsection 3.3.5), in particular, we introduce a modified ePE which is more robust than ePE with respect to noise or abnormal deviations (Subsection 3.3.6), and computational and storage requirements (Subsection 3.3.7). We summarize the results of comparison of the entropies and provide some hints for their application to real-world data in Section 3.4.

3.1 Theoretical underpinnings of approximate and sample entropy

In this section we study the theoretical underpinnings of the approximate entropy and the sample entropy. It is important to understand what they actually measure from a time series. Both entropies are stemming from the concepts of Rényi [Rén61, Rén70], Grassberger and Procaccia [GP83b, GP83a], Takens [Tak81] and Eckmann and Ruelle [ER85]. We show here that the approximate entropy is an estimate of the Eckmann-Ruelle entropy, proposed in [ER85] on the basis of the ideas from [GP83a, GP83b] for estimating the Kolmogorov-Sinai entropy, and the sample entropy is an estimate of the $H_2(T)$ -entropy, introduced, according to [BT11], in [Tak81]. Then we show the relationship between the Eckmann-Ruelle entropy, the $H_2(T)$ -entropy and the Kolmogorov-Sinai (KS) entropy (see Section 2.1 for more details about the KS entropy).

In Subsection 3.1.1 we recall the definitions of the correlation integral, which is the basic notion of this section, the correlation entropy [TV98], which we need for some proofs, and the $H_2(T)$ -entropy. In Subsection 3.1.2 we describe relationship between the $H_2(T)$ -entropy, the correlation entropy and the KS entropy. Then we consider the Eckmann-Ruelle entropy and its relationship to the KS entropy in Subsection 3.1.3. Subsection 3.1.4 is devoted to the estimation of $H_2(T)$ -entropy from a finite time series.

3.1.1 Correlation entropy and $H_2(T)$ -entropy

The correlation integral was originally defined in [GP83a]. Given a state space Ω , one can interpret the correlation integral as the mean probability of two points $\omega, \omega' \in \Omega$ being closer than a distance r .

Definition 12. Let (Ω, ρ) be a metric space with a Borel probability measure μ . Then the *correlation integral* is given by

$$C(r) = (\mu \times \mu)\{(\omega, \omega') \in \Omega \times \Omega \mid \rho(\omega, \omega') < r\}. \quad (3.1)$$

This definition is equivalent to the following

$$C(r) = \int_{\Omega} \mu(B(\omega, r)) d\mu(\omega) \text{ with} \\ B(\omega, r) = \{\omega' \in \Omega \mid \rho(\omega, \omega') < r\}.$$

Indeed, by Fubini's theorem [Cho00], it holds

$$\int_{\Omega \times \Omega} 1_{\delta_r}(\omega, \omega') d(\mu \times \mu) = \int_{\Omega} \left(\int_{\Omega} 1_{\delta_r}(\omega, \omega') d\mu(\omega') \right) d\mu(\omega) = \int_{\Omega} \mu(B(\omega, r)) d\mu(\omega), \\ \text{with } 1_{\delta_r} = \begin{cases} 1, & \rho(\omega, \omega') < r, \\ 0, & \text{otherwise.} \end{cases}$$

Further, we follow [BT11] when considering the concepts of the $H_2(T)$ -entropy and the correlation entropy. In order to introduce these entropies, we consider a dynamics T on the space (Ω, ρ) . On the basis of the dynamics T one defines a metric ρ_k , the ball $B_k(\omega, r)$ with a radius r around $\omega \in \Omega$ with respect to ρ_k , and the correlation integral $C(k, r)$ over the balls $B_k(\omega, r)$ in Ω :

$$\rho_k(\omega, \omega') = \max_{i=0,1,\dots,k-1} \rho(T^i(\omega), T^i(\omega')), \quad (3.2) \\ B_k(\omega, r) = \{\omega' \in \Omega \mid \rho_k(\omega, \omega') < r\}, \\ C(k, r) = \int_{\Omega} \mu(B_k(\omega, r)) d\mu(\omega).$$

Definition 13. Given a dynamical system $(\Omega, \mathbb{B}(\Omega), \mu, T)$ on the compact metric space (Ω, ρ) , a Borel probability measure μ and a continuous map $T : \Omega \leftrightarrow$, the $H_2(T)$ -entropy and the *correlation entropy* $\text{CE}(T, 1)$ are defined by

$$H_2(T) = \overline{\lim}_{r \rightarrow 0} \overline{\lim}_{k \rightarrow \infty} -\frac{1}{k} \ln \int_{\Omega} \mu(B_k(\omega, r)) d\mu(\omega), \quad (3.3)$$

$$\text{CE}(T, 1) = \lim_{r \rightarrow 0} \lim_{k \rightarrow \infty} -\frac{1}{k} \int_{\Omega} \ln \mu(B_k(\omega, r)) d\mu(\omega). \quad (3.4)$$

Note that the limit $r \rightarrow 0$ in (3.4) exists due to the monotonicity properties of $-\frac{1}{k} \int_{\Omega} \ln \mu(B_k(\omega, r)) d\mu(\omega)$, see [Ver00, Lemma 2.1] for details. The limit $k \rightarrow \infty$ in (3.4) exists by Lemma 2.14 from [Ver00].

Remark 6. The $H_2(T)$ -entropy and correlation entropy $\text{CE}(T, 1)$ are related to the Rényi entropies, introduced in [Rén61, Rén70]. According to [TV02], there were attempts in [HP83, GP84, ER85] to generalize the Rényi entropies of order q for the dynamical systems since the Rényi entropies were successfully applied in information theory.

However, in [TV98] it was shown that “the Rényi entropies of order q , $q \in \mathbb{R}$, are equal to either plus infinity ($q < 1$), or to the measure-theoretic (Kolmogorov-Sinai) entropy ($q \geq 1$)” [TV02]. Then, inspired by [ER85, GP84, Tak81], where the Rényi approach was used, Takens and Verbitskiy introduced in [TV98] the correlation entropy $\text{CE}(T, q)$ of order q (we consider in Definition 13 only the case $q = 1$), and very similar $H_2(T)$ -entropy has been introduced, according to [BT11], in [Tak81]. We refer also to Verbitskiy’s dissertation [Ver00] for an interesting study of properties of the correlation entropies $\text{CE}(T, q)$.

3.1.2 Relationship between $H_2(T)$ -entropy, correlation entropy and KS entropy

In this subsection, on the basis of the ideas and results from [TV98, Ver00, BT11], in Theorem 22 we show the relationship between the $H_2(T)$ -entropy, the correlation entropy and the KS entropy. In order to prove Theorem 22 we recall the Brin-Katok theorem.

Theorem 21 (Brin-Katok theorem [BK83]). *Given a dynamical system $(\Omega, \mathbb{B}(\Omega), \mu, T)$ on a compact metric space (Ω, ρ) with a continuous map $T : \Omega \leftrightarrow$ preserving a non-atomic² Borel probability measure μ . Then it holds:*

$$\int_{\Omega} \lim_{r \rightarrow 0} \overline{\lim}_{k \rightarrow \infty} -\frac{1}{k} \ln \mu(B_k(\omega, r)) d\mu(\omega) = \int_{\Omega} \lim_{r \rightarrow 0} \underline{\lim}_{k \rightarrow \infty} -\frac{1}{k} \ln \mu(B_k(\omega, r)) d\mu(\omega) = h_{\mu}(T). \quad (3.5)$$

Theorem 22. *Under the conditions of Theorem 21 it holds*

$$H_2(T) \leq \text{CE}(T, 1) = h_{\mu}(T). \quad (3.6)$$

Proof. First $H_2(T) \leq \text{CE}(T, 1)$ is provided by Jensen’s inequality:

$$\begin{aligned} \text{CE}(T, 1) &= \overline{\lim}_{r \rightarrow 0} \overline{\lim}_{k \rightarrow \infty} -\frac{1}{k} \int_{\Omega} \ln \mu(B_k(\omega, r)) d\mu(\omega) \\ &\geq \underset{\text{Jensen's inequality}}{\overline{\lim}_{r \rightarrow 0} \overline{\lim}_{k \rightarrow \infty} -\frac{1}{k} \ln \int_{\Omega} \mu(B_k(\omega, r)) d\mu(\omega)} = H_2(T). \end{aligned}$$

Now $\text{CE}(T, 1) = h_{\mu}(T)$ is shown, using the Fatou Lemma twice and the Brin-Katok formula (3.5):

$$\lim_{r \rightarrow 0} \overline{\lim}_{k \rightarrow \infty} \int_{\Omega} -\frac{1}{k} \ln \mu(B_k(\omega, r)) d\mu(\omega) \leq h_{\mu}(T) \leq \lim_{r \rightarrow 0} \underline{\lim}_{k \rightarrow \infty} \int_{\Omega} -\frac{1}{k} \ln \mu(B_k(\omega, r)) d\mu(\omega).$$

Therefore the following limit exists and is equal to the Kolmogorov-Sinai entropy:

$$\text{CE}(T, 1) = \lim_{r \rightarrow 0} \lim_{k \rightarrow \infty} \int_{\Omega} -\frac{1}{k} \ln \mu(B_k(\omega, r)) d\mu(\omega) = h_{\mu}(T).$$

□

²A measure μ is called *non-atomic* if for any measurable set A with $\mu(A) > 0$ there exists a measurable set $B \subset A$ such that $\mu(A) > \mu(B) > 0$

3.1.3 Eckmann-Ruelle's approximation of KS entropy

In this subsection we show the relationship between the Eckmann-Ruelle entropy and the KS entropy in order to explain where the approximate entropy is stemming from.

Theorem 23. *Under the conditions of Theorem 21 for an ergodic T , if for all $r \in \mathbb{R}_+$ the limit*

$$\lim_{k \rightarrow \infty} \lim_{n \rightarrow \infty} \frac{1}{n} \left(\sum_{j=0}^{n-1} \ln \mu(B_k(T^j(\omega), r)) - \sum_{j=0}^{n-1} \ln \mu(B_{k+1}(T^j(\omega), r)) \right) \quad (3.7)$$

exists then it holds

$$h_\mu(T) = \lim_{r \rightarrow 0} \lim_{k \rightarrow \infty} \lim_{n \rightarrow \infty} \frac{1}{n} \left(\sum_{j=0}^{n-1} \ln \mu(B_k(T^j(\omega), r)) - \sum_{j=0}^{n-1} \ln \mu(B_{k+1}(T^j(\omega), r)) \right). \quad (3.8)$$

Proof. First, we use Theorem 22 to represent the KS entropy in following way:

$$\begin{aligned} h_\mu(T) = \text{CE}(T, 1) &= \lim_{r \rightarrow 0} \lim_{k \rightarrow \infty} -\frac{1}{k} \int_{\Omega} \ln \mu(B_k(\omega, r)) d\mu(\omega) \\ &\text{for a.a. } \omega \in \Omega \text{ by Birkhoff's Ergodic Theorem} \\ &= \lim_{r \rightarrow 0} \lim_{k \rightarrow \infty} -\frac{1}{k} \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=0}^{n-1} \ln \mu(B_k(T^j(\omega), r)). \end{aligned}$$

Since the limit (3.7) exists, we obtain

$$\begin{aligned} &\lim_{r \rightarrow 0} \lim_{k \rightarrow \infty} \lim_{n \rightarrow \infty} \frac{1}{n} \left(\sum_{j=0}^{n-1} \ln \mu(B_k(T^j(\omega), r)) - \sum_{j=0}^{n-1} \ln \mu(B_{k+1}(T^j(\omega), r)) \right) \\ &= \lim_{r \rightarrow 0} \lim_{k \rightarrow \infty} \left(\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=0}^{n-1} \ln \mu(B_k(T^j(\omega), r)) - \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=0}^{n-1} \ln \mu(B_{k+1}(T^j(\omega), r)) \right) \\ &\stackrel{\text{by Cesaro's summation [Har91]}}{=} \lim_{r \rightarrow 0} \lim_{k \rightarrow \infty} -\frac{1}{k} \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=0}^{n-1} \ln \mu(B_k(T^j(\omega), r)) = h_\mu(T), \end{aligned}$$

which finishes the proof. \square

(Note, that in general it is not known whether limit (3.7) exists.)

Now recall that we observe $(x_t)_{t=1}^N$ with $x_t = X(T^t(\omega))$ via an observable $X : \Omega \rightarrow \mathbb{R}$ from a dynamical system (see Subsection 1.1). Then, for an ergodic T a term $\widehat{C}(i, k, r, (x_t)_{t=1}^N)$ is a natural estimate of $\mu(B_k(T^i(\omega), r))$:

$$\widehat{C}(i, k, r, (x_t)_{t=1}^N) = \frac{\#\left\{1 \leq j \leq N - k + 1 : \max_{l=0,1,\dots,k-1} |x_{i+l} - x_{j+l}| \leq r\right\}}{N - k + 1}, \quad (3.9)$$

see the following remark.

Remark 7. Further we come from the metric defined on the state space Ω by (3.2) to the maximum norm $\max_{l=0,1,\dots,k-1} |x_{i+l} - x_{j+l}|$ defined for the reconstruction vectors $(x_i, x_{i+1}, \dots, x_{i+k-1})$. This is possible by Takens' theorem which says that for any metric ρ on Ω , the orbit $\omega, T(\omega), \dots$ and the sequence of reconstruction vectors $(x_i, x_{i+1}, \dots, x_{i+k-1})$ are metrically equivalent (up to bounded distortion, see [BT11, Chapter 6] for more details).

Then the entropy proposed by Eckmann and Ruelle for a time series in [ER85] is given by:

$$\widehat{H}_{\text{ER}} = \lim_{r \rightarrow 0} \lim_{k \rightarrow \infty} \lim_{N \rightarrow \infty} (\Phi(k, r, (x_t)_{t=1}^N) - \Phi(k+1, r, (x_t)_{t=1}^N)), \quad (3.10)$$

$$\text{where } \Phi(k, r, (x_t)_{t=1}^N) = \frac{1}{N-k+1} \sum_{i=1}^{N-k+1} \ln(\widehat{C}(i, k, r, (x_t)_{t=1}^N)).$$

Now when comparing (3.10) with the representation of the KS entropy (3.8), one can see that the Eckmann-Ruelle entropy can be considered as an approximation of the KS entropy under the conditions of Theorem 23.

The approximate entropy, introduced in [Pin91] with the aim of measuring the complexity of a system by the observed time series, is, in fact, the estimate of the Eckmann-Ruelle entropy for finite $k, N \in \mathbb{N}$ and $r \in \mathbb{R}$ (compare (3.10)):

$$\text{ApEn}(k, r, (x_t)_{t=1}^N) = \Phi(k, r, (x_t)_{t=1}^N) - \Phi(k+1, r, (x_t)_{t=1}^N), \quad (3.11)$$

see Section 3.2 for further details regarding the approximate entropy.

3.1.4 Estimation of the $H_2(T)$ -entropy

In this subsection we consider estimation of the $H_2(T)$ -entropy and its relationship to the sample entropy on the basis of estimation of the correlation integral. An estimation of the correlation integral $C(k, r)$ is possible under certain generic conditions of Takens' reconstruction theorem (see [Tak81, BT11] for details).

Applying Birkhoff's ergodic theorem (see Subsection 1.2.1) twice we obtain:

$$\begin{aligned} C(k, r) &= \int_{\Omega} \mu(B_k(\omega, r)) d\mu(\omega) \quad \text{for a.a. } \omega_1 \in \Omega \quad \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=0}^{n-1} \mu(B_k(T^j(\omega_1), r)) \\ &= \lim_{\text{for a.a. } \omega_2 \in \Omega} \lim_{n \rightarrow \infty} \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} 1_{\delta_{r,k}}(T^i(\omega_1), T^j(\omega_2)), \end{aligned}$$

$$\text{with } 1_{\delta_{r,k}}(T^i(\omega_1), T^j(\omega_2)) = \begin{cases} 1, & \rho_k(T^i(\omega_1), T^j(\omega_2)) < r, \\ 0, & \text{otherwise.} \end{cases}$$

Then an estimate of the correlation integral of Ω from a time series $(x_i)_{i \in \mathbb{N}} = (X(T^i(\omega)))_{i \in \mathbb{N}}$ is given by [GP83b, BT11], see also Remark 7:

$$\widehat{C}(k, r, (x_t)_{t=1}^N) = \frac{2\#\left\{(i, j) : 1 \leq i < j \leq N - k + 1, \max_{l=0,1,\dots,k-1} |x_{i+l} - x_{j+l}| \leq r\right\}}{(N - k - 1)(N - k)}.$$
(3.12)

We refer to [BT11, Bor98] for further discussion and review of estimators of the correlation integral.

It was shown in [DK86] that for almost all $\omega \in \Omega$ for $(x_t)_{t \in \mathbb{N}} = (X(T^t(\omega)))_{t \in \mathbb{N}}$ it holds

$$\lim_{N \rightarrow \infty} \widehat{C}(k, r, (x_t)_{t=1}^N) = C(k, r).$$

Note that there are some limitations in the estimation of the correlation integral, which are related to the choice of the parameters r and k (see [Rue90, BT11]). For example, it is shown in [Rue90] that in order to obtain reliable estimates of the correlation integral one needs to take very long time series with a very small noise.

According to [BT11] the following estimate of $H_2(T)$ -entropy was proposed in [Tak81], and then applied in [GP83b, GP83a]:

$$\widehat{H}_2(k, r, (x_t)_{t=1}^N) = \ln \frac{\widehat{C}(k, r, (x_t)_{t=1}^N)}{\widehat{C}(k + 1, r, (x_t)_{t=1}^N)}.$$
(3.13)

The sample entropy, introduced in [RM00] with the aim of improving the approximate entropy, is, in fact, an estimate of $H_2(T)$ -entropy given by (3.13):

$$\text{SampEn}(k, r, (x_t)_{t=1}^N) = \widehat{H}_2(k, r, (x_t)_{t=1}^N) = \ln \frac{\widehat{C}(k, r, (x_t)_{t=1}^N)}{\widehat{C}(k + 1, r, (x_t)_{t=1}^N)}.$$

We consider the sample entropy in more details in Section 3.2.

3.2 Definitions of the entropies

3.2.1 Approximate entropy and sample entropy

In this subsection we recall the definitions of the approximate entropy (ApEn) and the sample entropy (SampEn) from [Pin91, RM00] and discuss their properties.

Definition 14. Given a time series $(x_t)_{t=1}^N$, a length $k \in \mathbb{N}$ of vectors to be compared and a tolerance $r \in \mathbb{R}$ for accepting similar vectors, the *approximate entropy* is defined as

$$\text{ApEn}(k, r, (x_t)_{t=1}^N) = \frac{\sum_{i=1}^{N-k+1} \ln \widehat{C}(i, k, r, (x_t)_{t=1}^N)}{N - k + 1} - \frac{\sum_{i=1}^{N-k} \ln \widehat{C}(i, k + 1, r, (x_t)_{t=1}^N)}{N - k},$$
(3.14)

where $\widehat{C}(i, k, r, (x_t)_{t=1}^N)$ is given by (3.9).

Definition 15. Given a time series $(x_t)_{t=1}^N$, a length $k \in \mathbb{N}$ of vectors to be compared and a tolerance $r \in \mathbb{R}$ for accepting similar vectors, the *sample entropy* is defined as

$$\text{SampEn}(k, r, (x_t)_{t=1}^N) = \ln \widehat{C}(k, r, (x_t)_{t=1}^N) - \ln \widehat{C}(k+1, r, (x_t)_{t=1}^N), \quad (3.15)$$

where $\widehat{C}(k, r, (x_t)_{t=1}^N)$ is given by (3.12).

Further we use short forms $\text{ApEn}(k, r, N)$ and $\text{SampEn}(k, r, N)$ instead of $\text{ApEn}(k, r, (x_t)_{t=1}^N)$ and $\text{SampEn}(k, r, (x_t)_{t=1}^N)$ when no confusion arises. Note that $\text{SampEn}(k, r, N)$ is undefined when either $\widehat{C}(k, r, N) = 0$ or $\widehat{C}(k+1, r, N) = 0$.

When computing ApEn by (3.14) one counts each vector as matching itself, because

$$\max_{l=0,1,\dots,k-1} |x_{i+l} - x_{i+l}| = 0 \leq r,$$

which introduces some bias in the result [RM00]. When computing SampEn one does not count self-matches due to $i < j$ in (3.15) which is more natural [RM00].

For application of both entropies the tolerance r is recommended to set to $r \in [0.1\sigma, 0.25\sigma]$, where σ is the standard deviation (SD) of a time series, the length of vectors to be compared is recommended to set to $k = 2$ [Pin91, Pin95, RM00].

In order to compare the complexities of two time series one has to fix k , r and N due to the significant variation of the values of $\text{ApEn}(k, r, N)$ and $\text{SampEn}(k, r, N)$ for different parameters [Pin95, RM00]. Apart from the aforementioned recommendations, there is no guidelines to specify k and r [HAEG09].

Applications

Approximate entropy and sample entropy have been applied in several cardiovascular studies [LRGM02, GGK12] (see for a review [AJK⁺06]), for quantifying the effect of anesthesia drugs on the brain activity [BRH00, BRR⁺00, JSK⁺08], for analysis of EEG data from patients with Alzheimer's disease [AHE⁺06], for separating sleep stages [AFK⁺05, BMC⁺05], for epileptic EEG analysis and epileptic seizures detection [KCAS05, LOR07, Oca09, JB12] and in other fields (see [RM00, CZYW09, YHS⁺13] for a review of applications). Costa, Goldberger et al. proposed also a multiscale variant of sample entropy in [CGP05].

3.2.2 Empirical permutation entropy

In this subsection the empirical permutation entropy (ePE) is defined and its properties are discussed.

First, we recall the definition of ordinal patterns which we need to introduce the empirical permutation entropy. Originally, ordinal patterns of order d were defined as permutations of the set $\{0, 1, \dots, d\}$ (see Definition 6, p. 13), but we use the following equivalent definition which provides simpler enumeration of ordinal patterns (see details in [KSE07]).

Definition 16. A delay vector $(x_t, x_{t-\tau}, \dots, x_{t-d\tau})$ is said to have an *ordinal pattern* $(i_1^\tau(t), i_2^\tau(t), \dots, i_d^\tau(t))$ of order $d \in \mathbb{N}$ and delay $\tau \in \mathbb{N}$, where for $l = 1, 2, \dots, d$

$$i_l^\tau(t) = \#\{r \in \{0, 1, \dots, l-1\} \mid x_{t-l\tau} \geq x_{t-r\tau}\}. \quad (3.16)$$

Simply speaking, each $i_l^\tau(t)$ codes how many points from $(x_t, x_{t-\tau}, \dots, x_{t-(l-1)\tau})$ are not larger than $x_{t-l\tau}$. Note that in Definition 6 the distance $\tau \in \mathbb{N}$ between the values in ordinal patterns is fixed to $\tau = 1$, so Definition 16 is more flexible for application to real-world data.

There are $(d+1)!$ ordinal patterns of order d , and one assigns to each of them a number from $\{0, 1, \dots, (d+1)! - 1\}$ in a one-to-one way by [KSE07]

$$n_d^\tau(t) = n_d^\tau((i_1^\tau(t), i_2^\tau(t), \dots, i_d^\tau(t))) = \sum_{l=1}^d i_l^\tau(t) \frac{(d+1)!}{(l+1)!}. \quad (3.17)$$

Definition 17. By the *empirical permutation entropy* (ePE) of order $d \in \mathbb{N}$ and of delay $\tau \in \mathbb{N}$ of a time series $(x_t)_{t=1}^N$ with $N \in \mathbb{N}$ one understands the quantity

$$\begin{aligned} \text{ePE}(d, \tau, (x_t)_{t=1}^N) &= -\frac{1}{d} \sum_{j=0}^{(d+1)!-1} p_j \ln p_j, \text{ where} \\ p_j &= \frac{\#\{i = d\tau + 1, d\tau + 2, \dots, N \mid n_d^\tau(i) = j\}}{N - d\tau} \text{ (with } 0 \ln 0 := 0\text{)}. \end{aligned}$$

Further we use a short form $\text{ePE}(d, \tau, N)$ instead of $\text{ePE}(d, \tau, (x_t)_{t=1}^N)$ when no confusion arises.

Empirical permutation entropy, originally introduced in [BP02] as a natural complexity measure for time series, is an estimate of the permutation entropy (see Chapter 2). Indeed, given an ergodic dynamical system $(\Omega, \mathbb{B}(\Omega), \mu, T)$, the empirical permutation entropy is computed from the distribution of ordinal patterns in the time series $(x_i)_{i \in \mathbb{N}_0} = (X(T^i(\omega)))_{i \in \mathbb{N}_0}$.

The higher the diversity of ordinal patterns of order d in a time series $(x_t)_{t=1}^N$ is, the larger the value of $\text{ePE}(d, \tau, (x_t)_{t=1}^N)$ is. It holds for $d, N, \tau \in \mathbb{N}$ that

$$0 \leq \text{ePE}(d, \tau, N) \leq \frac{\ln((d+1)!)}{d}, \quad (3.18)$$

which is restrictive for estimation of a large complexity as we show in Example 15 in Subsection 3.3.4 for EEG data. The choice of order d is rather simple. The larger d is, the better the estimate of complexity by the empirical permutation entropy is. On the other hand, too high d leads to an underestimation of the complexity of a system because due to the bounded length of a time series not all ordinal patterns representing the system can occur. In [AZS08],

$$5(d+1)! < N \quad (3.19)$$

is recommended. The choice of the delay τ is a bit complicated; in many applications $\tau = 1$ is used, however larger delays τ can provide additional information as we illustrate in Subsection 5.1.2 for EEG data (see also [RMW13] for the discussion of the choice of τ). Note also that an increase of a delay τ can lead to an increase of the values of $\text{ePE}(d, \tau, N)$, i.e. when increasing the delay τ one should mind the bound (3.18) (see Example 21 for EEG data in Subsection 5.1.2). We provide some hints for the choice of the delay τ for EEG analysis in Chapter 5.

Let us consider an example to give some insight into the meaning of a delay τ .

Example 12. In Figure 3.2 we present the values of $\text{ePE}(7, \tau, 10^6)$ computed from the orbit of the logistic map $T_{\text{LM}}(x) = Ax(1 - x)$ for $\tau = 1, 2, 4$ and $A \in [3.5, 4]$. (Length of the orbit is 10^6 , step between the values of A is $5 \cdot 10^{-4}$.)

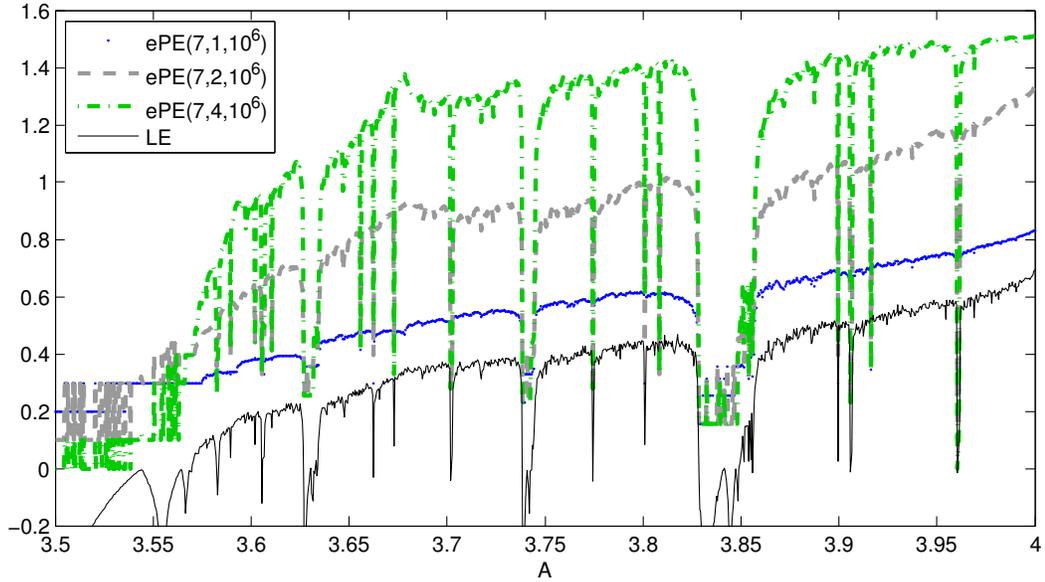


Figure 3.2: The values of empirical permutation entropy $\text{ePE}(7, \tau, 10^6)$ computed from the orbit of the logistic map for $\tau = 1, 2, 4$

By Theorem 7, the KS entropy $h_\mu(T_{\text{LM}})$ and the permutation entropy $h_\mu^{\mathbf{X}}(T_{\text{LM}})$ for $\mathbf{X} = \text{id}$ coincide as well as the KS entropy $h_\mu(T_{\text{LM}}^\tau)$ and the permutation entropy $h_\mu^{\mathbf{X}}(T_{\text{LM}}^\tau)$ for $\mathbf{X} = \text{id}$ coincide. Hence, by the entropy properties (see [Cho00, Theorem 8.8]) it holds

$$h_\mu^{\mathbf{X}}(T_{\text{LM}}^\tau) = h_\mu(T_{\text{LM}}^\tau) = \tau h_\mu(T_{\text{LM}}) = \tau h_\mu^{\mathbf{X}}(T_{\text{LM}}). \quad (3.20)$$

Then, since T_{LM} is an ergodic transformation, $\text{ePE}(d, \tau, (x_t)_{t=1}^N)$ for $\tau = 1$ is an estimate of the permutation entropy $h_\mu^{\mathbf{X}}(T_{\text{LM}})$ for $\mathbf{X} = \text{id}$. For $\tau > 1$, $h_\mu^{\mathbf{X}}(T_{\text{LM}}^\tau)$ can be

estimated not only from one orbit but from τ different orbits:

$$\begin{aligned}
& x, T_{\text{LM}}^\tau(x), T_{\text{LM}}^{2\tau}(x), \dots, T_{\text{LM}}^{d\tau}(x), \dots \\
& T_{\text{LM}}(x), T_{\text{LM}}^{\tau+1}(x), T_{\text{LM}}^{2\tau+1}(x), \dots, T_{\text{LM}}^{d\tau+1}(x), \dots \\
& \dots \\
& T_{\text{LM}}^{\tau-1}(x), T_{\text{LM}}^{\tau+\tau-1}(x), T_{\text{LM}}^{2\tau+\tau-1}(x), \dots, T_{\text{LM}}^{d\tau+\tau-1}(x), \dots
\end{aligned}$$

However, due to ergodicity, one still can say that $\text{ePE}(d, \tau, (x_t)_{t=1}^N)$ for $\tau > 1$ is an estimate of the permutation entropy $h_\mu^{\mathbf{X}}(T_{\text{LM}}^\tau)$ for $\mathbf{X} = \mathbf{id}$. Then empirical permutation entropy should increase proportionally to τ due to (3.20). In Figure 3.2 one can see this.

Empirical permutation entropy has been applied for detecting and visualizing EEG changes related to epileptic seizures (e.g., [CTG⁺04, KL03, LOR07, ODR10, LYLO14, KUU14]), for distinguishing brain states related to anesthesia [OSD08, LLL⁺10], for discriminating sleep stages in EEG data [NG11, KUU14], for analyzing and classifying heart rate variability data [FPSH06, BQMS12, PBL⁺12, GGK12], and for financial, physical and statistical time series analysis (see [Ami10, AK13] for a review of applications).

3.3 Comparing the practical properties of the entropies

In this section we compare the practical properties of the approximate entropy (ApEn), the sample entropy (SampEn) and the empirical permutation entropy (ePE). We illustrate the properties with the examples from chaotic maps, because chaotic maps are widely used for modeling real-world data.

Subsection 3.3.1 contains a description of the considered chaotic maps. Then in Subsection 3.3.2 it is shown that ePE is more robust to strictly monotone transformations of time series than ApEn and SampEn. In Subsection 3.3.3 we show experimentally that ePE seems to be less sensitive than ApEn and SampEn to the length of time series. Subsection 3.3.4 is intended to show that ApEn and SampEn are more appropriate than ePE to apply when complexity of a time series is high. Subsection 3.3.5 is devoted to a discussion of robustness with respect to noise of ePE, ApEn and SampEn. In Subsection 3.3.6 we introduce a modification of ePE which is more robust than ePE with respect to noise in some cases. Finally, we show that ePE is computed from a time series considerably faster than ApEn and SampEn in Subsection 3.3.7. In all experiments in this section we use MATLAB scripts from [Lee14a, Lee14b, Una14] when computing the values of ApEn, SampEn and ePE, correspondingly.

3.3.1 Preliminaries

The following chaotic maps are chosen (for illustration when comparing the entropies) due to their complex behavior in dependence on the parameter and due to the known

values of the KS entropy or the Lyapunov exponent (LE), see Subsection 2.1.2 for the relation between the KS entropy and the LE.

We consider the logistic map $T_{\text{LM}} : [0, 1] \leftrightarrow$ given by

$$T_{\text{LM}}(x) = Ax(1 - x) \tag{3.21}$$

with $A \in [3.5, 4]$. The Lyapunov exponent of T_{LM} is estimated numerically by [Spr03]:

$$\lambda_{\text{LM}} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \ln |A(1 - 2T_{\text{LM}}^i(x))|.$$

The tent map $T_{\text{TM}} : [0, 1] \leftrightarrow$ and the skewed (asymmetric) tent map $T_{\text{STM}} : [0, 1] \leftrightarrow$ are given by

$$T_{\text{TM}}(x) = A \min\{x, 1 - x\},$$

$$T_{\text{STM}}(x) = \begin{cases} \frac{x}{A} & \text{for } x \in [0, A), \\ \frac{(1-x)}{(1-A)} & \text{for } x \in [A, 1] \end{cases}$$

for $A \in (1, 2]$ and $A \in (0, 1)$, respectively. The values of KS entropy for T_{TM} and T_{STM} are given by

$$h_{\mu}(T_{\text{TM}}) = \ln A,$$

$$h_{\mu}(T_{\text{STM}}) = -A \ln A - (1 - A) \ln(1 - A),$$

correspondingly [YMS83, LPS93].

3.3.2 Robustness with respect to strictly monotone transformations

In this subsection we illustrate that ePE is more robust with respect to strictly monotone transformations than ApEn and SampEn.

Ordinal patterns are invariant with respect to strictly monotone increasing transformations, because such transformations do not change order relations between the values of a time series [Pom98, BP02]. Strictly monotone decreasing transformations “invert” ordinal patterns that correspond to the vectors without equal entries. Therefore, for a time series with a low frequency of occurrence of equal values, one can say that ePE is almost invariant with respect to strictly monotone transformations. (Note that modified ePE defined in Section 4.4 for ordinal patterns with tied ranks is invariant to strictly monotone transformations.) ApEn and SampEn are only relatively robust with respect to strictly monotone transformations, because they strongly depend on metric information due to the tolerance r (see Definitions 14, 15), see the following example for an illustration.

Example 13. In Figure 3.3 we present the values of ApEn, SampEn and ePE computed from the orbits of T_{STM} and from the same orbits distorted by $f(x) = \tanh(10x - 5)$. (The length of the orbit is 10^4 , the step between the values of $A \in [0, 1]$ is $7 \cdot 10^{-3}$.) The values of the KS entropy of T_{STM} are presented for comparison.

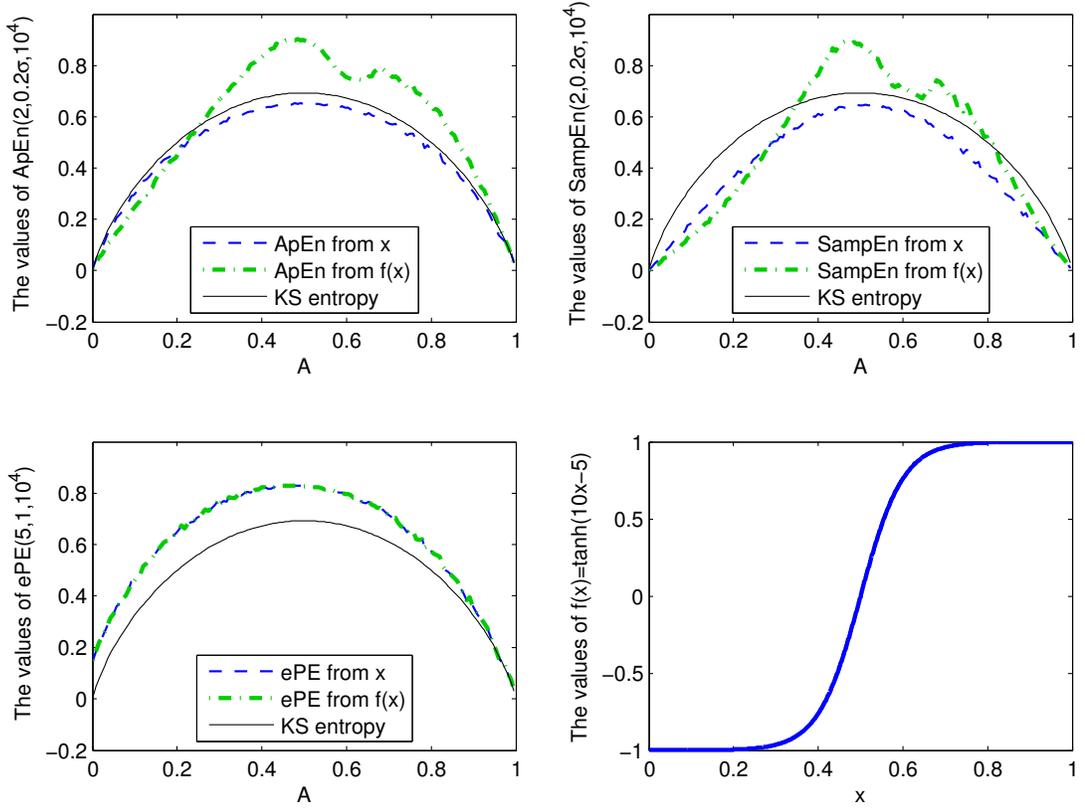


Figure 3.3: The values of approximate entropy (ApEn), sample entropy (SampEn) and empirical permutation entropy (ePE) computed from the orbits of the skewed tent map and from the same orbits distorted by the transformation $f(x) = \tanh(10x - 5)$

One can see that the values of ePE are the same when computed from the distorted and from the original orbits of T_{STM} , whereas the values of ApEn and SampEn change significantly. The distortions of ApEn and SampEn are caused by the alterations of distances between the values of a time series, whereas counting distances between the values of a time series is an important step in computing the values of ApEn and SampEn (see Definitions 14 and 15).

Robustness with respect to strictly monotone transformations (for a time series with low frequency of occurrence of equal values) is an important feature of ePE, because such transformations can be caused by changing the equipment when acquiring time series [BP02].

Note that in [HPC⁺05] it was proposed to use *ranking* preprocessing before computing ApEn and SampEn to make them invariant with respect to strictly monotone

transformations. However, it requires additional computational costs and study of the consequences of such preprocessing. Meanwhile, ePE has ranking as an intrinsic step of its algorithm, which is, of course, advantageous computationally.

3.3.3 Dependence on the length of a time series

In this subsection, we illustrate on the basis of the performed experiments, that ePE seems to be less sensitive to the length of a time series than ApEn and SampEn. Despite of the fact that ApEn and SampEn are often mentioned as appropriate for measuring complexity of short ($N > 100$) time series [Pin91, Pin95, RM00], we show in Example 14 that the values of ePE are less dependent than the values of ApEn and SampEn on the length of a time series for $N = 100, 200, 500$. However we do not have any strict theoretical explanation for this.

We introduce the following coefficients to assess how much the values of the entropies are changed with the length increasing from N to $L > N$:

$$\begin{aligned}\alpha_{\text{AE}}(k, r, N, L) &= \left| 1 - \frac{\text{ApEn}(k, r, N)}{\text{ApEn}(k, r, L)} \right| \\ \alpha_{\text{SE}}(k, r, N, L) &= \left| 1 - \frac{\text{SampEn}(k, r, N)}{\text{SampEn}(k, r, L)} \right| \\ \alpha_{\text{PE}}(d, \tau, N, L) &= \left| 1 - \frac{\text{ePE}(k, r, N)}{\text{ePE}(k, r, L)} \right|.\end{aligned}$$

The larger the coefficient $\alpha_{\text{AE}}(k, r, N, L)$ for $N < L$ is, the more sensitive to the length N of a time series the values of $\text{ApEn}(k, r, N)$ are (the same holds for SampEn with the coefficient α_{SE} and for ePE with the coefficient α_{PE}).

Example 14. In Figure 3.4 we present the values of the coefficients $\alpha_{\text{AE}}(k, r, N, L)$, $\alpha_{\text{SE}}(k, r, N, L)$, $\alpha_{\text{PE}}(d, \tau, N, L)$ for the entropies computed from the orbits of T_{TM} for the lengths $N = 100, 200, 500$ and $L = 5000$. One can see that the values of $\alpha_{\text{PE}}(d, \tau, N, L)$ are considerably smaller than the values of $\alpha_{\text{AE}}(k, r, N, L)$ and $\alpha_{\text{SE}}(k, r, N, L)$ for $N = 100, 200, 500$. This illustrates that ePE is less sensible than ApEn and SampEn to the length of a time series.

In Figure 3.5 we present the values of the coefficients $\alpha_{\text{AE}}(k, r, N, L)$, $\alpha_{\text{SE}}(k, r, N, L)$, $\alpha_{\text{PE}}(d, \tau, N, L)$ for the entropies computed from the orbits of T_{STM} for the lengths $N = 100, 200, 500$ and $L = 5000$. One can see that the values of $\alpha_{\text{PE}}(d, \tau, N, L)$ are a bit smaller than the values of $\alpha_{\text{AE}}(k, r, N, L)$ and $\alpha_{\text{SE}}(k, r, N, L)$.

In Figure 3.6 we present the values of the coefficients $\alpha_{\text{AE}}(k, r, N, L)$, $\alpha_{\text{SE}}(k, r, N, L)$, $\alpha_{\text{PE}}(d, \tau, N, L)$ for the entropies computed from the orbits of T_{LM} for the lengths $N = 100, 200, 500$ and $L = 5000$. One can see that the values of $\alpha_{\text{PE}}(d, \tau, N, L)$ are considerably smaller than the values of $\alpha_{\text{AE}}(k, r, N, L)$ and $\alpha_{\text{SE}}(k, r, N, L)$. This again illustrates that ePE is less sensible than ApEn and SampEn to the length of a time series. Note that the discontinuities in Figure 3.6 in the plot for $\text{SampEn}(k, r, N)$ are caused by

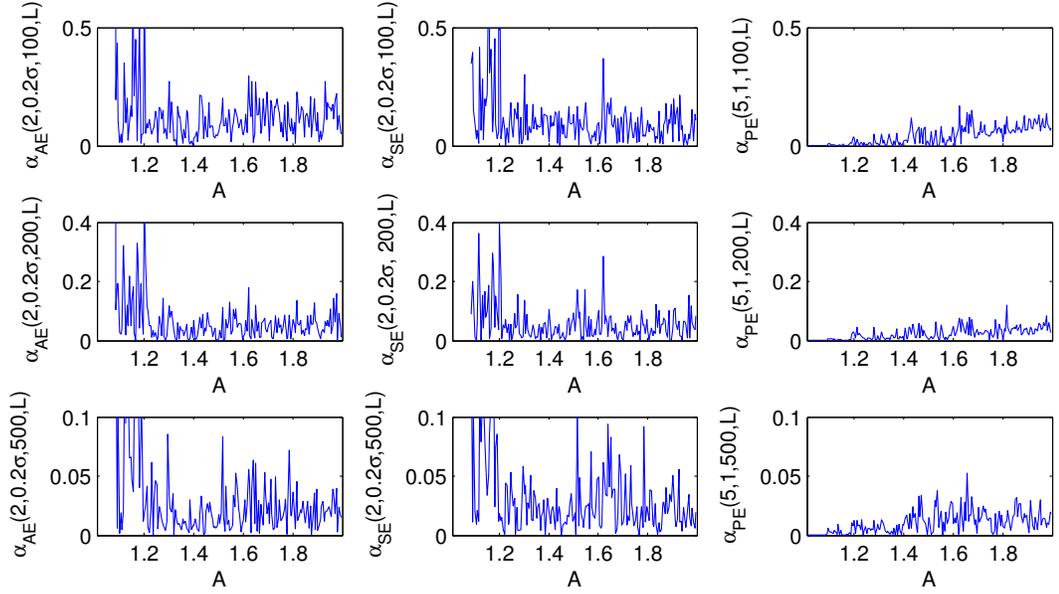


Figure 3.4: Sensitivities $\alpha_{AE}(k, r, N, L)$ of the approximate entropy, $\alpha_{SE}(k, r, N, L)$ of the sample entropy and $\alpha_{PE}(k, r, N, L)$ of the empirical permutation entropy, computed from the orbits of the tent map for $N = 100, 200, 500$ and $L = 5000$

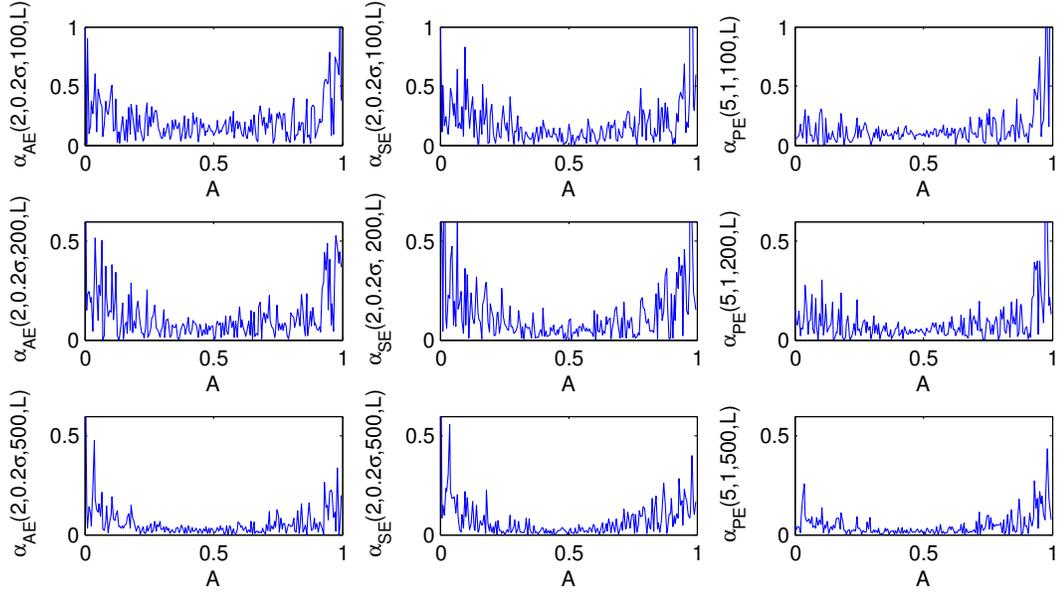


Figure 3.5: Sensitivities $\alpha_{AE}(k, r, N, L)$ of the approximate entropy, $\alpha_{SE}(k, r, N, L)$ of the sample entropy and $\alpha_{PE}(k, r, N, L)$ of the empirical permutation entropy, computed from the orbits of the skewed tent map for $N = 100, 200, 500$ and $L = 5000$

the undefined values of $\text{SampEn}(k, r, N)$ for either $\widehat{C}(k, r, N) = 0$ or $\widehat{C}(k + 1, r, N) = 0$.

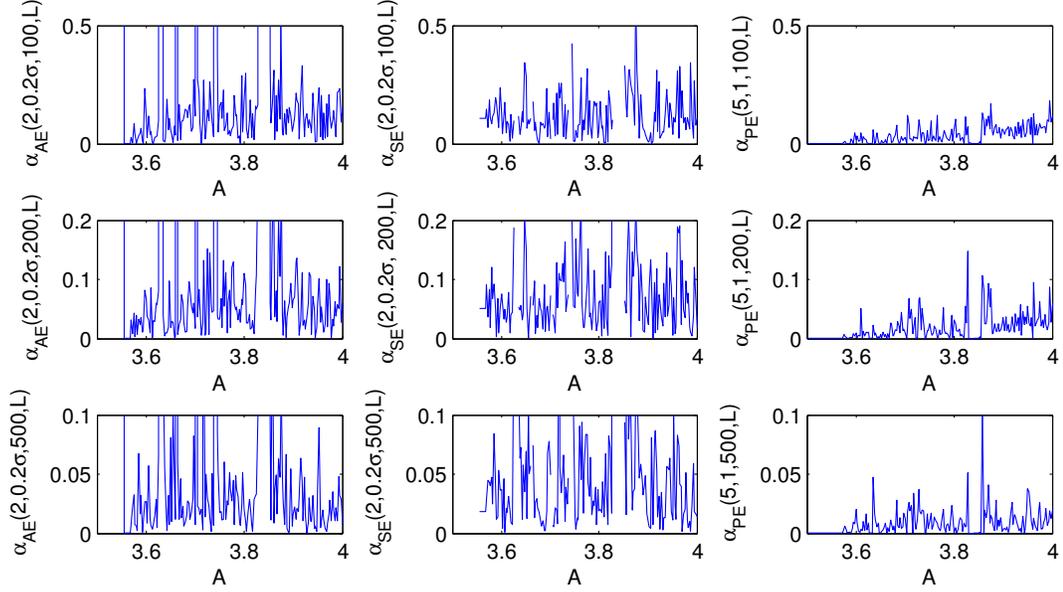


Figure 3.6: Sensitivities $\alpha_{AE}(k, r, N, L)$ of the approximate entropy, $\alpha_{SE}(k, r, N, L)$ of the sample entropy and $\alpha_{PE}(k, r, N, L)$ of the empirical permutation entropy, computed from the orbits of the logistic map for $N = 100, 200, 500$ and $L = 5000$

3.3.4 Estimation of large complexity

In this subsection we illustrate that ApEn and SampEn allow to estimate correctly larger range of complexities than ePE, because ePE has a low upper bound. Indeed, it holds for all $k, N \in \mathbb{N}$ and $r \in \mathbb{R}$ [RM00]:

$$0 \leq \text{ApEn}(k, r, N) \leq \ln(N - k),$$

$$0 \leq \text{SampEn}(k, r, N) \leq \ln(N - k) + \ln\left(\frac{N - k - 1}{2}\right),$$

whereas it holds for all $d, \tau, N \in \mathbb{N}$

$$\text{ePE}(d, \tau, N) \leq \frac{\ln(d + 1)!}{d}. \quad (3.22)$$

Example 15. Let us consider the β -transformation $T_\beta(x) = \beta x \pmod{1}$ for $\beta = 11$

$$T_{11}(x) = 11x \pmod{1}$$

with the KS entropy $\ln(11)$ [Cho00]. In Figure 3.7 we present the values of ApEn, SampEn and ePE computed from the orbits of T_{11} for different lengths N and for different orders d . One can see that the values of ApEn and SampEn approach $\ln(11)$ starting from the length $N \approx 10^5$ of a time series, whereas the values of $\text{ePE}(d, 1, N)$ are bounded by (3.22) that is not enough for $d \leq 9$. Note that $d > 9$ is usually not used in applications since it requires rather large length of a time series $N > 5 \cdot 10! = 18144000$ for reliable estimation of complexity [AZS08].

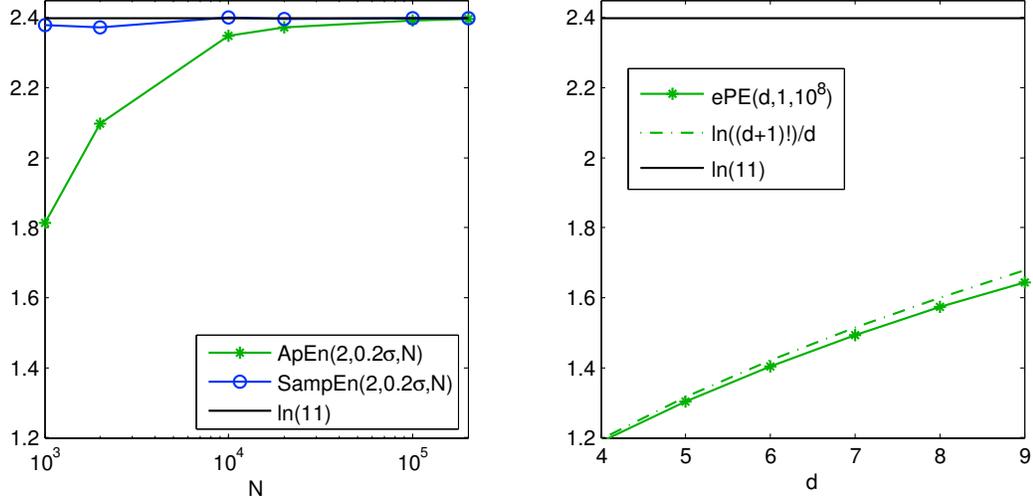


Figure 3.7: The values of approximate entropy, sample entropy and empirical permutation entropy computed from the orbit of the beta-transformation in dependence on the length of a time series N and the order d , correspondingly

The consequence is that ePE fails to distinguish between the complexities larger than $\frac{\ln((d+1)!)}{d}$. For example, we present in Figure 3.8 the values of ApEn, SampEn and ePE computed from the orbits of the beta-transformation

$$T_\beta(x) = \beta x \pmod{1}$$

for the values $\beta = 5, 7, \dots, 15$. Note that the KS entropy of the beta transformation is given by $\ln \beta$ [Cho00]. One can see that the values of ePE(9, 1, $4 \cdot 10^7$) are almost the same for the values $\beta = 5, 7, \dots, 15$ since they are bounded by $\frac{\ln(10!)}{9}$, whereas the values of ApEn and SampEn estimate the complexity correctly.

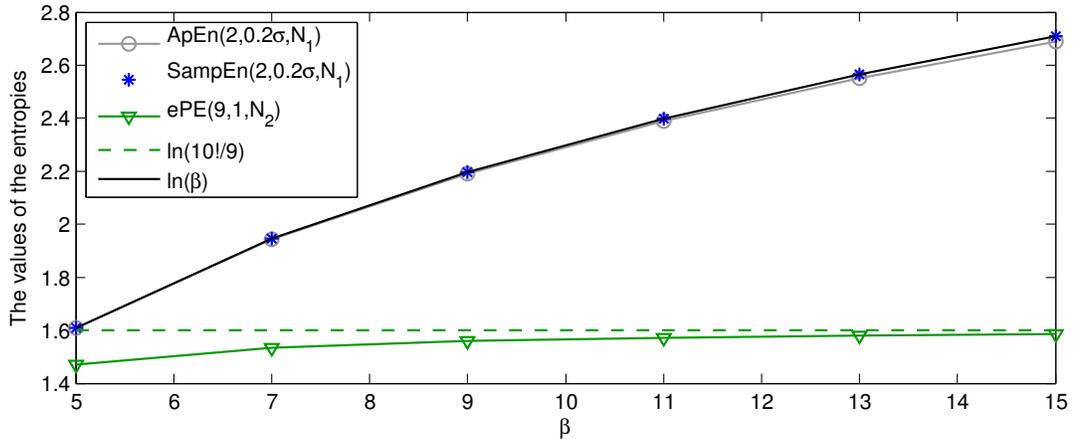


Figure 3.8: The values of approximate entropy, sample entropy and empirical permutation entropy computed from the orbit of the beta-transformation in dependence on the parameter β , $N_1 = 5 \cdot 10^4$, $N_2 = 4 \cdot 10^7$

3.3.5 Robustness to noise

In this subsection we illustrate that ePE is not as robust to noise as it is often reported (e.g. [ZZRP12, MLLF⁺12, LYLO14]) and we propose a way of how the robustness of ePE can be assessed. For that we introduce the following quantity MD that assesses a number of pairs of points that are abnormally close or abnormally distant in the vectors corresponding to the ordinal patterns.

Definition 18. Given $d, \tau, N \in \mathbb{N}$, and $\eta_1, \eta_2 \in \mathbb{R}$ with $0 \leq \eta_1 < \eta_2$, for a time series $(x_t)_{t=1}^N$ a quantity $\text{MD}(d, \tau, (x_t)_{t=1}^N, \eta_1, \eta_2)$ is defined by

$$\text{MD}(d, \tau, (x_t)_{t=1}^N, \eta_1, \eta_2) = \frac{2 \sum_{t=d\tau+1}^N \#\{(i, j) : 0 \leq i < j \leq d, |x_{t-i\tau} - x_{t-j\tau}| < \eta_1 \text{ or } |x_{t-i\tau} - x_{t-j\tau}| > \eta_2\}}{d(d+1)(N-d\tau)}.$$

Further we use the short form $\text{MD}(d, \tau, N, \eta_1, \eta_2)$ instead of $\text{MD}(d, \tau, (x_t)_{t=1}^N, \eta_1, \eta_2)$ when no confusion arises.

The lower threshold η_1 allows to detect the pairs of points that are abnormally close ($< \eta_1$) to each other, i.e. the order relation between these points could be easily changed by a small noise. The upper threshold η_2 allows to detect the pairs of points that are abnormally distant ($> \eta_2$) from each other. This means that MD counts “unreliable” pairs of points that can introduce a mistake when computing empirical permutation entropy, see the following examples for an illustration.

Note that $\frac{d(d+1)(N-d\tau)}{2}$ is the number of the ordered pairs of points within the ordinal patterns of order d and delay τ (we count each pair of points k times if it belongs to k ordinal patterns), i.e. it holds for all $d, \tau, N \in \mathbb{N}$ and $\eta_1, \eta_2 \in \mathbb{R}$ with $0 \leq \eta_1 < \eta_2$

$$0 \leq \text{MD}(d, \tau, N, \eta_1, \eta_2) \leq 1.$$

Example 16. In this example when computing the values of MD we deliberately set $\eta_2 = 2$, i.e. we do not use upper threshold η_2 .

In Figure 3.9 we present the values of SampEn, ePE and MD computed from the orbits of T_{STM} and from the same orbits contaminated by the noises with $\mathcal{N}(0, 0.05^2)$ and $\mathcal{N}(0, 0.1^2)$. (We present here only the values of SampEn, because the values of ApEn are very similar to them.) The length of the orbit is $5 \cdot 10^4$, the step between the values of A is $5 \cdot 10^{-4}$.

When comparing with the values of the KS entropy, one can see that the ePE values are distorted when computed from the noisy orbits (especially for the parameters $A \in [0, 0.1] \cup [0.9, 1]$), whereas the SampEn values are almost not distorted. The distortion of the ePE values can be explained by the large MD values for the parameters $A \in [0, 0.1] \cup [0.9, 1]$, which means that for these values of A there are many pairs of

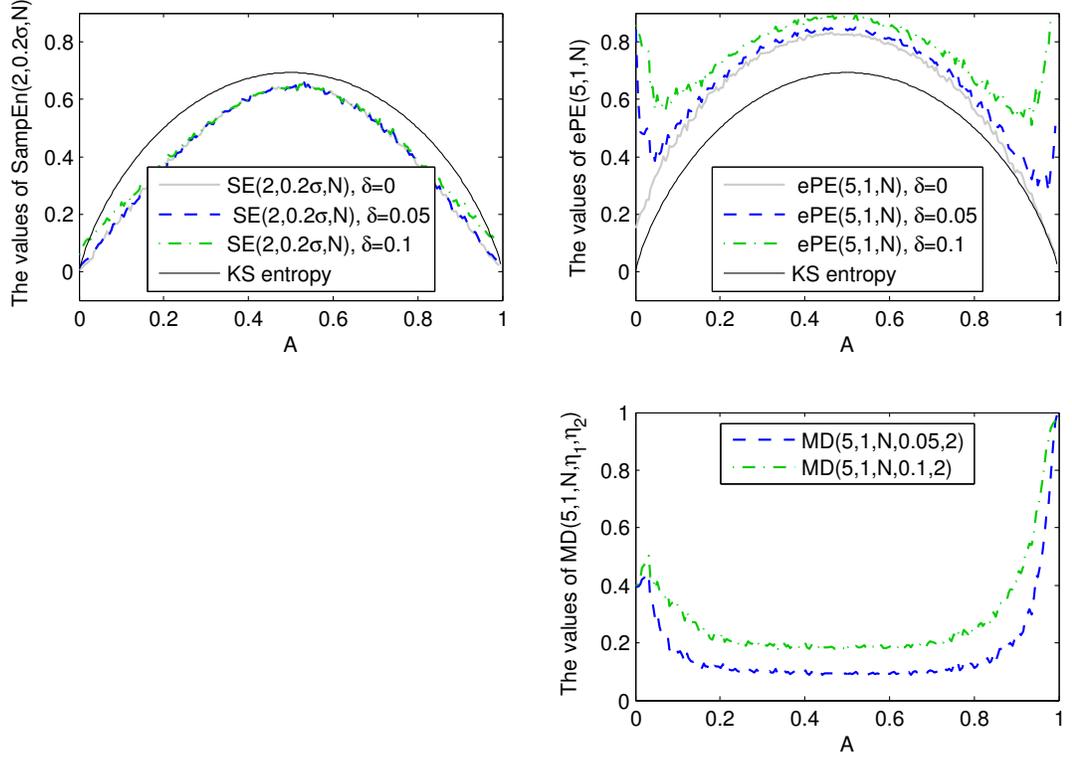


Figure 3.9: The values of the sample entropy, the empirical permutation entropy, and MD, computed from the orbits of the skewed tent map; δ stands for the standard deviation of the added noise; SE stands for SampEn; $N = 5000$

points with the absolute value of their difference within η_1 , i.e. they are easily changed by the noises with $\mathcal{N}(0, 0.05^2)$ and $\mathcal{N}(0, 0.1^2)$.

In Figure 3.10 we present the SampEn, ePE and MD values computed from the orbits of T_{LM} and from the same orbits contaminated with the noises with $\mathcal{N}(0, 0.1^2)$ and $\mathcal{N}(0, 0.15^2)$. The length of the orbit is $5 \cdot 10^4$, the step between the values of A is 10^{-4} . One can see that the values of SampEn are strongly distorted when computed from the noisy orbits for a noise with $\mathcal{N}(0, 0.15^2)$. The ePE values are also distorted, but these distortions are explained by the MD values. The larger the MD values are, the larger the distortions of the ePE values are.

The distortions of ePE caused by the noise can be reduced by varying the order d and the delay τ . For example, in Figure 3.11 we present the ePE and MD values computed for different delays τ from the orbits of T_{STM} and from the same orbits contaminated with noise with $\mathcal{N}(0, 0.2^2)$. The length of the orbit is 10^4 , the step between the values of A is $5 \cdot 10^{-4}$. When comparing with the KS entropy values, one can see that the values of ePE(5,3,N) are much more reliable than the values ePE(5,1,N), which is explained by the values $MD(5,3,N,0.2,2) < MD(5,1,N,0.2,2)$.

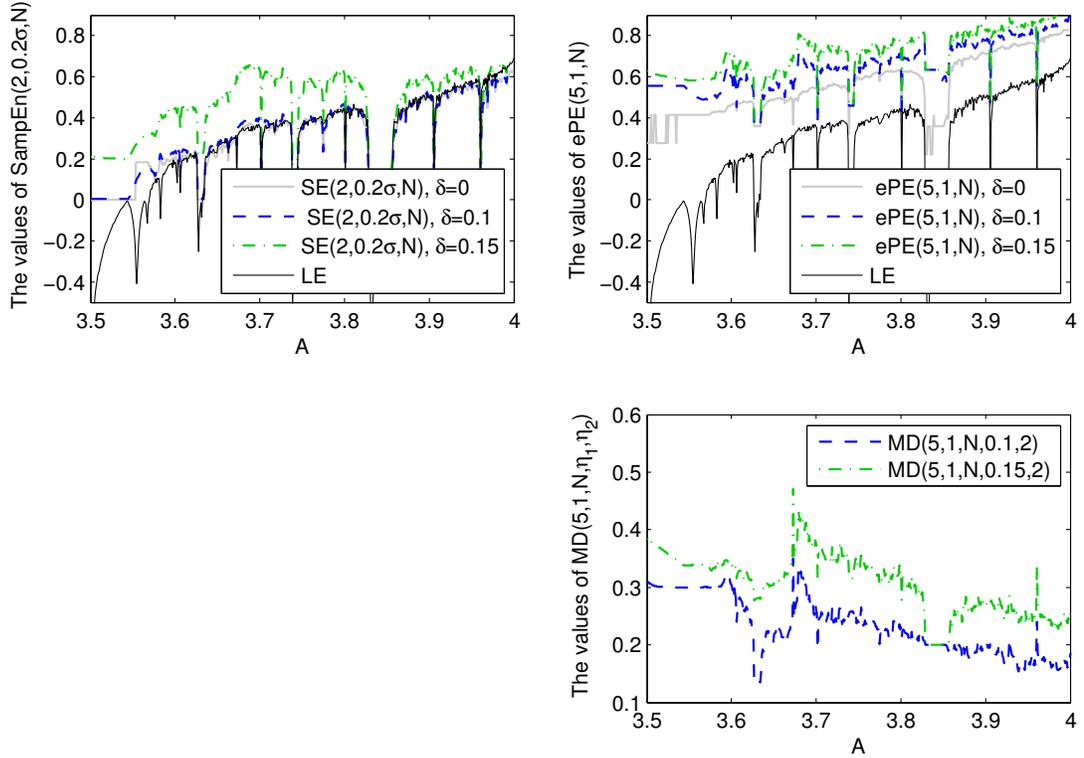


Figure 3.10: The values of the sample entropy, the empirical permutation entropy, and MD, computed from the orbits of the logistic map; δ stands for the standard deviation of the added noise; SE stands for SampEn; $N = 5000$

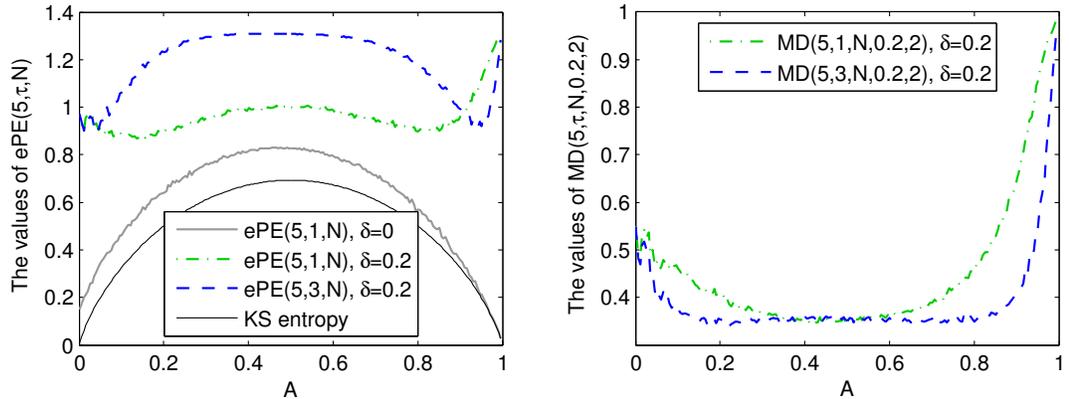


Figure 3.11: The values of empirical permutation entropy and MD computed from the orbits of the skewed tent map for different τ ; δ stands for the standard deviation of the added noise; $N = 10^4$

Remark 8. In Example 16 we used only the lower threshold η_1 by setting $\eta_2 = 2$ when computing MD, however in Section 5.2 we illustrate using MD for EEG data also for the upper threshold η_2 (see Example 25, p. 90).

We conclude that the introduced quantity MD helps in understanding when the ePE is robust to noise or to abnormal deviations of a time series. On the basis on the MD quantity we introduce a robust ePE in Subsection 3.3.6.

Remark 9. Note that the quantity MD is in some sense related to the estimate of the correlation integral. Compare the following representations for $\eta_2 = \infty$ and $\tau = 1$:

$$\widehat{C}(k, \eta_1, N) = \frac{2\#\left\{(i, j) : 1 \leq i < j \leq N - k + 1, \max_{l=0,1,\dots,k-1} |x_{i+l} - x_{j+l}| \leq \eta_1\right\}}{(N - k - 1)(N - k)},$$

$$\text{MD}(d, 1, N, \eta_1, \infty) = \frac{2 \sum_{t=d+1}^N \#\{(i, j) : 0 \leq i < j \leq d, |x_{t-i} - x_{t-j}| < \eta_1\}}{d(d+1)(N-d)}.$$

When computing $\text{MD}(d, 1, N, \eta_1, \infty)$ we count the pairs of the values $\{x_{t-i}, x_{t-j}\}$ with $|x_{t-i} - x_{t-j}| < \eta_1$, whereas when computing $\widehat{C}(k, \eta_1, N)$ we count the pairs of the vectors $\{(x_i, x_{i-1}, \dots, x_{i-k}), (x_j, x_{j-1}, \dots, x_{j-k})\}$ such that all pairwise points are within a tolerance η_1 : $|x_{i+l} - x_{j+l}| \leq \eta_1$ for all $l = 1, 2, \dots, k$.

3.3.6 Robust empirical permutation entropy

A natural idea is to enhance the robustness of ePE with respect to noise and abnormal changes by counting only ordinal patterns with sufficiently many “reliable” pairs of points.

Definition 19. Given $d, \tau \in \mathbb{N}$, for $\eta_1, \eta_2 \in \mathbb{R}$ with $0 \leq \eta_1 < \eta_2$, let us call an ordinal pattern of the vector $(x_t, x_{t-\tau}, \dots, x_{t-d\tau}) \in \mathbb{R}^{d+1}$ (η_1, η_2) -ordinal pattern if

$$\#\{(i, j) : 0 \leq i < j \leq d, |x_{t-i\tau} - x_{t-j\tau}| < \eta_1 \text{ or } |x_{t-i\tau} - x_{t-j\tau}| > \eta_2\} < \frac{(d+1)d}{8}. \quad (3.23)$$

The threshold $\frac{(d+1)d}{8}$ is chosen as a quarter of the amount of the ordered pairs of entries from the vector $(x_t, x_{t-\tau}, \dots, x_{t-d\tau})$.

Definition 20. For $\eta_1, \eta_2 \in \mathbb{R}$ with $0 \leq \eta_1 < \eta_2$, the *robust empirical permutation entropy* (rePE) of order $d \in \mathbb{N}$ and of delay $\tau \in \mathbb{N}$ of a time series $(x_t)_{t=1}^N$ is given by

$$\text{rePE}(d, \tau, (x_t)_{t=1}^N, \eta_1, \eta_2) = -\frac{1}{d} \sum_{j=0}^{(d+1)!-1} p_j \ln p_j, \text{ where}$$

$$p_j = \frac{\#\{i = d\tau + 1, d\tau + 2, \dots, N \mid (x_i, x_{i-\tau}, \dots, x_{i-d\tau}) \text{ has the } (\eta_1, \eta_2)\text{-ordinal pat. } j\}}{\#\{i = d\tau + 1, d\tau + 2, \dots, N \mid (x_i, x_{i-\tau}, \dots, x_{i-d\tau}) \text{ has a } (\eta_1, \eta_2)\text{-ordinal pat.}\}}$$

(with $0 \ln 0 := 0$ and $0/0 := 0$).

Further we use the short form $\text{rePE}(d, \tau, N, \eta_1, \eta_2)$ instead of $\text{rePE}(d, \tau, (x_t)_{t=1}^N, \eta_1, \eta_2)$ when no confusion arises.

Note that one cannot directly introduce robust ApEn or SampEn in a similar way because they are based on counting pairs within a tolerance r (see Definitions 14, 15).

Remark 10. A similar to the rePE quantity has been introduced in [OSD08], where the authors employ an additional ordinal pattern that corresponds to the vectors with any two points with the difference within a threshold. An empirical permutation entropy for this case is computed from the distribution of $(d + 1)! + 1$ ordinal patterns.

Example 17. In Figures 3.12-3.14 are presented the values of ePE and rePE, computed from the orbits of the maps T_{TM} , T_{STM} and T_{LM} and from the same orbits with the added noise with $\mathcal{N}(0, 0.1^2)$. The length of the orbit is 10^5 , the step between the values A is $5 \cdot 10^{-4}$ for T_{TM} , T_{STM} and 10^{-4} for T_{LM} . We deliberately set $\eta_2 = 2$, i.e. we do not use the upper threshold in this example. When comparing with the (KS entropy) LE values, one can see that for these maps rePE provides more reliable estimation of complexity than ePE.

In Figure 3.12 one can see that many values of rePE for $A < 1.4$ are equal to 0 which is related to the small number of $(0.1, 2)$ -ordinal patterns, whereas the values of ePE of noisy time series provide the wrong impression that the complexity of a time series for the parameter $A < 1.4$ is large (compare with the KS entropy values).

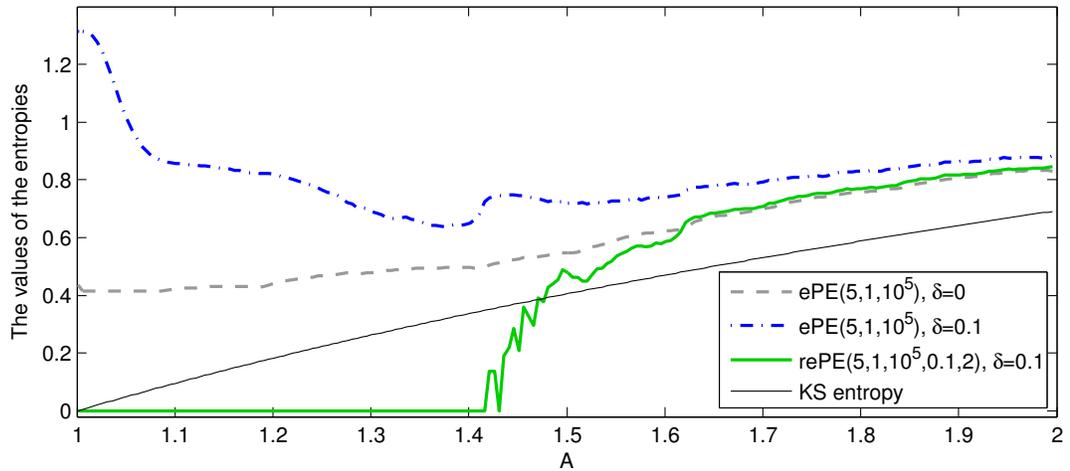


Figure 3.12: The values of robust empirical permutation entropy and empirical permutation entropy computed from the orbits of the tent map and from the same orbits contaminated with noise; δ stands for the standard deviation of noise

In Figure 3.14 one can see that the rePE values computed from the noisy orbits almost coincide with the ePE values computed from the “clean” orbits of T_{LM} , whereas the ePE values computed from the noisy orbits of T_{LM} are significantly distorted.

In Example 17 we have illustrated rePE only for the lower threshold η_1 , however in Section 5.2 we demonstrate an application of rePE for EEG data to detect abnormal changes also for the upper threshold η_2 .

Let us summarize that the introduced robust empirical permutation entropy (rePE) provides better robustness with respect to noise than the ePE and can be useful in many

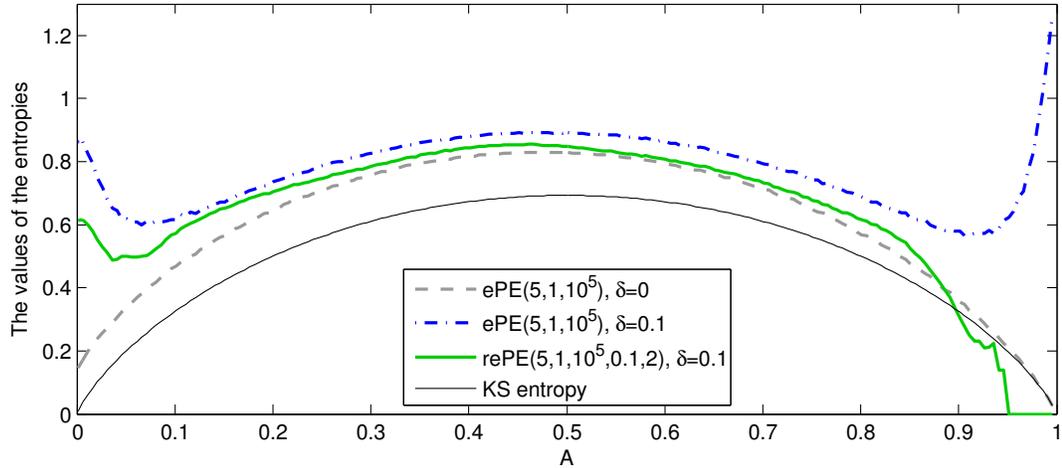


Figure 3.13: The values of robust empirical permutation entropy and empirical permutation entropy computed from the orbits of the skewed tent map and from the same orbits contaminated with noise; δ stands for the standard deviation of noise

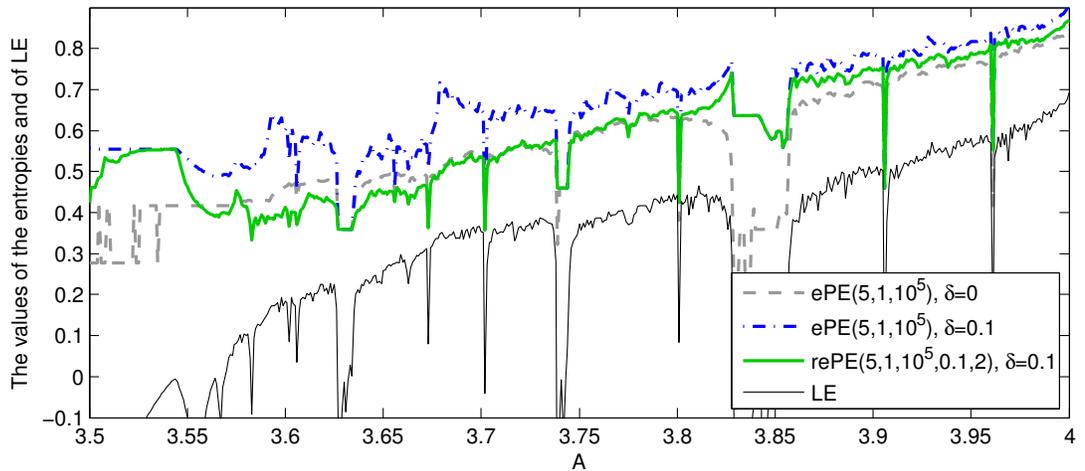


Figure 3.14: The values of robust empirical permutation entropy and empirical permutation entropy computed from the orbits of the logistic map and from the same orbits contaminated with noise; δ stands for the standard deviation of noise

cases. However, the rePE has two drawbacks. Firstly, choice of the thresholds η_1 and η_2 is ambiguous and needs further investigation. Secondly, rePE has a slower computational algorithm than ePE (see Section 4.6 for details). One can find a MATLAB script for computing the rePE in Appendix A.4.

3.3.7 Computational efficiency

In this subsection we show that the algorithm for computing ePE from [UK13] is much more efficient than the fast algorithms for computing ApEn and SampEn introduced in

[PWLL11]³. We present the efficiency of the methods in terms of computational time and storage use in dependence on the length N of a considered time series in Table 3.1.

Quantity	Computing method	Computational time	Storage use
ApEn	[PWLL11]	$O\left(N^{\frac{3}{2}}\right)$	$O(N)$
SampEn	[PWLL11]	$O\left(N^{\frac{3}{2}}\right)$	$O(N)$
ePE	[UK13]	$O(N)$	$O((d+1)!)$

Table 3.1: Computational and storage requirements when computing the approximate entropy, sample entropy and empirical permutation entropy

According to Table 3.1 the storage use is less when computing ePE than the storage use when computing ApEn and SampEn. The fast algorithm for computing ePE in sliding windows for the orders $d = 1, 2, \dots, 9$ is described in Chapter 4 and in [UK13].

For better illustration we present in Figure 3.15 the computational times of the entropies computed from the orbits of T_{LM} for $A = 4$ in dependence on the length N of a time series. The time is averaged over several trials.

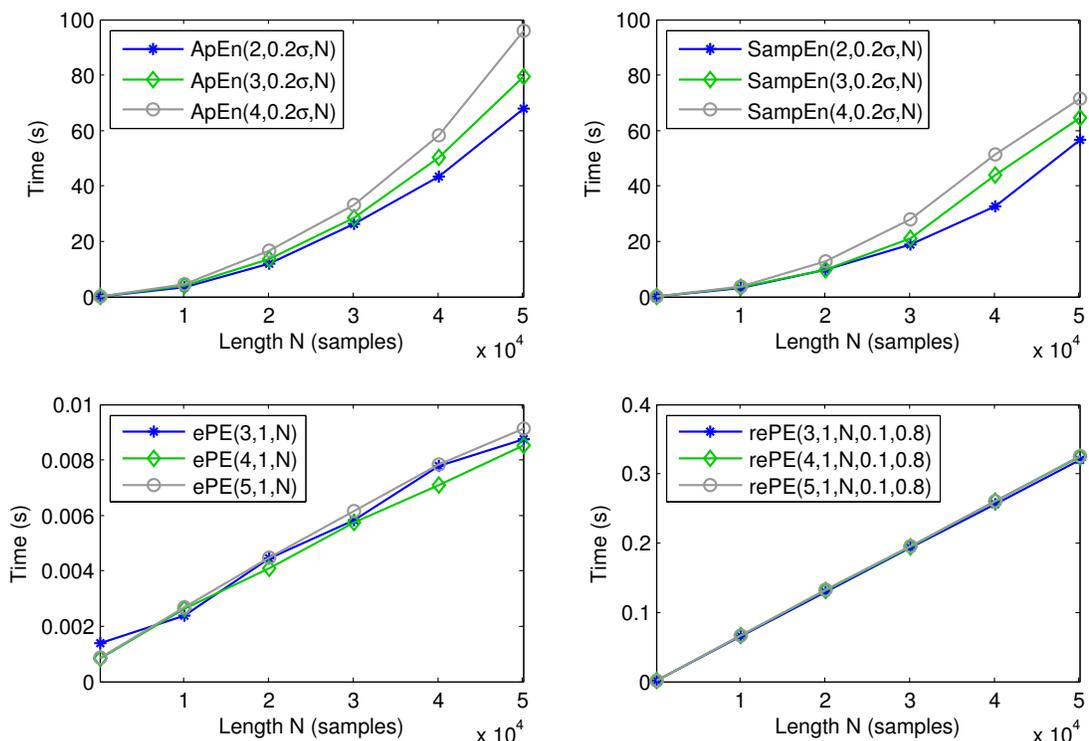


Figure 3.15: The times (measured by the MATLAB function “cputime”) of computing approximate entropy, sample entropy, empirical permutation entropy and robust empirical permutation entropy by the MATLAB scripts [Lee14a, Lee14b], [Una14] (“PE.m”) and “rePE” (Appendix A.5)

³the most fast to our knowledge algorithms for computing ApEn and SampEn

One can see that the ePE values are computed about 10^4 times faster than the ApEn and SampEn values from the same lengths of a time series; and the rePE values are computed about 250 times faster than the ApEn and SampEn values from the same lengths of a time series.

3.4 Summary

In this chapter, we have considered theoretical underpinnings for the approximate entropy (ApEn), the sample entropy (SampEn) and the empirical permutation entropy (ePE), which are important to understand where the entropies are stemming from. We have also compared the practical properties of ePE, ApEn, and SampEn. Let us now list the advantages of the ePE with respect to the ApEn and the SampEn.

- The computational algorithm of ePE is considerably faster and storage requirements are less than for computing ApEn and SampEn; this allows to compute ePE of large datasets in real time (see Subsection 3.3.7 for details).
- ePE is almost invariant with respect to strictly monotone transformations, whereas ApEn and SampEn are not (Subsection 3.3.2).
- Experimentally, ePE provides more stable results than ApEn and SampEn for short time series with the lengths $100 < N < 500$ (Subsection 3.3.3).
- One can assess robustness with respect to noise of ePE for a given time series by the introduced quantity MD and one can improve the robustness with respect to noise of ePE with the introduced quantity rePE (Subsections 3.3.5, 3.3.6).

The advantages of ApEn and SampEn in comparison with ePE are the following.

- ApEn and SampEn allow to estimate correctly a larger range of complexities than ePE (see Subsection 3.3.4).

3.4.1 The choice between approximate entropy, sample entropy and empirical permutation entropy

On the basis of the obtained results we formulate some hints for application of ePE, ApEn and SampEn.

ePE is more appropriate to apply

- for measuring the complexity of a time series in a very fast way;
- for measuring the complexity of short time series;
- for measuring the complexity of a time series that are possibly modified by strictly monotone transformations.

ApEn and SampEn are more appropriate to apply

- together with ePE (see Subsection 3.3.4 and example with EEG data in Subsection 5.3.3);
- for assessing complexity of a time series with a large entropy.

3.4.2 Hints for using empirical permutation entropy

In general, for using ePE we recommend

- to set the largest order d satisfying $5(d+1)! < N$ (according to the recommendation from [AZS08]), where N is the length of a time series;
- to take into account that the upper bound of $ePE(d, \tau, N)$ is $\frac{\ln((d+1)!)}{d+1}$. In particular, to take into account that an increase of τ can lead to an increase of the $ePE(d, \tau, N)$ values whereas ePE is bounded by $\frac{\ln((d+1)!)}{d+1}$ (see also Subsection 5.1.2, p. 82);
- to assess the robustness with respect to noise of ePE by the introduced quantity MD (see Example 25, p. 90 for application of MD to EEG data);
- to use rePE for noisy time series or time series contaminated with artifacts (see Section 5.2 for application of rePE to EEG data).

Chapter 4

Efficient computing of ordinal-patterns-based characteristics

Motivated by the good properties and many applications of empirical permutation entropy (see Chapter 3), we propose in this chapter an efficient method of computing it and ordinal patterns (in this chapter we simplify the description and provide more applications of the method from [UK13]). An efficient computing of ordinal patterns provides a fast calculation of not only empirical permutation entropy (ePE), but of many ordinal-patterns-based characteristics, such as the ordinal distributions itself [KS05], the empirical conditional entropy of ordinal patterns [UK14], transcripts [MBAJ09] and other derived measures [KLS07, PR11, Ban14]. The main idea of the efficient method is using the precomputed tables (so-called *lookup tables* in computer science) of successive values instead of computing ordinal patterns and the ePE value at each time point. It is possible to precompute such successive values, because ordinal patterns “overlap” and one can use the information from the previous ordinal pattern in the successive one. Since successive ePE values are computed for successive overlapping time-windows, the possible successive ePE values can be precomputed as well.

For illustration, we present in Figure 4.1 that the ePE values (bottom plot) computed by the efficient method reflect the epileptic seizure in one-channel electroencephalogram (EEG) data (seizure is marked in gray in the upper plot). The processing of the depicted 20 min of EEG data, recorded at a sampling rate of 256 Hz, takes about 1 s in MATLAB R2013b. The EEG data are from The European Epilepsy Database [Epi14]).

In Section 4.1 we recall the definition of ordinal patterns and consider their computing by the relatively fast method from [KSE07], then we recall the definition of ePE and consider the standard procedure of its computing from distributions of ordinal patterns. We introduce a new efficient method of computing ordinal patterns and ePE in Sections 4.2 and 4.3, correspondingly. In Section 4.4 we adapt the fast method to time series with a high frequency of occurrence of equal values (tied ranks). It is reasonable

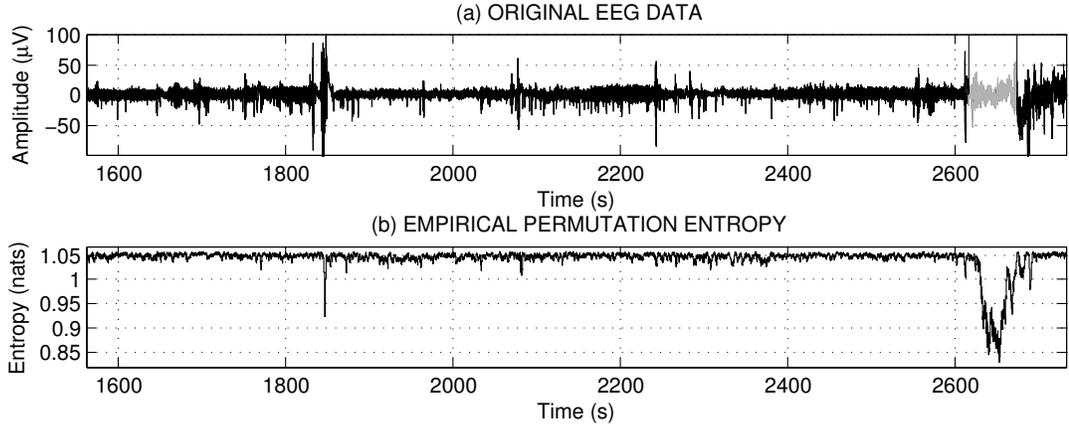


Figure 4.1: Empirical permutation entropy computed from the one-channel EEG data

to take into account these equalities for data digitized with a low resolution, for example, for heart rate variability data as in [BQMS12]. In Section 4.5 we adapt the efficient method for computing the empirical conditional entropy of ordinal patterns [UK14] and for computing of robust ePE (introduced in Subsection 3.3.6), correspondingly. Section 4.6 is devoted to the comparison between two known methods of computing ePE and the introduced fast methods.

Remark 11. Note that the proposed method of efficient computing of ordinal patterns and ePE is reasonable for the moderate orders $d \leq 9$ of ordinal patterns. This restriction is related to a relatively large size $((d + 1)!(d + 1))$ of the precomputed table for the orders $d > 9$. Note that the orders $d = 2, 3, \dots, 6$ are usually recommended for applications [BP02].

4.1 Computing ordinal patterns and the empirical permutation entropy

In Subsection 4.1.1 we consider ordinal patterns and compute them by the method introduced in [KSE07], in Subsection 4.1.2 we consider empirical permutation entropy (ePE) and compute it by the standard procedure.

4.1.1 Ordinal patterns

We recall here the definition of ordinal patterns and their enumeration given in Chapter 3. Note that for encoding ordinal patterns the *inversion numbers* were used in [KSE07]. Inversion numbers can be also referred as *Lehmer code* like in [GLW14].

Definition 21. A delay vector $(x_t, x_{t-\tau}, \dots, x_{t-d\tau})$ has the *ordinal pattern* (OP) $i_d^\tau(t) = (i_1, i_2, \dots, i_d)$ of order $d \in \mathbb{N}$ and delay $\tau \in \mathbb{N}$ when for $l = 1, 2, \dots, d$

$$i_l = \#\{r \in \{0, 1, \dots, l-1\} \mid x_{t-l\tau} \geq x_{t-r\tau}\}. \quad (4.1)$$

Note that we assume here occurrence of equal values (*tied ranks*) in a time series quite rare; by this reason the relation “equal to” is combined with the relation “greater than” in Definition 21. Regarding the time series with a high frequency of occurrence of equal values, we discuss OPs with tied ranks in Section 4.4.

There are $(d+1)!$ OPs of order d , and one assigns to each of them a number from $\{0, 1, \dots, (d+1)! - 1\}$ in a one-to-one way according to [KSE07]

$$n_d^\tau(t) = n_d^\tau(i_d^\tau(t)) = \sum_{l=1}^d i_l \frac{(d+1)!}{(l+1)!}. \quad (4.2)$$

For example, all OPs of order $d = 2$ are given in Table 4.1.

Ordinal pattern						
(i_1, i_2)	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)
$n_2(i_1, i_2) = 3i_1 + i_2$	0	1	2	3	4	5

Table 4.1: The ordinal patterns of order $d = 2$

In Figure 4.2 we present the OPs of the time series $(x_t)_{t=1}^{10}$, the delay $\tau = 2$ indicates a distance between points in OPs, the order $d = 2$ indicates number of points $(d+1) = 3$ in ordinal patterns.

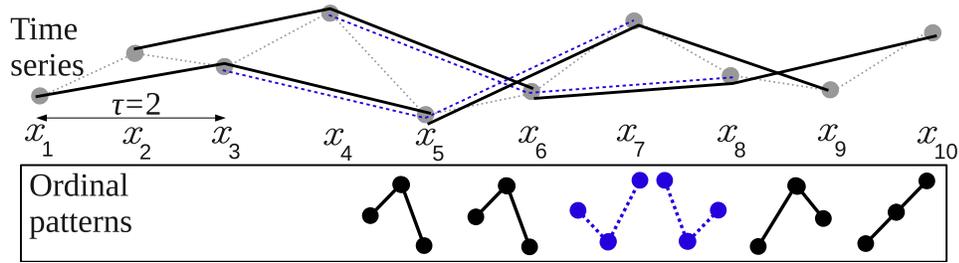


Figure 4.2: Illustration of computing the ordinal patterns of order $d = 2$

One can see that the blue OPs (dashed line) “overlap” the previous black OPs in $d = 2$ points, as well as the black OPs “overlap” the previous blue OPs. This maximal “overlapping” between OPs is usually used in order to obtain the maximal information from a time series [KSE07].

Due to the “overlapping” between OPs one obtains the successive OP $i_d^\tau(t + \tau) = (i'_1, i'_2, \dots, i'_d)$ from the previous one $i_d^\tau(t) = (i_1, i_2, \dots, i_d)$ by

$$i'_{l+1} = \begin{cases} i_l & \text{if } x_{t-l\tau} < x_{t+\tau}, \\ i_l + 1 & \text{otherwise} \end{cases} \quad (4.3)$$

for $l = 0, \dots, d$ with $i_0 = 0$. According to (4.3) one needs only d comparisons and at most d incrementation operations to obtain the successive OP when the current OP is given, which provides a relatively fast computing of OPs [KSE07]. When counting OPs in their number representation (4.2), which is more convenient than in the representation provided by (4.1), one needs d multiplications more.

4.1.2 The empirical permutation entropy

In order to reflect complexity changes in a time series in the course of time, the ePE values are usually computed in sliding time-windows of a fixed size, see Definition 22 (compare with Definition 17, p. 74).

Definition 22. By the empirical permutation entropy (ePE) of order d and of delay τ of a time-window $(x_t, x_{t-1}, \dots, x_{t-M-d\tau+1})$, $t, M \in \mathbb{N}$ one understands the quantity

$$\text{ePE}(d, \tau, M, t) = -\frac{1}{d} \sum_{j=0}^{(d+1)!-1} \frac{q_j(t)}{M} \ln \frac{q_j(t)}{M} = \ln M - \frac{1}{M} \sum_{j=0}^{(d+1)!-1} q_j(t) \ln q_j(t), \quad (4.4)$$

$$\text{where } q_j(t) = \#\{i = t, t-1, \dots, t-M+1 \mid n_d^\tau(i) = j\}$$

(with $0 \ln 0 := 0$).

We use a time-window $(x_t, x_{t-1}, \dots, x_{t-M-d\tau+1})$, because it contains exactly M ordinal patterns which is convenient for computations and further explanations.

In Figure 4.3 we illustrate computing ePE in the two maximally overlapped windows $(x_t)_{t=1}^9$ and $(x_t)_{t=2}^{10}$, containing $M = 5$ OPs of order $d = 2$ for a delay $\tau = 2$.

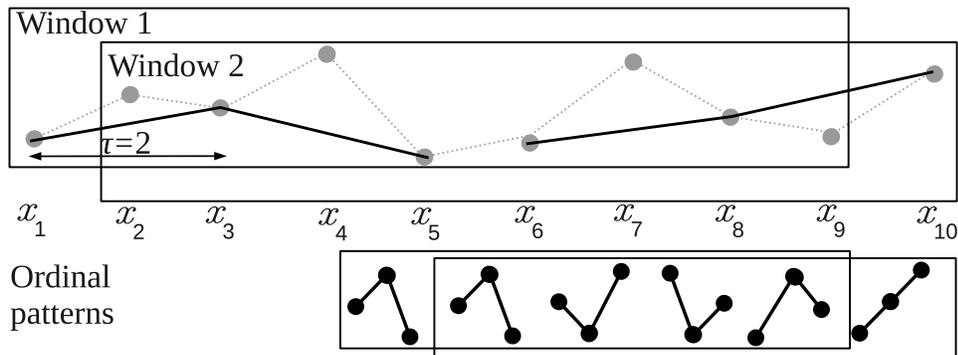


Figure 4.3: Illustration of computing the empirical permutation entropy in two successive and maximally overlapped sliding windows

There are

$$q_0(9) = 0, \quad q_1(9) = 1, \quad q_2(9) = 1, \quad q_3(9) = 1, \quad q_4(9) = 2, \quad q_5(9) = 0 \text{ and}$$

$$q_0(10) = 1, \quad q_1(10) = 1, \quad q_2(10) = 1, \quad q_3(10) = 1, \quad q_4(10) = 1, \quad q_5(10) = 0$$

OPs in Windows 1 and 2, correspondingly (see Table 4.1 for determining types of OPs). Then the ePE at times $t = 9$ and $t = 10$ is computed by (4.4) as

$$\text{ePE}(2, 2, 5, 9) = \ln 5 - \frac{1}{5}(1 \ln 1 + 1 \ln 1 + 1 \ln 1 + 2 \ln 2) = 1.3322$$

$$\text{ePE}(2, 2, 5, 10) = \ln 5 - \frac{1}{5}(1 \ln 1 + 1 \ln 1 + 1 \ln 1 + 1 \ln 1 + 1 \ln 1) = 1.6094.$$

4.2 Efficiently computing the numbers of ordinal patterns

In this section we precompute numbers of possible successive OPs for each ordinal pattern. Using the precomputed values allows to compute numbers of OPs about two times faster than by (4.2). For simplicity, we use here the number representation of OPs provided by (4.2), but, in fact, the type of number representation is not substantial for the method. It means that one can use the fast method of computing the numbers of OPs using different number representations.

4.2.1 Precomputed numbers of ordinal patterns

Given an OP with the number $n_d^\tau(t)$ there are $(d + 1)$ possible successive OPs with the numbers $n_d^\tau(t + \tau)$ (see Table 4.2). For example, for the OP with the number 0 there are three possible positions $l = 0, 1, 2$ of the next point and three possible successive ordinal patterns with the numbers 0, 3, 4, respectively.

	0	1	2	3	4	5
Current ordinal pattern						
Successive ordinal pattern		0 3 4			1 2 5	

Table 4.2: The successive ordinal patterns with the numbers $n_d^\tau(t + \tau)$ given the number $n_d^\tau(t)$ of current ordinal pattern

One can see that it is possible to introduce a function that determines the number of the next OP by the number of the current OP and by the position l of the next point.

Definition 23. For $d, n, l \in \mathbb{N}$ we define a function

$$\phi_d(n, l) = n_1 \tag{4.5}$$

such that for any vector $(x_1, x_2, \dots, x_{d+2}) \in \mathbb{R}^{d+2}$ for that n is the number of OP of $(x_1, x_2, \dots, x_{d+1})$, n_1 is the number of OP of $(x_2, x_3, \dots, x_{d+2})$, it holds

$$l = \#\{r \in \{2, 3, \dots, d+1\} \mid x_r \geq x_{d+2}\}. \tag{4.6}$$

Given the values of $\phi_d(n, l)$ for all n and l , one obtains $n_d^\tau(t + \tau) = \phi_d(n_d^\tau(t), l)$ just by computing

$$l = \#\{r \in \{0, 1, \dots, d-1\} \mid x_{t-r\tau} \geq x_{t+\tau}\}, \tag{4.7}$$

which is almost twice faster than by (4.2), see Table 4.3.

Computing $n_d^\tau(t + \tau)$	+	+1	*	<>	Total
by (4.2) and (4.3)	d	$\leq d$	$d-1$	d	$\leq 4d-1$
by (4.5)	0	$\leq d$	0	d	$\leq 2d$

Table 4.3: Efficiency of computing the number $n_d^\tau(t + \tau)$ from the number $n_d^\tau(t)$

The precomputed tables of $\phi_d(n, l)$ for $d = 1, 2, \dots, 8$ can be downloaded from [Una14].

4.2.2 Storage requirements

In order to efficiently compute the numbers of OPs by (4.5), one has to store $(d+1)!(d+1)$ values of $\phi_d(n, l)$: $(d+1)$ values for each of the $(d+1)!$ OPs. This is not a very large size since usually the orders $d = 2, 3, \dots, 6$ are recommended for applications [BP02].

When using the enumeration (4.2) one can reduce the size of the table from $(d+1)!(d+1)$ to $(d+1)!$ which can be important for the applications restricted by the storage size. The OPs with the numbers $(d+1)(k-1), \dots, (d+1)k-1$ for each $k = 1, 2, \dots, d!$ describe the same relation between the last d points and therefore have the same successive OPs. For example, the OPs with the numbers 0, 1, 2 as well as the OPs with the numbers 3, 4, 5 have the same successive OPs: $\phi_d(0, l) = \phi_d(1, l) = \phi_d(2, l)$ and $\phi_d(3, l) = \phi_d(4, l) = \phi_d(5, l)$ (see Table 4.2).

4.3 Efficiently computing the empirical permutation entropy

In this section we consider computing ePE in maximally overlapping sliding windows of a size M (i.e., the first point of the successive window is the second point of the

previous one). The case with a non-maximal overlapping is discussed in Remark 12 (Subsection 4.3.1).

4.3.1 Precomputed values of empirical permutation entropy

The successive windows

$$(x_{t-M-d\tau}, x_{t-M-d\tau+1}, \dots, x_{t-1}) \text{ and } (x_{t-M-d\tau+1}, x_{t-M-d\tau+2}, \dots, x_t)$$

differ in the points x_t and $x_{t-M-d\tau}$, therefore the ordinal distributions in the windows differ in the frequencies of occurrence of the OPs with the numbers $n_d^\tau(t)$ and $n_d^\tau(t-M)$. In order to obtain an ordinal distribution in the successive window given the current one, one needs to recalculate the frequency of the number $n_{\text{out}} = n_d^\tau(t-M)$ of the “outcoming” OP and of the number $n_{\text{in}} = n_d^\tau(t)$ of the “incoming” OP if they do not coincide:

$$\begin{aligned} q_{n_{\text{out}}}(t) &= q_{n_{\text{out}}}(t-1) - 1, \\ q_{n_{\text{in}}}(t) &= q_{n_{\text{in}}}(t-1) + 1. \end{aligned} \quad (4.8)$$

Then the value $\text{ePE}(d, \tau, M, t)$ is computed from the value $\text{ePE}(d, \tau, M, t-1)$ by

$$\begin{aligned} \text{ePE}(d, \tau, M, t) &= \text{ePE}(d, \tau, M, t-1) + g(q_{n_{\text{out}}}(t-1)) - g(q_{n_{\text{in}}}(t-1) + 1), \\ \text{where } g(j) &= \frac{1}{M}(j \ln j - (j-1) \ln(j-1)) \text{ for } j = 1, 2, \dots, M. \end{aligned} \quad (4.9)$$

The precomputed table is obtained by computing $g(j)$ for all $j = 1, 2, \dots, M$ by (4.10). So the size of a precomputed tables is a size M of a sliding window. For example, a window size of two seconds is used in the example in Figure 4.1, i.e., $M = 2 \cdot 256 = 512$ for a sampling rate of 256 Hz.

In Table 4.4 we show that computing $\text{ePE}(d, \tau, M, t)$ by (4.9) is considerably faster than by (4.4).

Calculation	+	*	ln	Total
by (4.4)	$\leq (d+1)! - 1$	$(d+1)!$	$(d+1)!$	$\leq 3(d+1)! - 1$
by (4.9)	2	0	0	2

Table 4.4: Efficiency of computing the successive value $\text{ePE}(d, \tau, M, t)$ from the current value $\text{ePE}(d, \tau, M, t-1)$ by (4.9) in comparison with computing $\text{ePE}(d, \tau, M, t)$ by (4.4)

Remark 12. In the case of non-maximal overlapping between successive windows, one can also compute the values of ePE by (4.9) and omit “unnecessary” intermediate ePE values, see for illustration Figure 4.4.

Roughly speaking, for a distance $D < \frac{3(d+1)!-1}{2}$ between the successive windows computing the values of ePE by (4.9) is faster than by (4.4) despite of computing “unnecessary” values for intermediate windows.

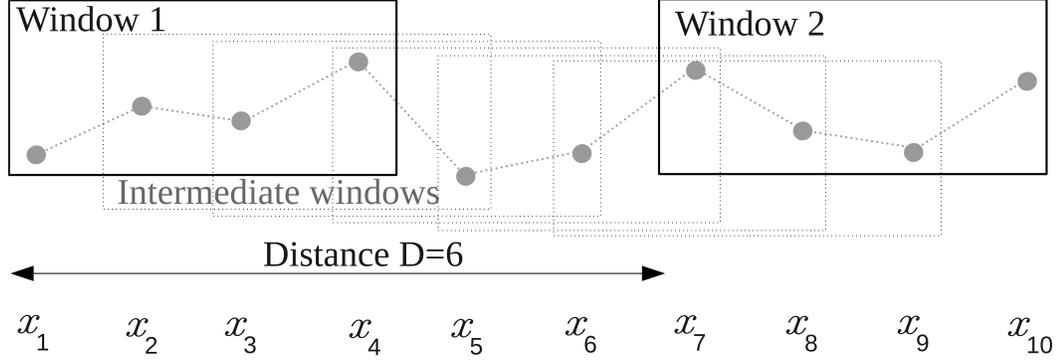


Figure 4.4: Non-maximal overlapping between successive windows

4.3.2 Round-off error

Computing the successive ePE values by (4.9) leads to an accumulation of relatively small round-off errors resulting from finite computer precision. Since there are two operations in (4.9) when computing ePE in sliding windows from a time series of the length W , the round-off error is bounded by $2W\psi$, where ψ is the machine precision. For instance, in Figure 4.5 we present the absolute values of the difference between the ePE values computed by (4.4) and the ePE values computed by (4.9), for the example shown in Figure 4.1.

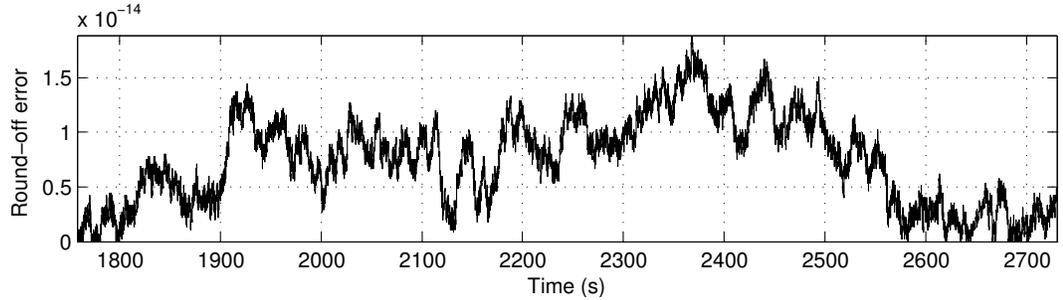


Figure 4.5: Absolute values of the difference between the ePE values computed by (4.4) and the ePE values computed by (4.9)

One can see that the error is very small in relation to the values of the ePE (compare with Figure 4.1).

For a relatively long time series one can recalculate the ePE by (4.4) after some time, depending on the computer precision, in order to avoid big accumulating errors and then continue calculations by (4.9). For example, if $\epsilon > 0$ is the maximal allowable error for computing the ePE, then one should recalculate the ePE by (4.4) every $\frac{\epsilon}{2\psi}$ points.

4.3.3 Scheme of the method

Given a time series $(x_t)_{t=1}^W$, a size of a sliding window M , an order d of OPs, and a delay τ , we summarize the method of efficient ePE computing in Figure 4.6.

One can see that the first value $\text{ePE}(d, \tau, M, t)$ is computed by (4.4) since there is no precomputed value $\text{ePE}(d, \tau, M, t - 1)$; numbers of the first τ OPs are computed by (4.2) since there are no precomputed values $n_d^\tau(t - \tau)$.

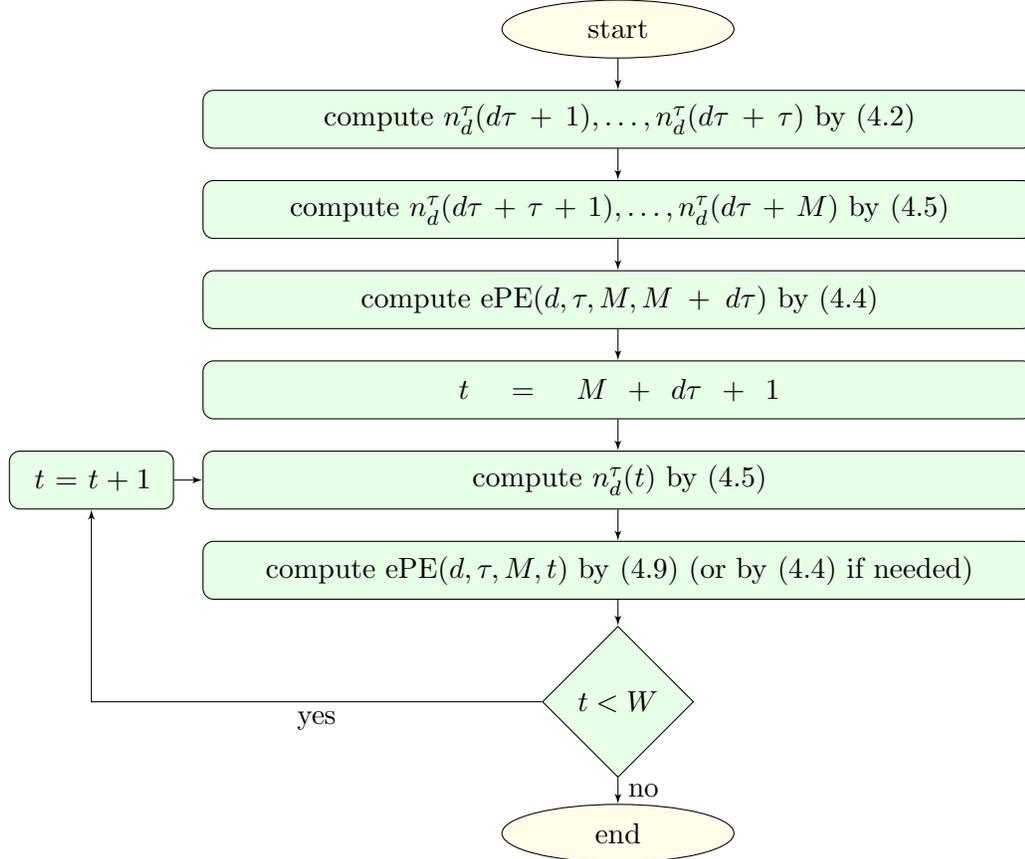


Figure 4.6: Algorithm of fast computing empirical permutation entropy in sliding windows

The MATLAB code for efficient ePE computing is given in Appendix A.1; it can be also downloaded from [Una14].

Note that, according to Figure 4.6, the efficiency of the proposed method (after some precomputing for the first window) depends only on a length of a time series W and order d (and depends neither on the window size M nor on the delay τ).

4.4 Efficiently computing numbers of ordinal patterns with tied ranks

In this section we adapt the method of efficient computing the numbers of OPs to the case of a time series with a high frequency of occurrence of equal values (*tied ranks*). To

this aim we define OPs with tied ranks (OPTs) in Subsection 4.4.1 and propose coding of them. In Subsection 4.4.2 we introduce the efficient enumeration of OPTs based on the same idea of precomputed values as for the “usual” OPs.

For illustration, we present all OPTs of order $d = 2$ in Figure 4.7.



Figure 4.7: The ordinal patterns with tied ranks of order $d = 2$

4.4.1 Ordinal patterns with tied ranks

Let us first provide a natural idea of the coding of ordinal patterns with tied ranks (OPTs) in Example 18, then we provide the formal definition of them.

Example 18. In Figure 4.8 we illustrate calculating an OPT of order $d = 5$. We go from the right to the left and we code by I_l the position of the point x_l with respect to the points on the right $x_{l+1}, x_{l+2}, \dots, x_6$.

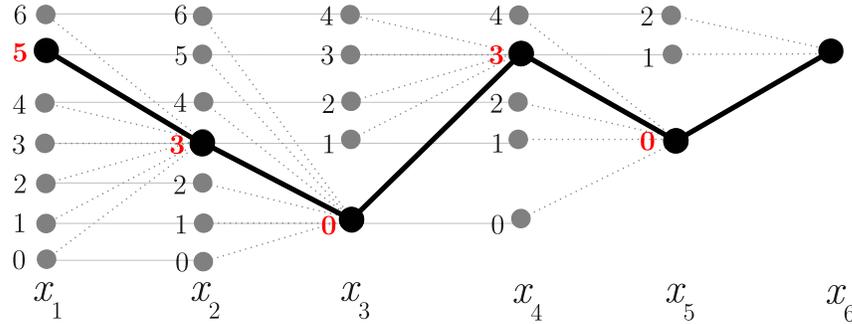


Figure 4.8: The ordinal pattern with tied ranks $(0, 3, 0, 3, 5)$

Definition 24. A delay vector $(x_t, x_{t-\tau}, \dots, x_{t-d\tau})$ is said to have the *ordinal pattern with tied ranks* (OPT) $I_d^r(t) = (I_1, I_2, \dots, I_d)$ of order $d \in \mathbb{N}$ and delay $\tau \in \mathbb{N}$ if for all $l = 1, 2, \dots, d$

$$I_l = b_l + 2\#\{r \in \{0, 1, \dots, l-1\} \mid x_{t-l\tau} > x_{t-r\tau}, b_r = 0\} \quad (4.11)$$

$$b_l = \begin{cases} 1 & \text{if } x_{t-l\tau} = x_{t-j\tau} \text{ for some } j \in \{0, 1, \dots, l-1\} \\ 0 & \text{otherwise.} \end{cases} \quad (4.12)$$

In the above definition b_l indicates whether the point $x_{t-l\tau}$ is equal to any point from $(x_t, x_{t-\tau}, \dots, x_{t-(l-1)\tau})$, I_l indicates the position of the point $x_{t-l\tau}$ in the vector $(x_t, x_{t-\tau}, \dots, x_{t-d\tau})$ as in Example 18, where $(I_1, I_2, I_3, I_4, I_5) = (0, 3, 0, 3, 5)$.

Note that the proposed coding of OPTs (compare with [BQMS12]) provides not only a concise representation of OPTs but also an efficient computing of the numbers of OPTs, see Subsection 4.4.2.

We assign to each OPT $I_d^\tau(t) = (I_1, I_2, \dots, I_d)$ a unique number $N_d^\tau(t) \in \{0, 1, \dots, (2d+1)!! - 1\}$ in the following way:

$$N_d^\tau(t) = N_d^\tau(I_d^\tau(t)) = \sum_{l=1}^d I_l(2l-1)!!, \quad (4.13)$$

where $!!$ stands for the odd factorial $(2l-1)!! = \prod_{j=1}^l (2j-1)$ (see Subsection 4.8.1 for the proof and the details of enumeration). The OPTs of order $d=2$ in their number representation are given in Table 4.5.

Ordinal pattern with tied ranks													
(I_1, I_2)	(0,0)	(1,0)	(2,0)	(0,1)	(1,1)	(2,1)	(0,2)	(1,2)	(2,2)	(0,3)	(2,3)	(0,4)	(2,4)
$N_2(I_1, I_2)$	0	1	2	3	4	5	6	7	8	9	11	12	14

Table 4.5: The ordinal patterns with tied ranks of order $d=2$ and their numbers

Remark 13. Note that there are “gaps” in the enumeration given by (4.13). For example, there are no OPTs of order $d=2$ corresponding to the numbers 10 and 13 (see Appendix 4.8.1 for details). One could provide an enumeration of OPTs without “gaps”, but the enumeration (4.13) provides an efficient computing of numbers of OPTs.

One obtains the successive OPT $I_d^\tau(t+\tau) = (I'_1, I'_2, \dots, I'_d)$ from the given one $I_d^\tau(t) = (I_1, I_2, \dots, I_d)$ (with (b_1, b_2, \dots, b_d)) by

$$I'_{l+1} = \begin{cases} I_l & \text{if } x_{t-l\tau} < x_{t+\tau} \text{ or } b_l = 1 \\ I_l + 1 & \text{if } x_{t-l\tau} = x_{t+\tau}, b_l = 0 \\ I_l + 2 & \text{if } x_{t-l\tau} > x_{t+\tau}, b_l = 0 \end{cases} \quad (4.14)$$

for $l = 0, 1, \dots, d-1$ with $I_0 = 0$ (compare with (4.3)). One needs, at most, $3d$ comparisons and, at most, d additions to obtain the successive OPT when the current one is given. This property is useful for a relatively fast computing OPTs, when one cannot use the precomputed table by some reason.

4.4.2 Precomputed numbers of ordinal patterns with tied ranks

Similar to the definition of the function ϕ_d in (4.5), we introduce here a function Φ_d for determining the number $N_d^\tau(t+\tau)$ of successive OPT from the given number $N_d^\tau(t)$ and from the position L of the next point¹.

¹In the following definition b_r is given by (4.12).

Definition 25. For $d, N, L \in \mathbb{N}$ we define a function

$$\Phi_d(N, L) = N_1 \quad (4.15)$$

such that for any vector $(x_1, x_2, \dots, x_{d+2}) \in \mathbb{R}^{d+2}$ for that N is the number of OPT of $(x_1, x_2, \dots, x_{d+1})$ and N_1 is the number of OPT of $(x_2, x_3, \dots, x_{d+2})$, it holds

$$L = B + 2\#\{r \in \{2, 3, \dots, d+1\} \mid x_r \geq x_{d+2}, b_r = 0\}, \text{ where} \quad (4.16)$$

$$B = \begin{cases} 1 & \text{if } x_{d+2} = x_j \text{ for some } j \in \{2, 3, \dots, d+1\}, \\ 0 & \text{otherwise.} \end{cases} \quad (4.17)$$

In the above definition L indicates the position of the next point x_{d+2} in relation to the points $(x_2, x_3, \dots, x_{d+1})$, B indicates whether the point x_{d+2} is equal to any point from $(x_2, x_3, \dots, x_{d+1})$.

The numbers $N_2^\tau(t + \tau)$ of successive OPTs of order $d = 2$ in dependence on the number of the current OPT $N_2^\tau(t)$ and on the position of the next point L are given in Table 4.6.

L	$N_2^\tau(t) \in \{0, 3, 6, 9, 12\}$	$N_2^\tau(t) \in \{1, 4, 7\}$	$N_2^\tau(t) \in \{2, 5, 8, 11, 14\}$
0	0	3	6
1	1	4	9
2	2	11	12
3	5	–	7
4	8	–	14

Table 4.6: The numbers $N_2^\tau(t + \tau) = \Phi_d(N_2^\tau(t), L)$ of successive ordinal patterns with tied ranks

In Table 4.7 one can see that computing the number $N_d^\tau(t + \tau)$ from $N_d^\tau(t)$ by (4.15) is faster than by (4.14).

Computation of $N_d^\tau(t + \tau)$	+	*	<>	Total
By (4.14)	$2d - 1$	d	$\leq 3d$	$\leq 6d - 1$
By (4.15)	$d + 1$	0	$\leq 3d$	$\leq 4d + 1$

Table 4.7: Efficiency of computing the number $N_d^\tau(t + \tau)$ of ordinal pattern with tied ranks from the number $N_d^\tau(t)$

The precomputed tables of numbers of successive OPTs for the orders $d = 1, 2, \dots, 6$ can be downloaded from [Una14]. The MATLAB code for computing ePE for the case of OPTs is given in Appendix A.3 and can also be downloaded from [Una14].

4.4.3 Storage requirements

In order to use (4.15) for the efficient computing of OPTs one has to store $(2d + 1)(2d + 1)!!$ values in the precomputed table. These are the values for each position $L = 0, 1, \dots, 2d$ for each of $(2d + 1)!!$ numbers of OPTs (although there are some empty entries, see for details Subsection 4.8.1). To give an impression we present storage requirements for “usual” OPs and for OPTs in dependence on the order d in Table 4.8.

Size of precomputed table	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$	$d = 6$	$d = 7$
for OPs, $(d + 1)(d + 1)!$	4	18	96	600	4320	35280	322560
for OPTs, $(2d + 1)(2d + 1)!!$	9	75	735	8505	114345	1756755	30405375

Table 4.8: Storage requirements for efficient computing the numbers of “usual” ordinal patterns and ordinal patterns with tied ranks

The enumeration given by (4.13) allows to reduce the table size, because one can group the numbers of OPTs of order d according to the same relations between the last d points (see Table 4.6, for example). To give an impression we also present the storage requirements for “usual” OPs and for OPTs in dependence on order d for short precomputed tables in Table 4.9.

Size of short precomputed table	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$	$d = 6$	$d = 7$
for OPs, $(d + 1)!$	2	6	24	120	720	5040	40320
for OPTs, $(2d + 1)!!$	3	15	105	945	10395	135135	2027025

Table 4.9: Storage requirements for efficient computing the numbers of “usual” ordinal patterns and ordinal patterns with tied ranks by using short precomputed tables

4.5 Efficiently computing empirical conditional entropy of ordinal patterns

In this section we adapt the method of efficient ePE computing to empirical conditional entropy of ordinal patterns (eCE), introduced in [KUU14]. The motivation for that is the fact that conditional entropy of ordinal patterns estimates the Kolmogorov-Sinai entropy better than ePE for several cases and can be also used in application (see [UK14, Una15] for details and examples).

In contrast to the ePE, the eCE is computed not only from the distribution of OPs but also from the distribution of $(2, d)$ -words that are pairs of successive OPs of order d (see Definition 7, p. 7). We describe efficient computing $(2, d)$ -words in Subsection 4.5.1 and efficient computing eCE from the distributions of $(2, d)$ -words in Subsection 4.5.2.

4.5.1 Efficiently computing $(2, d)$ -words

Here we propose a coding for the (n, d) -words for the case $n = 2$ in order to define eCE (see Definition 7 of (n, d) -words on p. 7).

Definition 26. A delay vector $(x_t, x_{t-\tau}, \dots, x_{t-(d+1)\tau})$ is said to have an $(2, d)$ -word $(i_d^\tau(t), i_d^\tau(t - \tau))$ when the OPs $i_d^\tau(t)$ and $i_d^\tau(t - \tau)$ are given by Definition 21.

There are $(d + 1)!(d + 1)$ possible $(2, d)$ -words, we assign to each of them a number from $\{0, 1, \dots, (d + 1)!(d + 1) - 1\}$ in a one-to-one way by

$$w_d^\tau(t) = w_d^\tau(i_d^\tau(t), i_d^\tau(t - \tau)) = (d + 1)n_d^\tau(t - \tau) + l, \quad (4.18)$$

$$\text{where } l = \#\{r \in \{1, 2, \dots, d\} \mid x_{t-r\tau} \geq x_t\}. \quad (4.19)$$

In Table 4.10 we illustrate that one needs only two more arithmetical operations to compute $w_d^\tau(t)$ together with $n_d^\tau(t)$ in comparison with computing $n_d^\tau(t)$ only, because the same l from (4.19) is used for both of them.

Computation of	+	+1	*	<>	Total
$n_d^\tau(t + \tau)$ by (4.5)	0	$\leq d$	0	d	$\leq 2d$
$n_d^\tau(t + \tau), w_d^\tau(t + \tau)$ by (4.5) and (4.18)	1	$\leq d$	1	d	$\leq 2d + 2$

Table 4.10: Efficiency of computing the successive numbers $n_d^\tau(t + \tau)$ and $w_d^\tau(t + \tau)$ from the current number $n_d^\tau(t)$

4.5.2 Efficiently computing empirical conditional entropy of ordinal patterns

We define here empirical conditional entropy of ordinal patterns introduced in [KUU14].

Definition 27. By the *empirical conditional entropy of ordinal patterns* (eCE) of order $d \in \mathbb{N}$ and of delay $\tau \in \mathbb{N}$ of a time-window $(x_t, x_{t-1}, \dots, x_{t-M-(d+1)\tau+1})$ one understands the quantity

$$\begin{aligned} \text{eCE}(d, \tau, M, t) &= \sum_{j=0}^{(d+1)!-1} \frac{q_j(t-1)}{M} \ln \frac{q_j(t-1)}{M} \\ &\quad - \sum_{j=0}^{(d+1)!(d+1)-1} \frac{p_j(t)}{M} \ln \frac{p_j(t)}{M}, \text{ where} \\ p_j(t) &= \#\{i = t, t-1, \dots, t-M+1 \mid w_d^\tau(i) = j\} \\ q_j(t-1) &= \#\{i = t-1, t-2, \dots, t-M+2 \mid n_d^\tau(i) = j\} \end{aligned} \quad (4.20)$$

(with $0 \ln 0 := 0$).

In order to obtain the distributions of OPs and of $(2, d)$ -words in the successive window $(x_t, x_{t-1}, \dots, x_{t-M-d\tau+1})$ given the current one $(x_{t-1}, x_{t-2}, \dots, x_{t-M-d\tau})$, one needs to recalculate the frequency of the “outcoming” $w_{\text{out}} = w_d^\tau(t-M)$ and “incoming” $w_{\text{in}} = w_d^\tau(t)$ $(2, d)$ -words if they do not coincide (as well as frequency of the “outcoming” $n_{\text{out}} = n_d^\tau(t-M)$ and “incoming” $n_{\text{in}} = n_d^\tau(t)$ OPs if they do not coincide):

$$\begin{aligned} p_{w_{\text{out}}}(t) &= p_{w_{\text{out}}}(t-1) - 1, & q_{n_{\text{out}}}(t) &= q_{n_{\text{out}}}(t-1) - 1, \\ p_{w_{\text{in}}}(t) &= p_{w_{\text{in}}}(t-1) + 1, & q_{n_{\text{in}}}(t) &= q_{n_{\text{in}}}(t-1) + 1. \end{aligned}$$

Then $\text{eCE}(d, \tau, M, t)$ given $\text{eCE}(d, \tau, M, t-1)$ is computed by

$$\begin{aligned} \text{eCE}(d, \tau, M, t) &= \text{eCE}(d, \tau, M, t-1) - g(q_{n_{\text{in}}}(t-1) + 1) + g(q_{n_{\text{out}}}(t-1)) \\ &\quad + g(p_{w_{\text{in}}}(t-1) + 1) - g(p_{w_{\text{out}}}(t-1)), \end{aligned} \quad (4.21)$$

where g is defined by (4.10). One can see that when computing (4.21) one uses the same precomputed tables for OPs and for the values of the function g as for computing ePE .

In Table 4.11 we show that one needs only two more arithmetical operations to calculate $\text{eCE}(d, \tau, M, t)$ from $\text{eCE}(d, \tau, M, t-1)$ compared with computing $\text{ePE}(d, \tau, M, t)$ from $\text{ePE}(d, \tau, M, t-1)$, which is much faster than direct computation by (4.20).

Calculation	+	*	ln	Total
$\text{ePE}(d, \tau, M, t)$ by (4.9)	2	0	0	2
$\text{eCE}(d, \tau, M, t)$ by (4.20)	$(d+2)! - 2$	$(d+2)! - 2$	$2(d+2)! - 7$	$4(d+2)! - 11$
$\text{eCE}(d, \tau, M, t)$ by (4.21)	4	0	0	4

Table 4.11: Efficiency of computing the successive value $\text{eCE}(d, \tau, M, t)$ from the current value $\text{eCE}(d, \tau, M, t-1)$ by (4.21) in comparison with computing by (4.20) and in comparison with computing the successive value $\text{ePE}(d, \tau, M, t)$ from the current value $\text{ePE}(d, \tau, M, t-1)$ by (4.9)

In Figure 4.9 we present the summarized algorithm of an efficient eCE computing.

Remark 14. Note that we also efficiently compute robust empirical permutation entropy, introduced in Chapter 3, on the basis of precomputing ordinal patterns, see Appendix A.5 for the realization in MATLAB.

4.6 Comparing efficiency of methods

In this section we illustrate the efficiency of the proposed method of computing the ePE with the efficiencies of the method introduced in [KSE07] and of the standard method available in the Internet (see “PE.m” in Appendix A.1, “oldPE.m” in Appendix A.2, “pec.m” from [Ouy14] for the realizations in MATLAB). For comparison, we present also the time of computing the ePE for OPTs, the time of computing empirical conditional

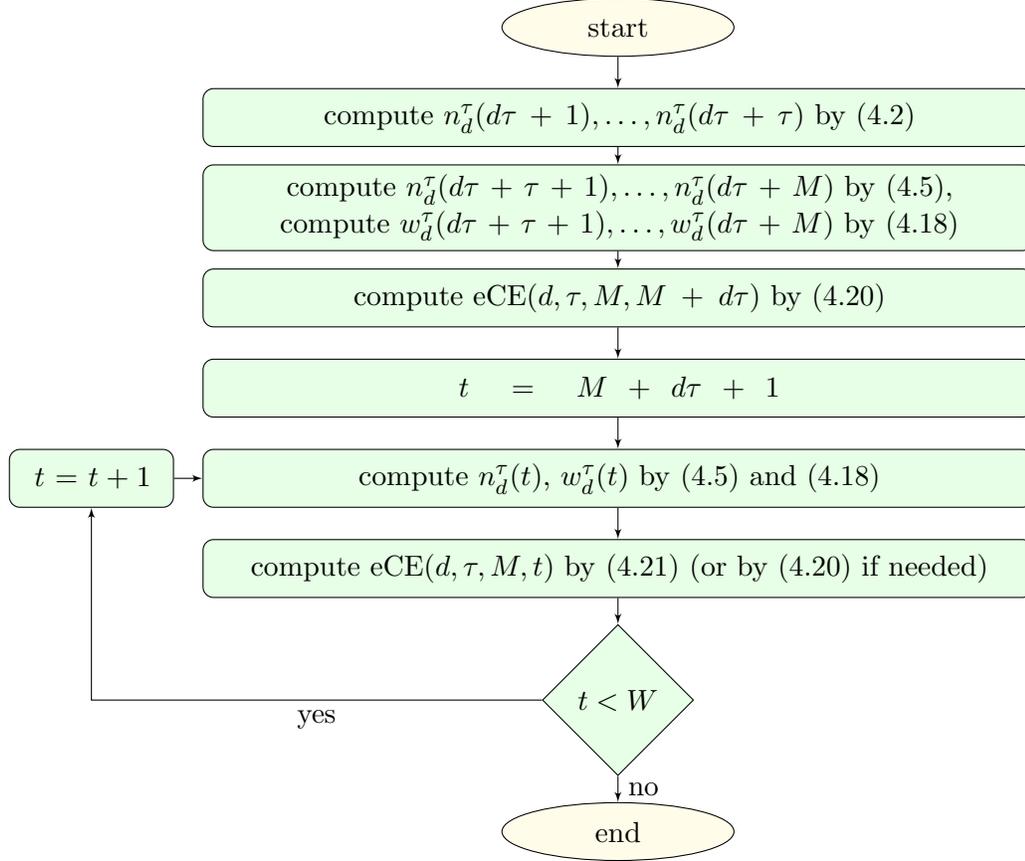


Figure 4.9: Algorithm of computing empirical conditional entropy of ordinal patterns in sliding windows

entropy of ordinal patterns and the time of computing robust empirical permutation entropy (see Appendix A.3-A.5 for the corresponding MATLAB scripts). For estimating the execution time of MATLAB scripts we use the MATLAB function “cputime”. (Note that CPU times computed by the MATLAB function “cputime” are dependent on PC and MATLAB version, in our case MATLAB 2013b and OS Linux 2.6.37.6-24, processor Intel(R) Core(TM) i5-2400 CPU @ 3.10Hz. The algorithms can work faster if programmed, for example, in C language.) The time is averaged over several trials.

In Table 4.12 the methods are compared for a one-channel EEG dataset recorded at a sampling rate of 256 Hz, for the orders $d = 3, 6, 7$, the delay $\tau = 4$ and different lengths of a time series (for other datasets similar results were obtained).

In Table 4.13 we compare the methods for a sliding window of 512 samples (2 s) for the same EEG dataset in dependence on the orders $d = 3, 6, 7$ and on the length W of a time series. A maximal overlapping between the sliding windows and the delay $\tau = 4$ are used. (Note that “pec.m” realized by G. Ouyang is not adapted for sliding windows and the corresponding results are not presented in Table 4.13.)

Length W	1000 s			2000 s			4000 s		
Order d	3	6	7	3	6	7	3	6	7
“pec.m”	8.02	933	7430	16	1869	14917	32	3733	29820
“oldPE.m”	1.17	1.19	1.19	2.32	2.36	2.39	4.66	4.76	4.77
“PE.m”	0.04	0.06	0.07	0.08	0.11	0.13	0.15	0.21	0.25
“PEeq.m”	0.52	0.57	1.12	1.03	1.08	1.67	2.08	2.15	2.76
“CondEn.m”	0.04	0.06	0.09	0.09	0.12	0.16	0.16	0.23	0.29
“rePE.m”	1.62	1.66	1.69	3.33	3.39	3.43	6.42	6.79	6.80

Table 4.12: Time (s) of computing in MATLAB R2013b (measured by a MATLAB function “cputime”) of the empirical permutation entropy, the empirical permutation entropy for the ordinal patterns with tied ranks, the empirical conditional entropy of ordinal patterns and the robust empirical permutation entropy from a time series of lengths W , $\tau = 4$

Length W	15 min			30 min			60 min		
Order d	3	6	7	3	6	7	3	6	7
“oldPE.m”	5.74	26.6	131	10.6	53.9	261	21.1	107	519
“PE.m”	0.10	0.11	0.11	0.19	0.19	0.22	0.35	0.39	0.42
“PEeq.m”	0.57	0.63	1.19	1.16	1.23	1.78	2.30	2.42	3.03
“CondEn.m”	0.18	0.19	0.22	0.34	0.38	0.43	0.68	0.76	0.84
“rePE.m”	6.04	29.2	138	12.0	58.4	279	23.8	115	549

Table 4.13: Time (s) of computing in MATLAB R2013b (by a MATLAB function “cputime”) the empirical permutation entropy, the empirical permutation entropy for the ordinal patterns with tied ranks, the empirical conditional entropy of ordinal patterns and the robust empirical permutation entropy from $\frac{W-d\tau}{M}$ sliding windows of size $M = 2$ s, $\tau = 4$

4.7 Conclusions

We conclude that

- the proposed method of efficient computing the numbers of ordinal patterns is almost two times faster than in [KSE07] (see Table 4.12), it can be also applied to fast computing different ordinal-patterns-based characteristics such as, for example, transcripts and ordinal distributions itself;
- the proposed method of efficient computing the empirical permutation entropy is considerably faster than the known methods ones (see Tables 4.12 and 4.13), and it allows to measure the complexity of large datasets in real-time;
- the proposed coding and enumeration of ordinal patterns with tied ranks is natural, convenient and allows for efficient computing of them;
- the proposed methods of computing empirical permutation entropy for ordinal

patterns with tied ranks, and the empirical conditional entropy of ordinal patterns are fast and can be applied to real-time processing of large datasets (see Table 4.13).

4.8 Supplementary materials

4.8.1 Number representation of ordinal patterns with tied ranks

Let us discuss first, why the enumeration (4.13) of OPTs has “gaps”, i.e. why some numbers computed by (4.13) do not correspond to any OPT. Consider the OPT $I_d^r(t) = (I_1, I_2, \dots, I_d)$ with the vector (b_1, b_2, \dots, b_d) , that indicates equalities between the points of the vector $(x_t, x_{t-\tau}, \dots, x_{t-d\tau})$ and is computed by (4.12). The more $b_r = 1$ for $r = 1, 2, \dots, l - 1$ are, the less the range of I_l is:

$$I_l \leq 2 \left(l - \sum_{r=1}^{l-1} b_r \right). \quad (4.22)$$

That is the more points in $(x_t, x_{t-\tau}, \dots, x_{t-d\tau})$ are equal to any other point, the less distinct values are in the vector. However, when enumerating OPTs by (4.13), we consider all possible combinations of $I_l \in \{0, 1, \dots, 2l\}$ for $l = 1, 2, \dots, d$, and, according to (4.22), some of these combinations do not correspond to any OPT. That is why the enumeration has “gaps”.

We show now that different OPTs of order d have different numbers computed by (4.13). Let us define a set \mathcal{I}_d of all vectors (I_1, I_2, \dots, I_d) as

$$\mathcal{I}_d = \{(I_1, I_2, \dots, I_d) \mid I_l \in \{0, 1, \dots, 2l\} \text{ for } l = 1, 2, \dots, d\}.$$

Proposition 24. *For every $d \in \mathbb{N}$, the assignment*

$$(I_1, I_2, \dots, I_d) \mapsto N_d((I_1, I_2, \dots, I_d)),$$

where $N_d((I_1, I_2, \dots, I_d))$ is computed by (4.13), defines a bijection from the set \mathcal{I}_d onto $\{0, 1, \dots, (2d + 1)!! - 1\}$.

Proof. Note that $N(I_1) = I_1$. Then by (4.13) for all $d \geq 2$ one has the recursion

$$N_d\left((I_l)_{l=1}^d\right) = N_{d-1}\left((I_l)_{l=1}^{d-1}\right) + (2d - 1)!! I_d$$

which by induction on d provides different $N_d((I_l)_{l=1}^d)$ for different (I_1, I_2, \dots, I_d) . \square

Note again that not all vectors from the set \mathcal{I}_d are OPTs according to (4.22), but all OPTs of order d have different numbers computed by (4.13).

4.8.2 Amount of ordinal patterns with tied ranks

One can see from (4.22) that there are less than $(2d + 1)!!$ OPTs due to the “gaps” in the enumeration. In order to find the actual amount of OPTs we observe that the OPTs of order d can be represented by *Cayley permutations* of the set $\{0, 1, \dots, d\}$ [MF84].

Definition 28. [MF84] A *Cayley permutation* of length d is a permutation ρ of d elements with possible repetitions from a set $\{x_1, x_2, \dots, x_d\}$ with $x_1 < x_2 < \dots < x_d$ and with an order relation, subject to the condition that if an element x_i appears in ρ , then all elements $x_j < x_i$ also appear in ρ .

Definition 28 means, in fact, that OPTs is one of possible coding of Cayley permutations. For example, we present in Table 4.10 the Cayley permutations of length 3 of a set $\{1, 2, 3\}$ and the corresponding coding by the OPs of order 2.

													
Cayley permutations	123	122	132	112	111	121	213	211	231	212	221	312	321
OPTs	(0,0)	(1,0)	(2,0)	(0,1)	(1,1)	(2,1)	(0,2)	(1,2)	(2,2)	(0,3)	(2,3)	(0,4)	(2,4)

Figure 4.10: Cayley permutations of order 3 of a set $\{1, 2, 3\}$ are coded by the ordinal patterns with tied ranks of order 2

The amount of Cayley permutations is counted using the *ordered Bell numbers* [MF84]. Therefore the amount of OPTs of order d is computed by the $(d + 1)$ -th ordered Bell number $B(d + 1)$ in the following way:

$$B(d + 1) = \sum_{k=0}^{d+1} \sum_{j=0}^k (-1)^{k-j} \frac{k!}{j!(k-j)!} j^{d+1}. \quad (4.23)$$

We present in Table 4.14 the amounts of numbers for OPTs computed by (4.13), the amounts of OPTs of order d that are computed by (4.23), and the amounts of “usual” OPs of order d .

Order d	1	2	3	4	5	6	7
the amount of numbers for OPTs, $(2d + 1)!!$	3	15	105	945	10395	135135	2027025
the amount of OPTs, $B(d + 1)$	3	13	75	541	4683	47293	545835
the amount of OPs, $(d + 1)!$	2	6	24	120	720	5040	40320

Table 4.14: Amounts of numbers for ordinal patterns with tied ranks, amounts of ordinal patterns with tied ranks and of ordinal patterns for different orders d

Chapter 5

Measuring complexity of EEG

Detecting seizures in epileptic electroencephalogram (EEG)¹ is an important problem in biomedical research nowadays [LE98, MWWM99, MAEL07, Leh08]. In this chapter we discuss seizure detection in epileptic EEG data from The Bonn EEG Database [Bon14] and from The European Epilepsy Database [Epi14] by empirical permutation entropy (ePE), robust empirical permutation entropy (rePE), approximate entropy (ApEn) and sample entropy (SampEn), see Chapter 3 for details about the entropies. We have three main purposes in this chapter. First, we discuss the applicability of ePE for the analysis of EEG data, in particular, the choice of the parameters for its computing. Second, we illustrate the potential of the rePE for application to real-world data. Third, we compare ePE, rePE, ApEn and SampEn for detecting epileptic seizures in EEG data.

Section 5.1 is devoted to the discussion of the choice of the parameters when applying ePE. Section 5.2 is intended to application of ePE, rePE, ApEn and SampEn for detecting epileptic seizures in the EEG data from [Epi14]. In Section 5.3 we compare ability of ePE, ApEn and SampEn to discriminate between different complexities of EEG data from [Bon14]. Finally, we make conclusions and discuss future work in Section 5.4.

5.1 Applying empirical permutation entropy for analysis of EEG data

In this section we work with EEG data from [Epi14]. We start from the data description in Subsection 5.1.1, then we discuss the choice of the parameters for ePE in Subsection 5.1.2.

5.1.1 Description of EEG data from The European Epilepsy Database

The European database contains multichannel surface EEG data sampled at 256 Hz [Epi14] (see [IFDT⁺12] for more information). In Tables 5.1 and 5.2 we provide the description of the EEG recordings that we use further in experiments. In Table 5.1 we

¹We refer to [TT09] for a good tutorial about recording, montages, and characteristics of EEG.

provide the information about the patients, in Table 5.2 we describe the EEG recordings with epileptic seizures, indicating types of seizures and human state (*vigilance*) in the period when a seizure occurs. According to [RK68, TT09], there are 6 stages of human vigilance: the *awake state* (W), two stages of *light sleep* (S1, S2), two stages of *deep or slow-wave* sleep (S3, S4) and *rapid eye movement* (REM) sleep.

Patient	Information	Etiology
1	male, 36 years old	malformation
89	female, 67 years old	-
308	male, 28 years old	malformation
454	female, 41 years old	hippocampal_sclerosis
586	male, 32 years old	tumor
795	female, 35 years old	-
852	female, 54 years old	hippocampal_sclerosis

Table 5.1: Description of the patients from [Epi14]

Patient	Numbers of recordings	Seizure type	Vigilance
1	86	UC	W
89	133, 140, 159, 164	CP	W
89	100	CP	UC
795	2, 27, 49, 54, 74, 79, 89, 99, 101	CP	W
454	37	CP	W
586	100	CP	W
852	32, 36, 86	CP	S2
852	39	CP	S1
852	50, 88, 90	CP	W
852	62, 74, 84	UC	W

Table 5.2: Description of the EEG recordings with seizures from [Epi14]; here CP and UC stand for complex partial and unclassified, correspondingly

Further in this section we refer to recording b from patient a as $a.b$, for example, 852_50 stands for the recording 50 from the patient 852.

5.1.2 Choice of delay, order, and EEG channel

In this subsection we discuss the influence of the choice of EEG channel (Example 19), of the delay τ (Examples 20-21), of the order d and of the window size N when applying the empirical permutation entropy $ePE(d, \tau, N)$ for epileptic seizure detection.

Recall that ePE should satisfy the weak stationarity requirement [BP02]. Therefore the size of a sliding window when computing ePE should be chosen in such a way that the distribution of ordinal patterns does not change in this window [BP02]. Size of a sliding window for EEG data is usually chosen equal to 2 s [BD00], however, we often

use in this chapter the sliding windows of 4 s size which provides good results in the experiments. The order d is in most cases set to $d = 4$, maximal for this window size according to the recommendation (3.19) (see Subsection 3.2.2, p. 43).

Example 19. Choice of EEG channel

EEG channels of the data from [Epi14] for every seizure are divided into four groups: the *origin* channels related to the origin of the seizure, the *early propagation* channels related to the early propagation of the seizure, the *late propagation* channels related to the late propagation of the seizure and all other (*unmarked*) channels. Note that one channel can belong to several groups. Usually, the ePE values reflect epileptic seizure better when computed from the origin, early and late propagation channels (in comparison to unmarked channels). To illustrate this, we present in Figure 5.1 the ePE values computed from the EEG recording 586_100 (see Tables 5.1, 5.2) for channels F4 (unmarked), C3 (early propagation) and F7 (origin).

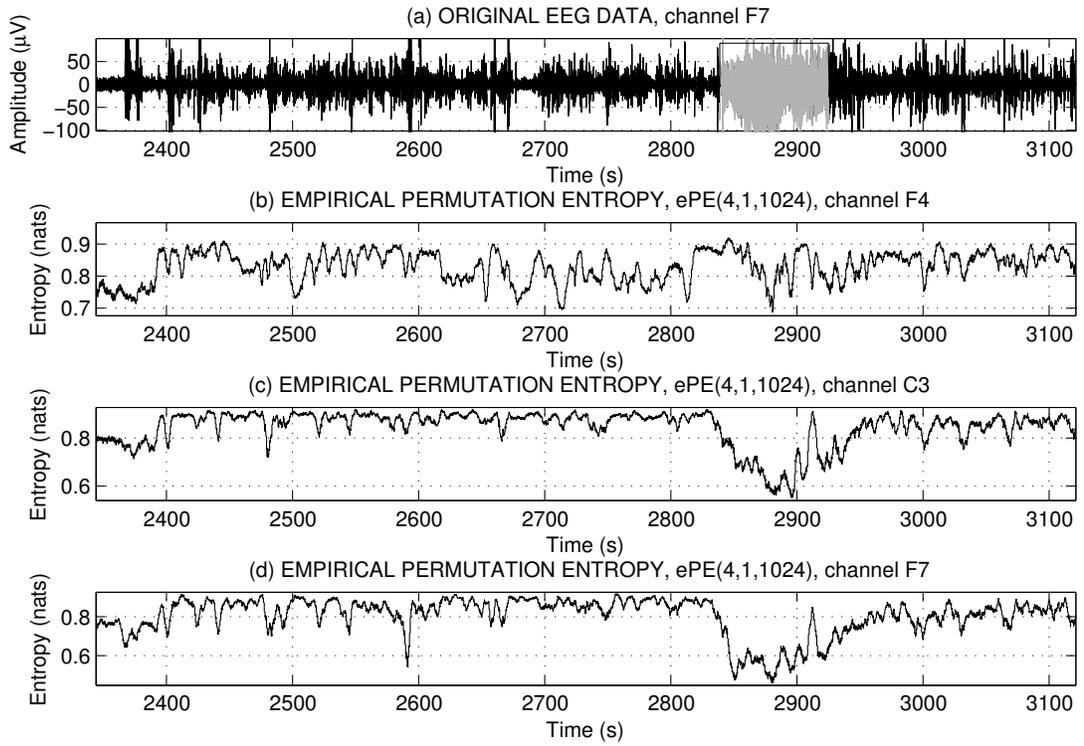


Figure 5.1: The values of the empirical permutation entropy computed from the EEG recording 586_100 for channels F4, C3, and F7

Indeed, the ePE values reflect the epileptic seizure for channels C3 and F7 by a decrease of its values whereas the seizure is almost not reflected for channel F4. The EEG data are filtered with a third order Butterworth bandpass filter, 2-42 Hz as proposed in [MDSBA08] for epileptic EEG data with the same sampling rate.

Example 20. Choice of delay τ

In this example we demonstrate that the $ePE(d, \tau, N)$ values computed for different delays τ illustrate different features of the underlying dynamics of EEG data. In Figure 5.2 we present the values of $ePE(4, \tau, 1024)$ computed from the EEG recording 795_74 (see Tables 5.1, 5.2), channel C3 (early propagation) for $\tau = 1, 6, 13$ (the seizure is marked in gray in the upper plot). The EEG data are filtered with a third order Butterworth bandpass filter, 2-42 Hz. One can see that the seizure is reflected by a decrease of the $ePE(4, \tau, 1024)$ values for $\tau = 6, 13$, whereas it is almost not reflected by the $ePE(4, \tau, 1024)$ values for $\tau = 1$.

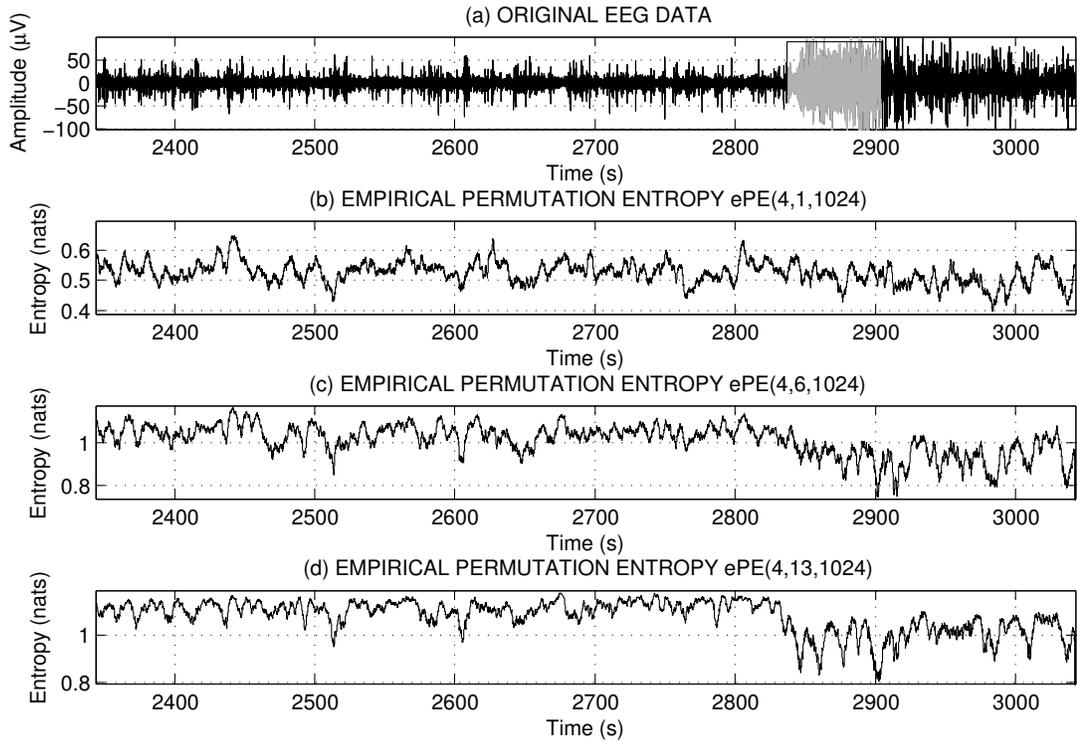


Figure 5.2: The values of the empirical permutation entropy $ePE(4, \tau, 1024)$ for different delays τ computed from the EEG recording 795_74, channel C3

However, values of $ePE(d, \tau, N)$ with relatively large τ do not always reflect epileptic seizures better than that for small τ . For example, in Figure 5.3 we present the ePE values computed from the EEG recording 586_100 (see Tables 5.1, 5.2), channel C3 (late propagation) for the delays $\tau = 1, 6, 13$. One can see that here the seizure is better reflected for $\tau = 1$ than for $\tau = 6, 13$ by a decrease of the $ePE(4, \tau, 1024)$ values. (Data are filtered with a third order Butterworth bandpass filter, 2-42 Hz.)

At the moment we cannot specify which τ reflect which features of EEG data, we recommend to consider different delays τ when analyzing EEG data by the ePE (see also Subsection 5.3.3).

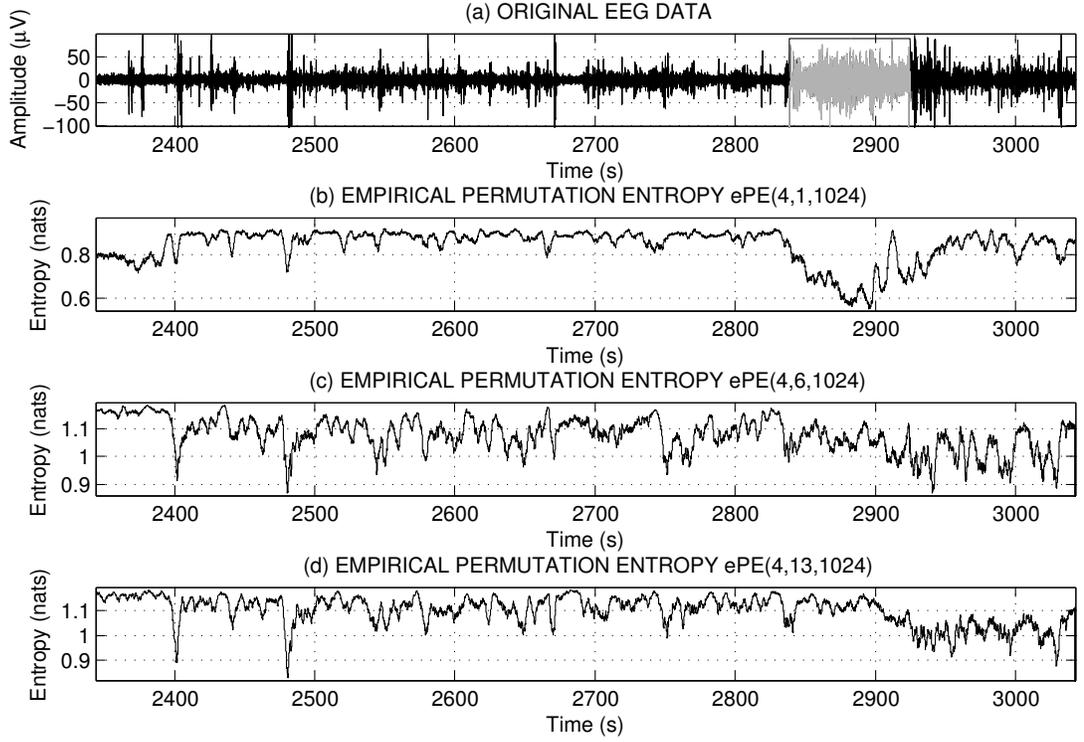


Figure 5.3: The values of the empirical permutation entropy $ePE(4, \tau, N)$ for different delays τ computed from the EEG recording 586_100, channel C3

Example 21. Possible problems when choosing delay τ

In this example we illustrate that an increase of delay τ can lead to an increase of the $ePE(d, \tau, N)$ values (see Example 12, p. 44 for theoretical background), and in this situation one should take into account the following bound:

$$ePE(d, \tau, N) \leq \frac{\ln((d+1)!)}{d}.$$

In Figure 5.4 we present the ePE values computed from the EEG recording 852_36 (see Tables 5.1, 5.2), channel C3 (late propagation) for the delays $\tau = 1, 6, 13$. One can see that an increase of delay τ leads to an increase of the $ePE(3, \tau, N)$ values such that they almost attain the upper bound $\frac{\ln((3+1)!)}{3} = 1.0594$ (dashed line). This does not allow to reflect the seizure by an increase of ePE values for $\tau = 6$ and $\tau = 13$. (The epileptic seizure is reflected by an **increase** of the ePE values here since the seizure occurs during a sleep stage S2, we discuss this in more details in Subsection 5.2.1.)

Note that there are two possible ways to avoid this problem when applying ePE for the analysis of real-world data:

- to use small delays τ ;
- to increase the order d if the length of a sliding window is large enough (see recommendation (3.19) in Subsection 3.2.2).

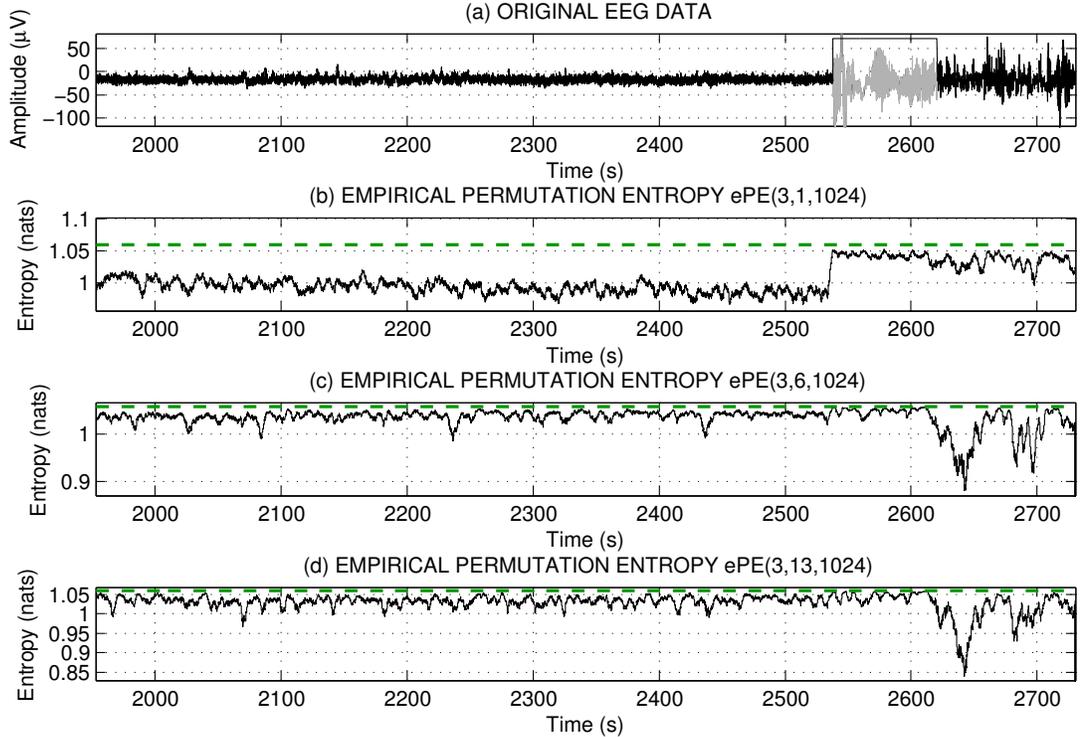


Figure 5.4: The values of the empirical permutation entropy $ePE(d, \tau, N)$ are bounded by $\frac{\ln((d+1)!)}{d}$ (dashed line), which does not allow to reflect the seizure for $\tau = 6$ and $\tau = 13$ for $d = 3$

5.2 Detecting epileptic seizures in EEG data by empirical permutation entropy, robust empirical permutation entropy, approximate entropy and sample entropy

In this section we discuss a detection of epileptic seizures by ePE, rePE, ApEn and SampEn for EEG data from [Epi14] (see description in Subsection 5.1.1).

- In Subsection 5.2.1 we demonstrate that the seizures that occur in the awake state (W) are often reflected by a decrease of the ePE, ApEn and SampEn values, whereas the seizures that occur during sleep are often reflected by an increase of the ePE, ApEn and SampEn values. Note that there are many results that report only a **decrease** of the ePE values during the seizure-related time (e.g. [LYLO14, CTG⁺04, KL03]).
- In Subsection 5.2.2 we demonstrate for short-term EEG data that rePE often provides better results than ePE, ApEn and SampEn for epileptic seizure detection.
- In Subsection 5.2.3 we demonstrate for long-term EEG data that rePE often provides better results than ePE for epileptic seizure detection.

Throughout the section we compute all entropies in maximally overlapping sliding windows of 4 s size, in all examples we use a third order Butterworth bandpass filter, 2-42 Hz as proposed in [MDSBA08] for EEG data of the same sampling rate. When computing the ApEn and SampEn values, we use the recommended for applications parameters $m = 2$ and $r = 0.2\sigma$, where σ is a standard deviation of a time series.

5.2.1 Detecting epileptic seizures in dependence on vigilance state

Example 22. An increase of the ePE, ApEn and SampEn values for the seizure occurred in sleep.

In Figure 5.5 we present the values of ePE, ApEn and SampEn computed from the EEG recording 852_36 (see Tables 5.1, 5.2), channel C4 (late propagation). One can see an increase of the ePE, ApEn and SampEn values during the time related to the seizure (marked in gray in the upper plot), which occurs in sleep stage S2.

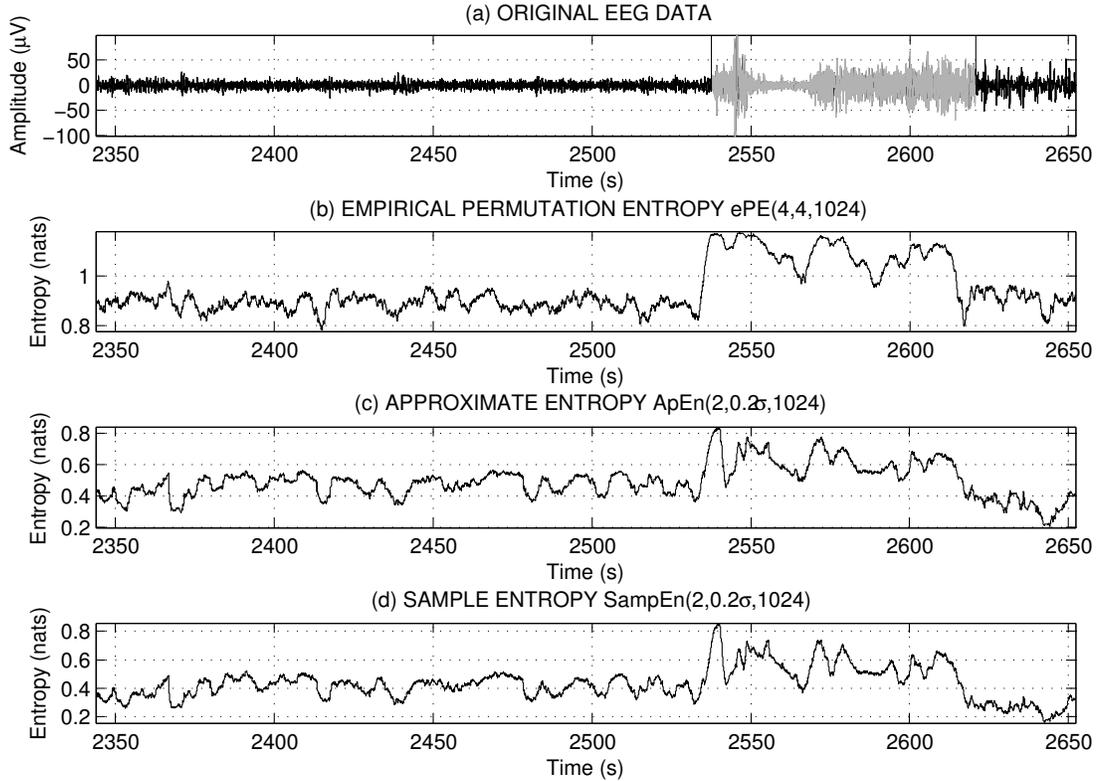


Figure 5.5: The values of the empirical permutation entropy, the approximate entropy and the sample entropy computed from the EEG recording 852_36, channel C4

This increase of complexity is explained by smaller, in general, values of the entropies, during the sleep stages S1-S4 [NG11, AFK⁺05]. Note that we have observed an increase of the ePE, ApEn and SampEn values during the seizure-related times if the seizures occur in the sleep stage S2 for many of the EEG recordings from [Epi14] (many exceptions

were observed for the recording 586 which is highly contaminated with noise). We assume that the same situation takes place also for the sleep stages S1, S3, S4 and REM, however, there are not enough recordings in [Epi14] to prove it.

Example 23. A decrease of the ePE, ApEn and SampEn values for the seizure occurred in the awake state.

In Figure 5.6 we present the values of ePE, ApEn and SampEn computed from the EEG recording 1.86 (see Tables 5.1, 5.2), channel F4 (early propagation). One can see a decrease of the ePE, ApEn and SampEn values during the time related to the seizure (marked in gray in the upper plot), the seizure occurs in the awake state (W).

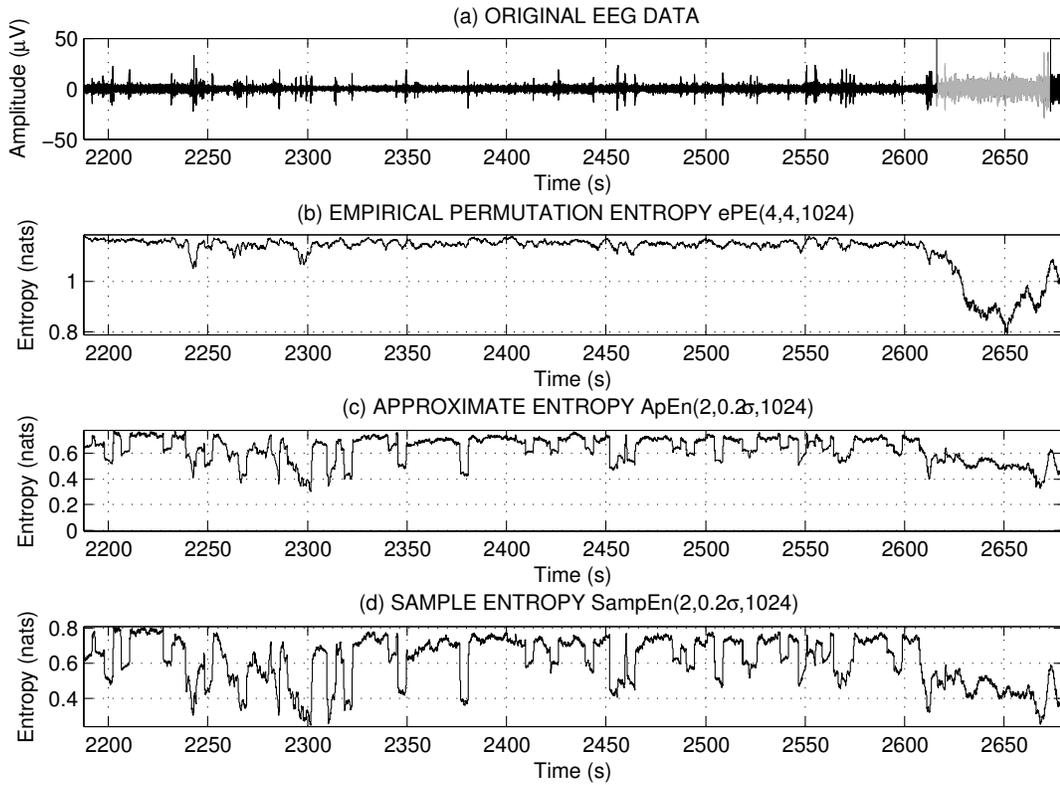


Figure 5.6: The values of empirical permutation entropy, approximate entropy and sample entropy computed from the EEG recording 1.86, channel F4

Note also a decrease of the ApEn and SampEn values in the time period 2240-2325 s, related to a sensitivity of ApEn and SampEn to artifacts², whereas the ePE values for this example are a bit more robust to artifacts. This illustrates that ApEn and SampEn need some preprocessing or combination with other complexity measures in order to detect seizures, otherwise they may provide many false alarms.

We have observed a decrease of the ePE, ApEn and SampEn values for many of

²EEG is often contaminated with different kinds of *artifacts* that are changes in EEG caused by eye movement, muscle activity, electrode movement, etc (see, e.g. [Lib12, TT09] for more details).

EEG recordings from [Epi14] during the seizure-related times if the seizures occur in the awake state. However, for some cases there is an increase the ePE values during the seizure-related times in the awake state, as it happens for the recordings from the patient 852 (see Example 27, p. 92). At the moment we do not have any rigorous explanation for that and we relate it to the individual features of EEG data for the patient 852.

5.2.2 Detecting epileptic seizures in short-term EEG data

In this subsection we illustrate that in many cases rePE values reflect epileptic seizures occurred in the awake state better than ePE, ApEn and SampEn values.

Remark 15. In this subsection we do not apply rePE for detecting epileptic seizures in sleep since ePE detects epileptic seizures in sleep rather well and rePE does not provide any significant improvement. We relate this to a small number of artifacts and noise in sleep EEG data.

Throughout this subsection we use the empirically chosen thresholds $\eta_1 = 0$ and $\eta_2 = 5$ when computing $\text{rePE}(d, \tau, N, \eta_1, \eta_2)$ from EEG data. This means that we do not use the lower threshold η_1 since we did not obtain improvement of the results when applying $\text{rePE}(d, \tau, N, \eta_1, \eta_2)$ with $\eta_1 > 0$ for the EEG data from [Epi14]. We also use the empirically chosen parameters $\tau = 4$ and $d = 4$ which have shown good results in the experiments.

Example 24. rePE values reflect epileptic seizures (occurred in the awake state) better than ePE, ApEn and SampEn values.

We present in Figure 5.7 the values of ePE, rePE and SampEn computed from the EEG recording 795_2 (see Tables 5.1, 5.2), channel F4 (late propagation). We do not present the ApEn values since they are very similar to the SampEn values. One can see that the seizure (marked in gray in the upper plot) is much better reflected by the values of rePE than by the values of ePE and SampEn. Note also that a large increase of the ePE and SampEn values at about 700 s, this increase is related, presumably, to an EEG artifact. This illustrates that a sensitivity of the ePE, SampEn and ApEn to EEG artifacts could hamper a correct epileptic seizure detection in EEG data by these measures. We have encountered this problem many times when analyzing the EEG data from [Epi14] by ePE, SampEn and ApEn. Meanwhile rePE often seems to be a better alternative for epileptic seizure detection in the awake state. In the following example we explain why the rePE values reflect epileptic seizures better than the ePE values.

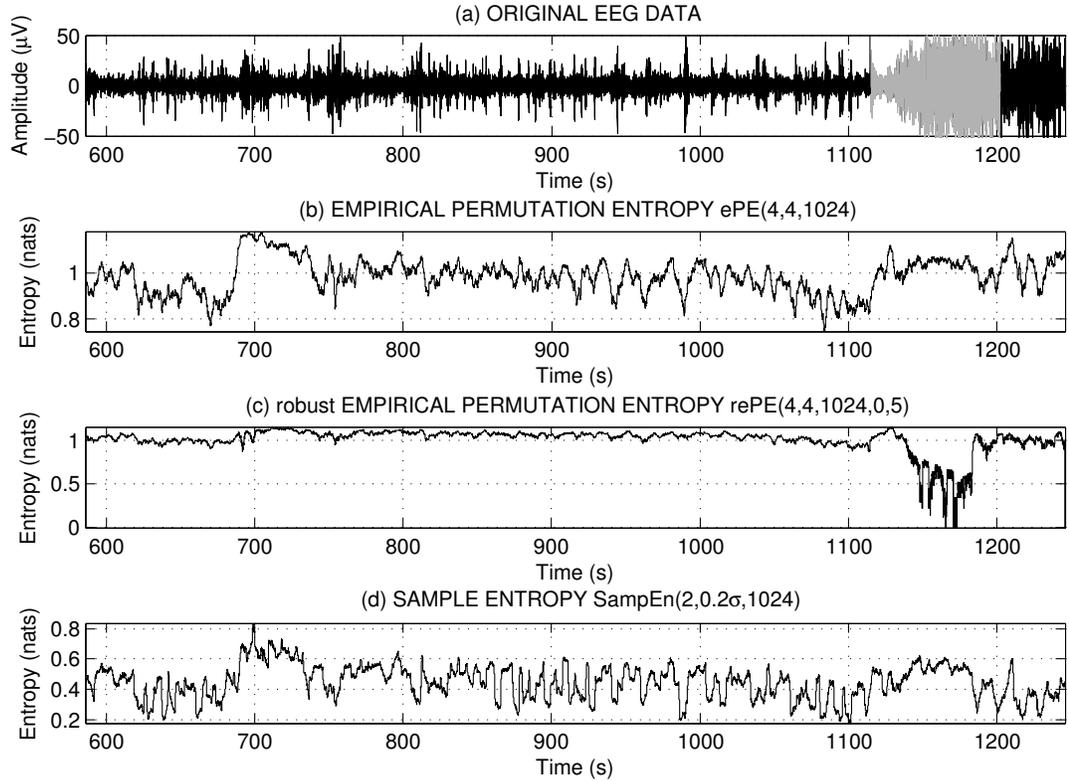


Figure 5.7: The values of the empirical permutation entropy, the robust empirical permutation entropy and the sample entropy computed from the EEG recording 795_2, channel F4

Example 25. Explanation for the rePE behavior.

We present in Figure 5.8 the values of rePE, ePE and MD (see Subsection 3.3.5, p. 52 for the MD definition) computed from the EEG recording 454_37 (see Tables 5.1, 5.2), channel F4 (unmarked). One can see the prominent decrease of the rePE values during the time related to the seizure (marked in gray in the upper plot), whereas the seizure is almost not reflected by the ePE values. This is explained by the increasing values of MD during the seizure-related time. This means that in the seizure-related time there are many pairs of points that are abnormally ($> \eta_2 = 5$) distant from each other and the corresponding ordinal patterns are not counted when computing rePE which leads to a decrease of the rePE values. Note also for this example a high sensitivity of ePE to EEG artifacts that, in particular, does not allow to reflect the epileptic seizure by the ePE values.

We conclude that for short-term EEG data in many cases rePE provides better results than ePE, ApEn and SampEn for epileptic seizure detection if seizures occurred in the awake state. However, further investigation of rePE is necessary, especially the choice of the thresholds η_1, η_2 is of interest.

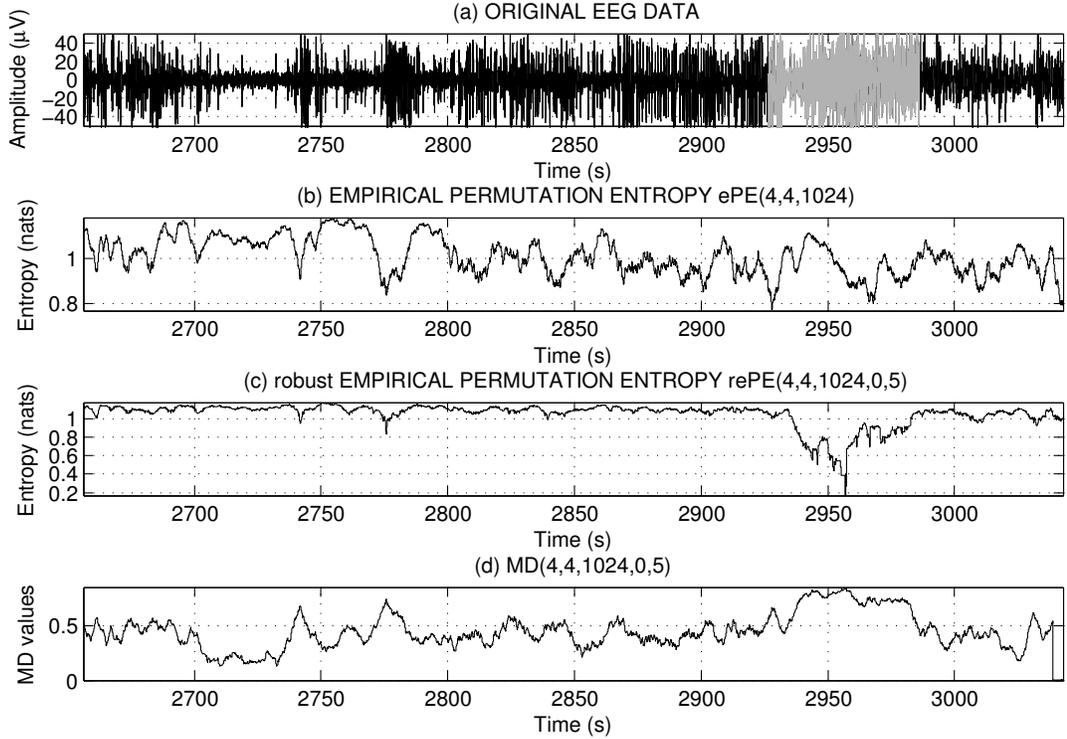


Figure 5.8: The values of the empirical permutation entropy and the robust empirical permutation entropy computed from the EEG recording 454_37, channel F4

5.2.3 Detecting epileptic seizures in long-term EEG data

In this subsection we illustrate epileptic seizure detection for long-term EEG data by ePE and rePE. We do not compare here ePE and rePE with ApEn and SampEn since, first, ApEn and SampEn do not show good results in epileptic seizure detection for short-term EEG recordings (see Example 24) and, second, ApEn and SampEn are very time-consuming in comparison with ePE and rePE (see Subsection 4.6).

Example 26. Epileptic seizure detection for the EEG recordings from the patient 795.

In this example we use the empirically chosen thresholds $\eta_1 = 0$, $\eta_2 = 4$ and the parameters $d = 4$, $\tau = 4$ that provided good results for epileptic seizure detection in the awake state by rePE for several long-term EEG recordings. In Figure 5.9 we present the values of ePE and rePE computed from the EEG recordings from patient 795 (see Tables 5.1, 5.2), channel C3. We indicate the seizures by green vertical lines. Note that all the seizures for this patient occurred in the awake state (see Table 5.2). We mark the ePE and rePE values less than 0.6 and 0.1, correspondingly, by **x**. Here all the seizures are reflected by decreases of the rePE values, whereas the ePE values almost do not reflect the seizures (also for other values of τ). However, there are some decreases of the rePE values that are, presumably, related to EEG artifacts, but not to the seizures.

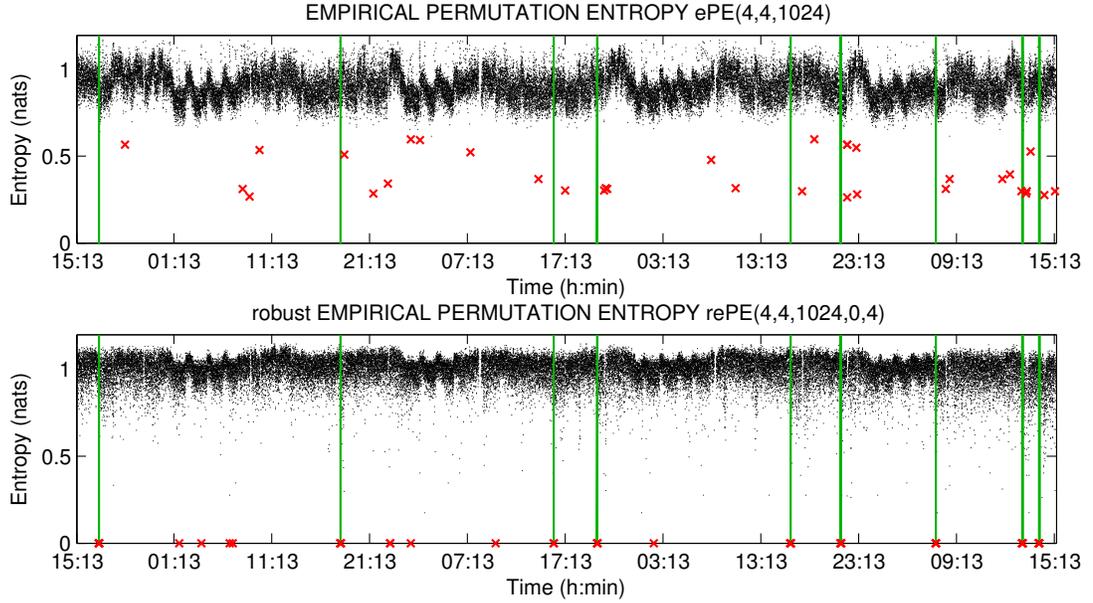


Figure 5.9: The values of empirical permutation entropy (ePE) and robust empirical permutation entropy (rePE) computed from the EEG recordings from patient 795, channel C3, the seizures are indicated by the green vertical lines, the ePE and rePE values, less than 0.6 and 0.1, correspondingly, are marked by \times

We present a MATLAB script for processing the EEG recordings from the patient 795 by ePE and rePE and for plotting the results (Figure 5.9) in Appendix A.6.

Example 27. Sleep stages separation for the EEG recording 852.

While analyzing the EEG data from [Epi14], we have found out that the ePE and rePE values seem to separate the sleep stages. Here we illustrate two points: presumable sleep stages separation and seizure detection by the ePE and rePE values. To this aim, in Figure 5.10 we present values of $ePE(3, 1, 1024)$ and $rePE(3, 1, 1024, 0.5, 3)$ computed from the EEG recordings from the patient 852 (see Tables 5.1, 5.2), channel C3. We indicate the seizures by green vertical lines. We mark also the ePE and rePE values higher than 0.69 and 0.84, correspondingly, by \times . Note also that for this patient the ePE values increase during the seizure-related times when seizures occurred in the awake state (see Subsection 5.2.1 for discussion).

One can see the four clear prominent decreases of the ePE and rePE values related to sleep (see the time axis to detect probable sleep during night time). Note that the sleep stages are hardly seen in Figure 5.9 when the values of ePE and rePE are computed for $\tau = 4$. We also note lower rePE variance during sleep. However, detecting sleep stages by ePE is beyond the scope of this thesis, we refer to [NG11, KUU14, Una15] for some interesting results in this direction.

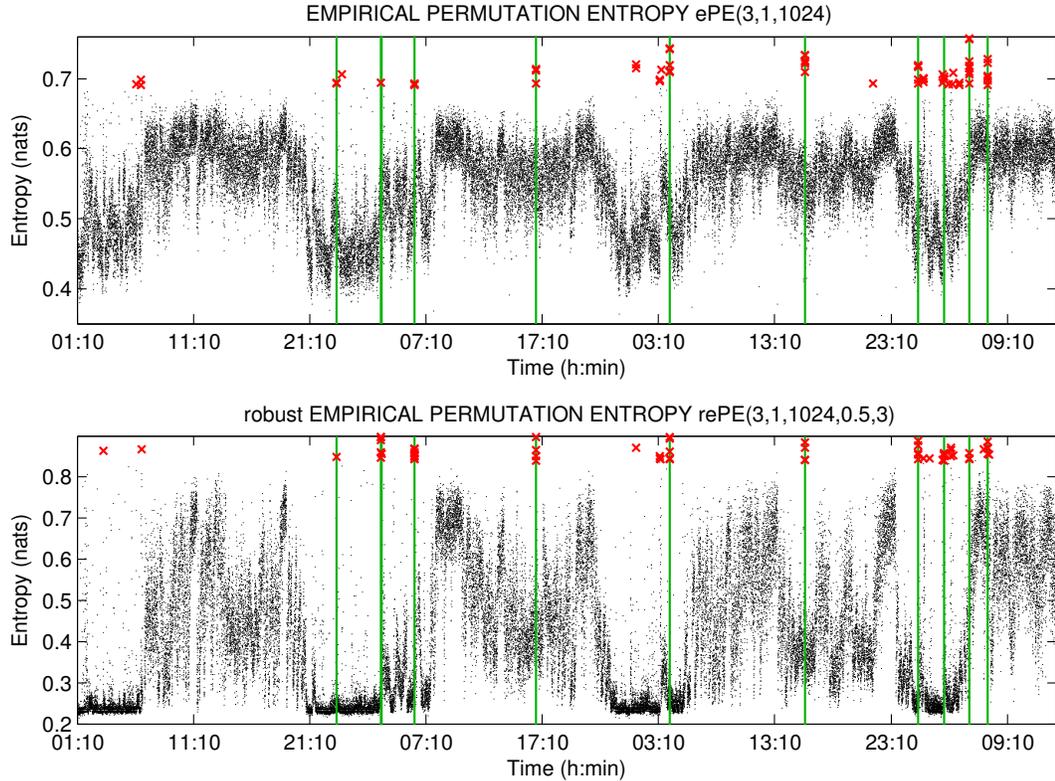


Figure 5.10: The values of empirical permutation entropy (ePE) and the robust empirical permutation entropy (rePE) computed from the EEG recordings from the patient 852, channel C3, the seizures are indicated by the green vertical lines, the ePE and rePE values, higher than 0.69 and 0.84, correspondingly, are marked by \times

We conclude that it is advantageous to combine non-metric ePE with using some metric information to detect epileptic seizures, like we did when using rePE. The rePE values often correctly reflect epileptic seizures (occurred in the awake state), however, there are still many “false alarms” related to artifacts.

Remark 16. (Prediction of epileptic seizures) We would like to emphasize that due to the sensitivity of ePE to artifacts, noise, choice of the channel and delay τ we do not speak about prediction of epileptic seizures in this chapter. However, in some cases the ePE values provide something that looks like a prediction of an epileptic seizure. For example, in Figure 5.11 we present the ePE values computed from the EEG recording 89_140 (see Tables 5.1, 5.2) for channels F4 (late propagation), C3 (early propagation) and F7 (origin). One can see that the ePE values start to increase about 40 s before the seizure (marked in gray in the upper plot), which is rather typical for all the seizures from this patient except for the EEG recording 89_133. However, these changes can be, presumably, related to change of EEG activity (e.g. change of sleep stage) that happens before the seizure. Note also that similar (looking like prediction) results were obtained for the patient 308.

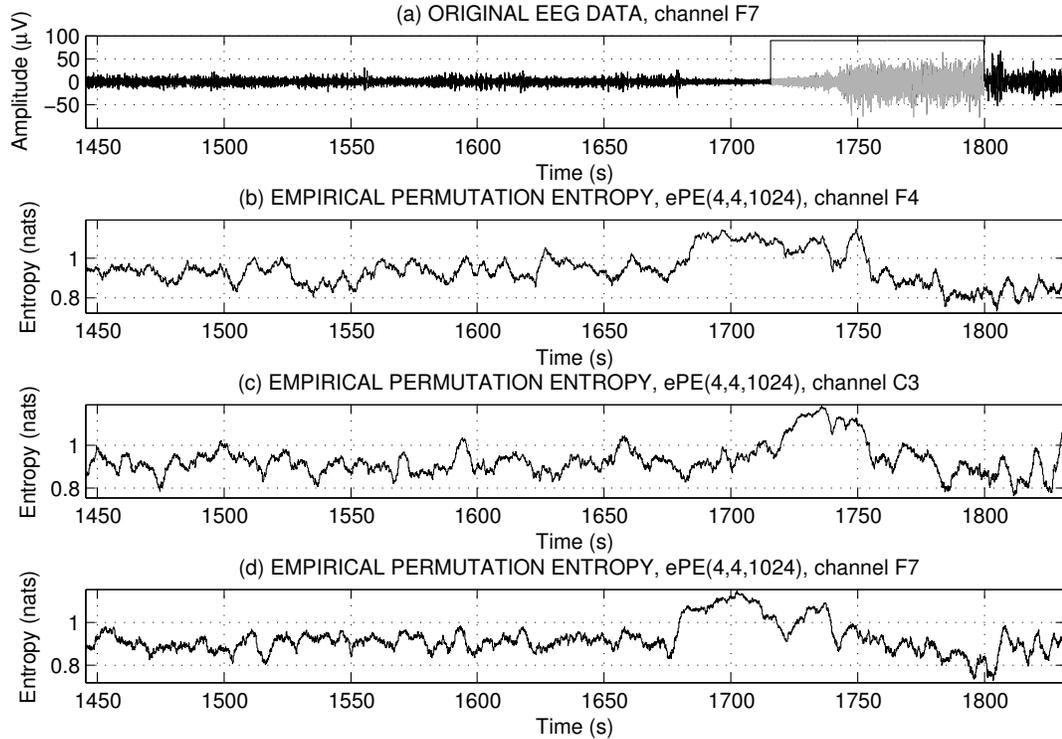


Figure 5.11: The values of the empirical permutation entropy computed from the EEG recording 89_140, channels F4, C3 and F7

5.3 Discrimination between different complexities of EEG data

In this section we illustrate the ability of ApEn, SampEn and ePE to discriminate between different complexities of EEG recordings from the Bonn EEG Database [Bon14]. Note that some results from this section are already presented in [KUU14]. In this section we do not make any preprocessing for EEG data.

5.3.1 Description of EEG data from the Bonn EEG Database

There are five groups of recordings [ALM⁺01]:

- A – surface EEG recorded from healthy subjects with open eyes;
- B – surface EEG recorded from healthy subjects with closed eyes;
- C – intracranial EEG recorded from subjects with epilepsy during a seizure-free period from hippocampal formation of the opposite hemisphere of the brain;
- D – intracranial EEG recorded from subjects with epilepsy during a seizure-free period from within the epileptogenic zone;

- E – intracranial EEG recorded from subjects with epilepsy during a seizure period.

Each group contains 100 one-channel EEG recordings of 23.6 s duration recorded at a sampling rate of 173.61 Hz, the recordings are free from artifacts; we refer to [ALM⁺01] for more details. For simplicity we refer further to the recordings from the group A as “recordings A”, to the recordings from the group B as “recordings B” and so on.

Note that we take here the entire original time series since they fulfill a weak stationarity criterion formulated in [ALM⁺01, Section IIB2], see also Subsection 5.1.2 for discussion about choice of window size.

5.3.2 Discriminating recordings by empirical permutation entropy, approximate entropy and sample entropy

In Figure 5.12 one can see that ApEn and SampEn separate the recordings A and B from the recordings C, D and E, whereas ePE separates the recordings E from the recordings A, B, C and D. Therefore it is a natural idea to present the values of ePE versus the values of ApEn and SampEn for each recording. Indeed, in Figure 5.13 one can see a good separation between the recordings A and B, the recordings C and D, and the recordings E.

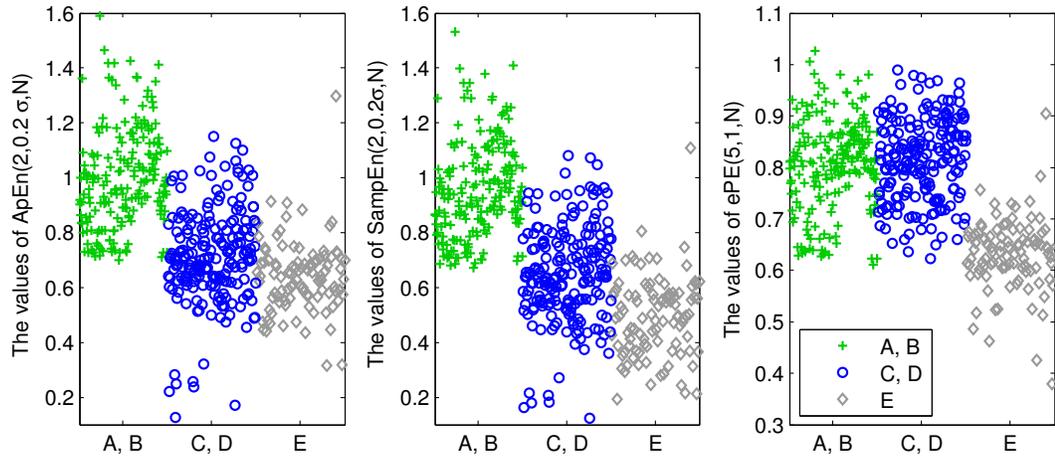


Figure 5.12: The values of the entropies computed from the recordings from the groups A and B, C and D, and E; $N = 4097$, σ stands for the standard deviation of a time series

5.3.3 Discriminating recordings by empirical permutation entropy computed for different delays

We have found also that varying the delay τ can be used for separation between the groups of recordings when computing ePE. In Figure 5.14 we present the values $ePE(4, \tau, N)$ for $\tau = 1, 2, 3$. Note an increase of the ePE values for an increase of τ , we plot also an upper bound $\frac{\ln(5!)}{4}$ (blue dashed line) for $ePE(4, \tau, N)$ to show why we do

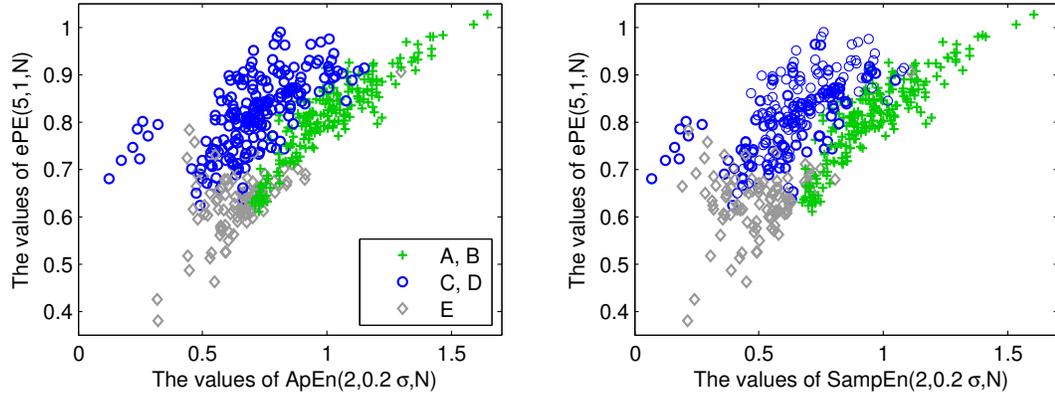


Figure 5.13: The values of empirical permutation entropy versus the values of approximate entropy and sample entropy; $N = 4097$, σ stands for the standard deviation of a time series

not consider $\tau > 3$ here. One can see that the delay $\tau = 1$ provides a separation of the recordings E from other recordings, whereas the delay $\tau = 3$ provides a separation of the recordings A and B from other recordings. The natural idea now is to present the values of ePE versus themselves for different delays τ .

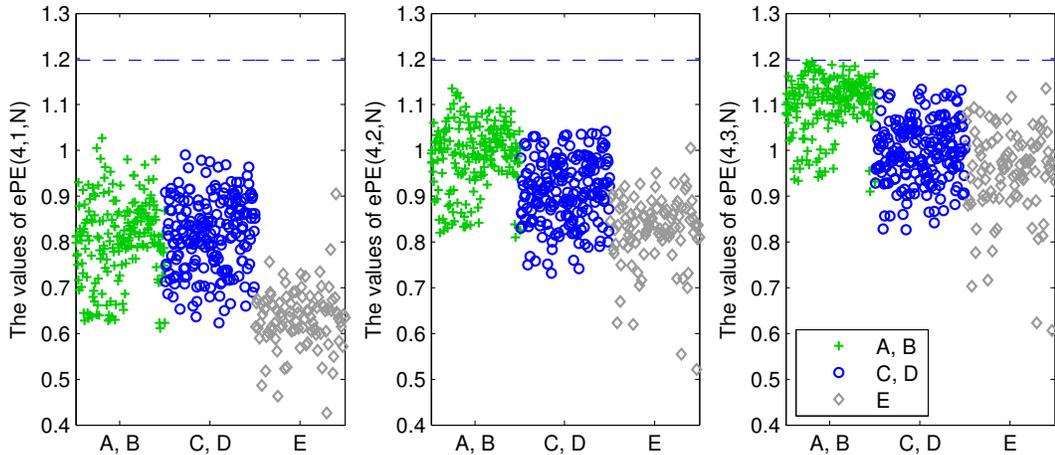


Figure 5.14: The values of the empirical permutation entropy computed from the recordings A and B, C and D, and E, $N = 4097$

In Figure 5.15 one can see a good separation between the recordings A and B, the recordings C and D, and the recordings E.

The results of discrimination are presented only for illustration of discrimination ability of ePE, ApEn and SampEn, therefore we do not compare the results with other methods (for a review of classification methods applied for the Bonn data set see [TTF09]). Note also that we did not use any preprocessing of data, we even found that filtering worsen the results of discrimination.

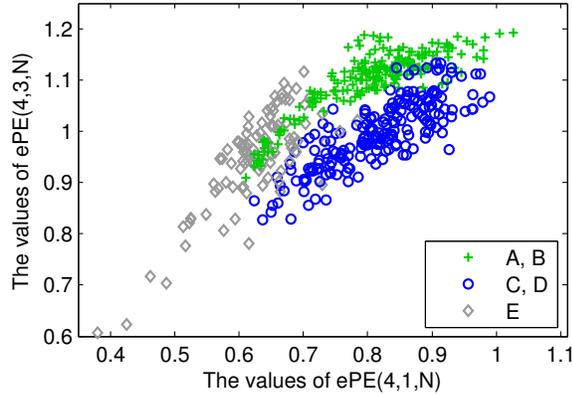


Figure 5.15: The values of $ePE(5, 1, N)$ versus the values of $ePE(5, 3, N)$, $N = 4097$

Application of the entropies to the epileptic data from Bonn EEG Database has shown the following.

- It can be useful to apply approximate entropy, sample entropy and empirical permutation entropy together since they reveal different features of the dynamics underlying a time series.
- It can be useful to apply empirical permutation entropy for different delays τ since they reveal different features of the dynamics underlying a time series.

See also [NG12] for use of permutation entropy for automated epileptic seizure detection with the help of support vector machines for the same data set as we considered in the section.

5.4 Conclusion

On the basis of the experiments performed for the EEG data from [Epi14] and [Bon14] by the empirical permutation entropy (ePE), robust empirical permutation entropy (rePE), approximate entropy (ApEn) and sample entropy (SampEn) we make the following conclusions.

- In many cases the ePE, ApEn and SampEn values decrease during the seizure-related time when the seizure occurs in the awake state, whereas the ePE, ApEn and SampEn values increase during the seizure-related time when the seizure occurs during sleep.
- The ePE values strongly depend on the chosen EEG channel. By this reason we recommend to compute the ePE values from all the EEG channels when analyzing EEG data by ePE.

- The ePE, ApEn and SampEn are very sensitive with respect to EEG artifacts which hampers correct detecting the epileptic seizures in the awake state in EEG data.
- The rePE is a promising new quantity; according to the results of our experiments, it often provides better results than ePE, ApEn and SampEn for epileptic seizure detecting in the awake state. However, rePE requires further investigation. In general, it is useful to combine ePE with using some metric information obtained from a time series.
- It is useful to combine ePE with ApEn and SampEn since it allows to reveal different features of underlying system and it is helpful for discrimination of time series with different underlying complexity.
- It is useful to combine ePE computed for different delays τ since it allows to reveal different features of underlying system and it is helpful for discrimination of time series with different underlying complexity.

Now we list the promising directions of future studies and applications of the ePE and the rePE.

- study of the thresholds η_1, η_2 when applying $\text{rePE}(d, \tau, N, \eta_1, \eta_2)$ to real-world data;
- study of various delays τ when applying ePE and rePE to real-world data;
- study of the variances of ePE and rePE which can be particularly interesting for epileptic seizures prediction or sleep stages separation;
- further study of combining the ePE with different measures of complexity and with using metric information from a time series;
- further application of rePE to real-world data, in particular, for sleep stages separation.

Appendix A

MATLAB code

A.1 Computing empirical permutation entropy by the new method

Figures/PE.m

```
% PE.m - algorithm for the fast calculation of an empirical
% permutaion entropy in maximally overlapping sliding windows
% the function is realized by the method proposed in [UK13]

% INPUT (x - the considered time series, Tau - a delay,
% d - an order of the ordinal patterns, WS - size of a sliding window)
% OUTPUT [ ePE - the values of empirical permutation entropy]

function ePE = PE(x, Tau, d, WS)
load(['table' num2str(d) '.mat']);% the precomputed table
pTbl = eval(['table' num2str(d)]);
Length = numel(x);           % length of the time series
d1 = d+1;
dTau = d*Tau;
nPat = factorial(d1);        % amount of ordinal patterns of order d
opd = zeros(1, nPat);       % distribution of ordinal patterns
ePE = zeros(1, Length);     % empirical permutation entropy
op = zeros(1, d);           % ordinal pattern (i1,i2,...,id)
prevOP = zeros(1, Tau);    % previous ordinal patterns for 1:tau
opW = zeros(1, WS);        % ordinal patterns in the window
ancNum = nPat./factorial(2:d1); % ancillary numbers
peTbl(1:WS) = -(1:WS).*log(1:WS); % table of values g(j)
peTbl(2:WS) = (peTbl(2:WS)-peTbl(1:WS-1))./WS;
for iTau = 1:Tau
    cnt = iTau;
    op(1) = (x(dTau+iTau-Tau) >= x(dTau+iTau));
    for j = 2:d
        op(j) = sum(x((d-j)*Tau+iTau) >= x((d1-j)*Tau+iTau:Tau:dTau+iTau));
    end
    opW(cnt) = sum(op.*ancNum); % the first ordinal pattern
    opd(opW(cnt)+1) = opd(opW(cnt)+1)+1;
    for j = dTau+Tau+iTau:Tau:WS+dTau % loop for the first window
        cnt = cnt+Tau;
        posL = 1; % the position l of the next point
        for i = j-dTau:Tau:j-Tau
            if(x(i) >= x(j))
                posL = posL+1;
            end
        end
        opW(cnt) = pTbl(opW(cnt-Tau)*d1+posL);
        opd(opW(cnt)+1) = opd(opW(cnt)+1)+1;
    end
    prevOP(iTau) = opW(cnt);
end
end
```

```

ordDistNorm = opd/WS;
ePE(WS+Tau*d) = -nansum(ordDistNorm(1:nPat).*log(ordDistNorm(1:nPat)));

iTau = mod(WS, Tau)+1;           % current shift  $l:\tau$ 
iPat = 1;                         % position of the current pattern in the window
for t = WS+Tau*d+1:Length        % loop over all points
    posL = 1;                      % the position  $l$  of the next point
    for j = t-dTau:Tau:t-Tau
        if(x(j) >= x(t))
            posL = posL+1;
        end
    end
    nNew = pTbl(prevOP(iTau)*d1+posL); % "incoming" ordinal pattern
    nOut = opW(iPat);                % "outcoming" ordinal pattern
    prevOP(iTau) = nNew;
    opW(iPat) = nNew;
    nNew = nNew+1;
    nOut = nOut+1;
    if nNew ~= nOut                % update the distribution of ordinal patterns
        opd(nNew) = opd(nNew)+1; % "incoming" ordinal pattern
        opd(nOut) = opd(nOut)-1; % "outcoming" ordinal pattern
        ePE(t) = ePE(t-1)+(peTbl(opd(nNew))-peTbl(opd(nOut)+1));
    else
        ePE(t) = ePE(t-1);
    end
    iTau = iTau+1;
    iPat = iPat+1;
    if(iTau > Tau) iTau = 1; end
    if(iPat > WS) iPat = 1; end
end
ePE = ePE(WS+Tau*d:end);

```

A.2 Computing empirical permutation entropy by the old method

Figures/oldPE.m

```

% oldPE.m - algorithm for the calculation of an empirical
% permutation entropy in maximally overlapping sliding windows
% the function is realized by the method proposed in [KSE07]

% INPUT (x - the considered time series, Tau - a delay,
% d - an order of the ordinal patterns, WS - size of a sliding window)
% OUTPUT [ ePE - the values of empirical permutation entropy]

function ePE = oldPE(x, Tau, d, WS)
Length = numel(x);           % length of the time series
d1 = d+1;
dTau = d*Tau;
nPat = factorial(d1);        % amount of ordinal patterns of order d
opd = zeros(1, nPat);       % distribution of ordinal patterns
ePE = zeros(1, Length);     % empirical permutation entropy
op = zeros(Tau, d);         % ordinal pattern  $(i_1, i_2, \dots, i_d)$ 
opW = zeros(1, WS);         % ordinal patterns in the window
ancNum = nPat./factorial(2:d1); % ancillary numbers
for iTau = 1:Tau            % loop for the first window
    cnt = iTau;
    op(iTau, 1) = (x(dTau+iTau-Tau) >= x(dTau+iTau));
    for k = 2:d
        op(iTau, k) = sum(x((d-k)*Tau+iTau)>=x((d1-k)*Tau+iTau:Tau:dTau+iTau));
    end
    opW(cnt) = sum(op(iTau, :).*ancNum)+1; % the first ordinal pattern
    opd(opW(cnt)) = opd(opW(cnt))+1;
    for t = dTau+Tau+iTau:Tau:WS+dTau    % loop for the next ord. patterns
        op(iTau, 2:d) = op(iTau, 1:d-1);
        op(iTau, 1) = (x(t-Tau) >= x(t));
    end
end

```

```

        for j = 2:d
            if(x(t-j* $\tau$ ) >= x(t))
                op(iTau, j) = op(iTau, j)+1;
            end
        end
        opNumber = sum(op(iTau, :).*ancNum)+1;
        opd(opNumber) = opd(opNumber)+1;
        cnt = cnt+ $\tau$ ;
        opW(cnt) = opNumber;           % the next ordinal pattern
    end
end
ordDistNorm = opd/WS;
ePE(WS+ $\tau$ *d) = -nansum(ordDistNorm(1:nPat).*log(ordDistNorm(1:nPat)));

iTau = mod(WS,  $\tau$ )+1;                % current shift 1: $\tau$ 
iPat = 1;                              % current pattern in the window
for t = WS+d $\tau$ +1:Length                % loop for all time-series
    op(iTau, 2:d) = op(iTau, 1:d-1);
    op(iTau, 1) = (x(t- $\tau$ ) >= x(t));
    for j = 2:d
        if(x(t-j* $\tau$ ) >= x(t))
            op(iTau, j) = op(iTau, j)+1;
        end
    end
    nNew = sum(op(iTau, :).*ancNum)+1;   % "incoming" ordinal pattern n
    nOut = opW(iPat);                    % "outcoming" ordinal pattern
    opW(iPat) = nNew;
    if nNew ~= nOut                      % update the distribution
        opd(nNew) = opd(nNew)+1;        % "incoming" ordinal pattern
        opd(nOut) = opd(nOut)-1;        % "outcoming" ordinal pattern
        ordDistNorm = opd/WS;
        ePE(t) = -nansum(ordDistNorm(1:nPat).*log(ordDistNorm(1:nPat)));
    else
        ePE(t) = ePE(t-1);
    end
    iTau = iTau+1;
    iPat = iPat+1;
    if(iTau >  $\tau$ ) iTau = 1; end
    if(iPat > WS) iPat = 1; end
end
ePE = ePE(WS+ $\tau$ *d:end);

```

A.3 Computing empirical permutation entropy for ordinal patterns with tied ranks

Figures/PEeq.m

```

% PEEq.m - algorithm for the fast calculation of an empirical
% permutation entropy for the case of ordinal patterns with tied ranks
% in maximally overlapping sliding windows
% the function is realized by the method proposed in [UK13]

% INPUT (x - the considered time series, Tau - a delay,
% d - an order of the ordinal patterns with tied ranks,
% WS - size of a sliding window)
% OUTPUT [ ePE - the values of empirical permutation entropy]

function ePE = PEEq(x, Tau, d, WS)
load(['tableEq' num2str(d) '.mat']); % the precomputed table
opTbl = eval(['tableEq' num2str(d)]); % of successive ordinal patterns
L = numel(x);                        % length of time series
dTau = d* $\tau$ ;
nPat = 1;
for i = 3:2:2*d+1
    nPat = nPat*i;
end
end

```

```

opd = zeros(1, nPat);           % distribution of the modified ordinal patterns
ePE = zeros(1, L);             % empirical permutation entropy
b = zeros(Tau, d);             % indicator of equality (b1, b2, ..., bd)
prevOP = zeros(1, Tau);        % previous modified ordinal patterns for 1:τ
opW = zeros(1, WS);            % modified ordinal patterns in the window
ancNum = ones(1, d);           % ancillary numbers
for j = 2:d
    ancNum(j) = ancNum(j-1)*(2*j-1);
end
peTbl(1:WS) = -(1:WS).*log(1:WS); % table of values g(j)
peTbl(2:WS) = (peTbl(2:WS)-peTbl(1:WS-1))./WS;
for iTau = 1:Tau               % all shifts
    cnt = iTau;
    mOP = zeros(1, d);
    t = dTau+iTau;             % current time t of the last point in mOP
    for j = 1:d                 % determining modified ordinal patterns
        for i = j-1:-1:0
            if(i == 0 || b(iTau, i) == 0)
                if(x(t-j*Tau) > x(t-i*Tau))
                    mOP(j) = mOP(j)+2;
                elseif(x(t-j*Tau) == x(t-i*Tau))
                    b(iTau, j) = 1;
                end
            end
        end
    end
    mOP(1:d) = mOP(1:d)+b(iTau, 1:d); % add equality indicator
    opW(cnt) = sum(mOP.*ancNum);
    opd(opW(cnt)+1) = opd(opW(cnt)+1)+1;
    cnt = cnt+Tau;
    for t = iTau+Tau*(d+1):Tau:WS+Tau*d % loop for the first window
        b(iTau, 2:d) = b(iTau, 1:d-1); % renew (b1, b2, ..., bd)
        b(iTau, 1) = 0;
        posL = 1;                % position L of the next point
        eqFlag = 0;              % indicator of equality B
        for i = 1:d;              % determining the position L
            if(b(iTau, i) == 0)
                if(x(t-i*Tau) > x(t))
                    posL = posL+2;
                elseif(x(t) == x(t-i*Tau))
                    eqFlag = 1;
                    b(iTau, i) = 1;
                end
            end
        end
        posL = posL+eqFlag; % position L of the next point
        opW(cnt) = opTbl(opW(cnt-Tau)*(2*d+1)+posL);
        opd(opW(cnt)+1) = opd(opW(cnt)+1)+1;
        cnt = cnt+Tau;
    end
    prevOP(iTau) = opW(t-dTau);
end
OPDnorm = opd/WS;               % normalization of the ordinal distribution
ePE(WS+Tau*d) = -nansum(OPDnorm(1:nPat).*log(OPDnorm(1:nPat)));

iTau = mod(WS, Tau)+1;         % current shift 1:τ
iOP = 1;                        % position of the current pattern in the window
for t = WS+Tau*d+1:L           % loop for all points in a time series
    b(iTau, 2:d) = b(iTau, 1:d-1);
    b(iTau, 1) = 0;
    posL = 1;
    eqFlag = 0;                 % x(j)==x(i)?
    for i = 1:d;                 % determining the position L
        if(b(iTau, i) == 0)
            if(x(t-i*Tau) > x(t))
                posL = posL+2;
            elseif(x(t) == x(t-i*Tau))
                eqFlag = 1;
                b(iTau, i) = 1;
            end
        end
    end
end

```

```

    end
end
posL = posL+eqFlag; % position L of the next point
nNew = opTbl(prevOP(iTau)*(2*d+1)+posL); % "incoming" ordinal pattern
nOut = opW(iOP); % "outcoming" ordinal pattern
prevOP(iTau) = nNew;
opW(iOP) = nNew;
nNew = nNew+1;
nOut = nOut+1;
if nNew ~= nOut % if nNew == nOut, ePE does not change
    opd(nNew) = opd(nNew)+1; % "incoming" ordinal pattern
    opd(nOut) = opd(nOut)-1; % "outcoming" ordinal pattern
    ePE(t) = ePE(t-1)+peTbl(opd(nNew))-peTbl(opd(nOut)+1);
else
    ePE(t) = ePE(t-1);
end
iTau = iTau+1;
iOP = iOP+1;
if(iTau > Tau) iTau = 1; end
if( iOP > WS) iOP = 1; end
end
ePE = ePE(WS+Tau*d:end);

```

A.4 Computing empirical conditional entropy of ordinal patterns

Figures/CondEn.m

```

% CondEn.m - algorithm for the fast calculation of an empirical
% conditional entropy of ordinal patterns in sliding windows

% INPUT (x - the considered time series, Tau - a delay,
% d - an order of the ordinal patterns, WS - size of a sliding window)
% OUTPUT [ eCE - the empirical conditional entropy of ordinal patterns]

function eCE = CondEn(x, Tau, d, WS)

load(['table' num2str(d) '.mat']); % the precomputed table
Length = max(size(x)); % the length of the time series
d1 = d+1; % for fast computation
dTau = d*Tau;
dTau1 = d1*Tau;
nPat = factorial(d1); % the number of ordinal patterns of order d
ordDist = zeros(1, nPat); % the distribution of the ordinal patterns
wordDist = zeros(1, nPat*d1); % the distribution of (2,d)-words

op = zeros(1, d); % ordinal pattern (i1, i2, ..., id)
prevOP = zeros(1, Tau); % previous ordinal patterns for 1:tau
prevWord = zeros(1, Tau); % previous (2,d)-words for 1:tau
opWin = zeros(1, WS); % ordinal patterns in the window
wordWin = zeros(1, WS); % (2,d)-words in the window
peTbl(1:WS) = -(1:WS).*log(1:WS); % table of values g(j)
peTbl(2:WS) = (peTbl(2:WS)-peTbl(1:WS-1))./WS;

ancNum = nPat./factorial(2:d1); % the ancillary numbers
patTbl = eval(['table' num2str(d)]);

for iTau = 1:Tau
    cnt = iTau;
    op(1) = (x(dTau+iTau-Tau) >= x(dTau+iTau));
    for j = 2:d
        op(j) = sum(x((d-j)*Tau+iTau) >= x((d1-j)*Tau+iTau:Tau:dTau+iTau));
    end
    opNum = sum(op.*ancNum); % the first ordinal pattern
    % the ordinal distribution for the window

```

```

for j = dTau1+iTau:Tau:WS+(d+1)*Tau
    word2 = opNum*d1;
    for l = j-dTau:Tau:j-1
        if (x(l) >= x(j))
            word2 = word2+1;
        end;
    end;
    opNum = patTbl(word2+1);
    opWin(cnt) = opNum;
    ordDist(opNum+1) = ordDist(opNum+1)+1;
    wordWin(cnt) = word2;
    wordDist(word2+1) = wordDist(word2+1)+1;
    cnt = cnt+Tau;
end
prevOP(iTau) = opWin(cnt-Tau);
prevWord(iTau) = wordWin(cnt-Tau);
end
ordDistNorm = ordDist/WS;
wordDistNorm = wordDist/WS;
ePE = - nansum(ordDistNorm.*log(ordDistNorm));
eCE(WS+Tau*(d+1)) = -ePE- nansum(wordDistNorm.*log(wordDistNorm));

iTau = mod(WS, Tau)+1;           % current shift l:τ
iPat = 1;                       % position of the current pattern in the window
for t = WS+Tau*(d+1)+1:Length % loop over all points
    posL = 0;                    % the position l of the next point
    for j = t-dTau:Tau:t-Tau
        if(x(j) >= x(t))
            posL = posL+1;
        end
    end
    nNew1 = prevOP(iTau)*d1+posL;% "incoming" (2,d)-word
    nOut1 = wordWin(iPat);      % "outcoming" (2,d)-word
    prevWord(iTau) = nNew1;
    wordWin(iPat) = nNew1;

    nNew = patTbl(nNew1+1);     % "incoming" ordinal pattern
    nOut = opWin(iPat);         % "outcoming" ordinal pattern
    prevOP(iTau) = nNew;
    opWin(iPat) = nNew;

    nNew1 = nNew1+1; nOut1 = nOut1+1;
    nNew = nNew+1; nOut = nOut+1;
    % update the distribution of (2,d)-words
    if nNew1 ~= nOut1
        ordDist(nNew) = ordDist(nNew)+1; % "incoming" ordinal pattern
        ordDist(nOut) = ordDist(nOut)-1; % "outcoming" ordinal pattern
        wordDist(nNew1) = wordDist(nNew1)+1; % "incoming" (2,d)-word
        wordDist(nOut1) = wordDist(nOut1)-1; % "outcoming" (2,d)-word
        eCE(t) = eCE(t-1)-peTbl(ordDist(nNew))+peTbl(ordDist(nOut)+1)+...
                peTbl(wordDist(nNew1))-peTbl(wordDist(nOut1)+1);
    else
        eCE(t) = eCE(t-1);
    end
end
iTau = iTau+1;
iPat = iPat+1;
if(iTau > Tau) iTau = 1; end
if(iPat > WS) iPat = 1; end
end
eCE = eCE(WS+Tau*(d+1):end);

```

A.5 Computing robust empirical permutation entropy

Figures/rePE.m

```

% rePE.m - algorithm for the fast calculation of a robust empirical
% permutaion entropy in maximally overlapping sliding windows

% INPUT (x - the considered time series, Tau - a delay,
% d - an order of the ordinal patterns, WS - size of a sliding window,
% thr1 and thr2 - the lower and upper thresholds)
% OUTPUT [ re_PE - the values of robust empirical permutation entropy,
% MD - the values of MD]

function [MD, re_PE] = rePE(x, Tau, d, WS, thr1, thr2)
load(['table' num2str(d) '.mat']); % the precomputed table
pTbl = eval(['table' num2str(d)]);
Length = numel(x); % length of the time series
d1 = d+1;
dTau = d*Tau;
nPat = factorial(d1); % amount of ordinal patterns of order d
opd = zeros(1, nPat); % distribution of ordinal patterns
op = zeros(1, d); % ordinal pattern (i1, i2, ..., id)
ancNum = nPat./factorial(2:d1); % ancillary numbers
MDthr = (d+1)*d/8;
prevOP = zeros(1, Tau); % previous ordinal patterns for 1:tau
opW = zeros(1, WS); % ordinal patterns in the window
re_PE = zeros(1, Length- WS-dTau);

MD = zeros(1, Length);
for iTau = 1:Tau
    MDar1 = zeros(1, d);
    MDar2 = zeros(1, d);
    for i = 1:d
        MDar1(i)=sum(abs(x(iTau+(i-1)*Tau)-x(iTau+i*Tau:Tau:iTau+dTau))<thr1);
        MDar2(i)=sum(abs(x(iTau+(i-1)*Tau)-x(iTau+i*Tau:Tau:iTau+dTau))>thr2);
    end
    MD(iTau) = sum(MDar1)+sum(MDar2);
    MDar1(1:d-1) = MDar1(2:d);
    MDar2(1:d-1) = MDar2(2:d);
    MDar1(d) = 0;
    MDar2(d) = 0;
    for i = iTau+Tau:Tau:Length-dTau-Tau
        for j =0:d-1
            MDar1(j+1) = MDar1(j+1) + (abs( x(i+j*Tau)-x(i+dTau) ) < thr1);
            MDar2(j+1) = MDar2(j+1) + (abs( x(i+j*Tau)-x(i+dTau) ) > thr2);
        end
        MD(i) = sum(MDar1)+sum(MDar2);
        MDar1(1:d-1) = MDar1(2:d);
        MDar1(d) = 0;
        MDar2(1:d-1) = MDar2(2:d);
        MDar2(d) = 0;
    end
end
end

for iTau = 1:Tau % the first sliding window
    cnt = iTau;
    op(1) = (x(dTau+iTau-Tau) >= x(dTau+iTau));
    for j = 2:d
        op(j) = sum(x((d-j)*Tau+iTau) >= x((d1-j)*Tau+iTau:Tau:dTau+iTau));
    end
    opW(cnt) = sum(op.*ancNum); % the first ordinal pattern
    OPnumber = opW(cnt);
    if(MD(cnt)<MDthr)
        opd(OPnumber+1) = opd(OPnumber+1)+1;
    end
end
for j = dTau+Tau+iTau:Tau:WS+dTau % loop for the first window
    cnt = cnt+Tau;
    posL = 1; % the position l of the next point
    for i = j-dTau:Tau:j-Tau

```

```

        if(x(i) >= x(j))
            posL = posL+1;
        end
    end
    opW(cnt) = pTbl(opW(cnt-Tau)*d1+posL);
    OPnumber = opW(cnt);
    if(MD(cnt)<MDthr)
        opd(OPnumber+1) = opd(OPnumber+1)+1;
    end
    end
    prevOP(iTau) = opW(cnt);
end
ordDistNorm = opd/sum(opd);
re_PE(WS+Tau*d) = -nansum(ordDistNorm(1:nPat).*log(ordDistNorm(1:nPat)))/d;

iTau = mod(WS, Tau)+1;           % current shift  $l:\tau$ 
iPat = 1;                         % position of the current pattern in the window
for t = WS+Tau*d+1:Length         % loop over all points
    posL = 1;                       % the position  $l$  of the next point
    for j = t-dTau:Tau:t-Tau
        if(x(j) >= x(t))
            posL = posL+1;
        end
    end
    nNew = pTbl(prevOP(iTau)*d1+posL); % "incoming" ordinal pattern
    nOut = opW(iPat);                 % "outcoming" ordinal pattern
    prevOP(iTau) = nNew;
    opW(iPat) = nNew;
    nNew = nNew+1;
    nOut = nOut+1;
    % update the distribution of ordinal patterns
    if (MD(t-dTau)<MDthr)
        opd(nNew) = opd(nNew)+1; % "incoming" ordinal pattern
    end
    if (MD(t-WS-dTau)<MDthr)
        opd(nOut) = opd(nOut)-1; % "outcoming" ordinal pattern
    end
    ordDistNorm = opd/sum(opd);
    re_PE(t) = -nansum(ordDistNorm(1:nPat).*log(ordDistNorm(1:nPat)))/d;

    iTau = iTau+1;
    iPat = iPat+1;
    if(iTau > Tau) iTau = 1; end
    if(iPat > WS) iPat = 1; end
end
re_PE = re_PE(WS+Tau*d:end);

```

A.6 Computing empirical permutation entropy of long-term EEG recording

Figures/OnePatient795.m

```

% empirical permutation entropy is computed for the EEG recording 795 from
% The Euroipean Epilepsy Database [Epi14] in maximally overlapping sliding
% windows
close all; clear all;
[ALLEEG, EEG, CURRENTSET, ALLCOM] = eeglab;
AVR_EPE = zeros(1, 270*921600/256);
AVR_REPE = zeros(1, 270*921600/256);
sPlot = zeros(1, 270*921600/256);
DataPath = '/home/unakafov/EEG_analysis/Data/Epileptic_Freiburg_Data/
rec_79500102';
Cnt = 1; Cnt1 = 1;
Channel = 5; d = 4; FREQ = 256; Tau1 = 4; Tau2 = 4;
WS = 1024; SeizureMark = 2; Thr1 = 0; Thr2 = 4;
Filter = 1;% Filter = 1 "ON"; Filter = 0 "OFF";

```

```

ErValue = 1.5;

tic;
for i = 0:9
    disp(['file ' num2str(i) ' from 102']);

    if(i == 2)
        sPlot(Cnt+round(2*285346/WS):Cnt+round(2*307916/WS)) = SeizureMark;
    end

    EEG = pop_loadbv(DataPath, ['79502102_000' num2str(i) '.vhdr'], [],Channel);
    NewData = EEG.data;

    if( Filter == 1)
        % high-pass filtering
        [ S_numer, S_denom ] = butter( 3, 2 / FREQ, 'high' );
        NewData = filter( S_numer, S_denom, NewData);
        disp('Filtering 1 has finished');

        % low-pass filtering
        [ S_numer, S_denom ] = butter( 3, 42 / FREQ );
        NewData = filter( S_numer, S_denom, NewData);
        disp('Filtering 2 has finished');
    end

    ePE = PE(NewData, Tau1, d, WS);
    [MD, rePE1] = rePE(NewData, Tau2, d, WS, Thr1, Thr2);
    for j = 1:WS:length(ePE)-WS
        if(numel(NewData(abs(NewData(j:j+WS))<0.04))<10)
            AVR_EPE(1, Cnt) = min(ePE(1, j:j+WS));
            AVR_EPE(1, Cnt+1) = max(ePE(1, j:j+WS));
            AVR_REPE(Cnt) = min(rePE1(1, j:j+WS));
            AVR_REPE(Cnt+1) = max(rePE1(1, j:j+WS));
        else
            AVR_EPE(1, Cnt) = ErValue;
            AVR_EPE(1, Cnt+1) = ErValue;
            AVR_REPE(Cnt) = ErValue;
            AVR_REPE(Cnt+1) = ErValue;
        end
        Cnt = Cnt+2;
    end
end
disp('Time of processing 10 files');
toc;

tic;
for i = 10:99
    disp(['file ' num2str(i) ' from 102']);
    if(i == 27)
        sPlot(Cnt+round(2*463602/WS):Cnt+round(2*478818/WS)) = SeizureMark;
    end
    if(i == 49)
        sPlot(Cnt+round(2*549346/WS):Cnt+round(2*566246/WS)) = SeizureMark;
    end
    if(i == 54)
        sPlot(Cnt+round(2*14483/WS):Cnt+round(2*30618/WS)) = SeizureMark;
    end
    if(i == 74)
        sPlot(Cnt+round(2*726283/WS):Cnt+round(2*743531/WS)) = SeizureMark;
    end
    if(i == 79)
        sPlot(Cnt+round(2*868568/WS):Cnt+round(2*885189/WS)) = SeizureMark;
    end
    if(i == 89)
        sPlot(Cnt+round(2*648969/WS):Cnt+round(2*668370/WS)) = SeizureMark;
    end
    if(i == 99)
        sPlot(Cnt+round(2*491476/WS):Cnt+round(2*510564/WS)) = SeizureMark;
    end
end

```

```

EEG = pop_loadbv(DataPath, ['79502102_00' num2str(i) '.vhdr'], [], Channel);
NewData = EEG.data;

if( Filter == 1)
    % high-pass filtering
    [ S_numer, S_denom ] = butter( 3, 2 / FREQ, 'high' );
    NewData = filter( S_numer, S_denom, NewData);
    disp('Filtering 1 has finished');

    % low-pass filtering
    [ S_numer, S_denom ] = butter( 3, 42 / FREQ );
    NewData = filter( S_numer, S_denom, NewData);
    disp('Filtering 2 has finished');
end

ePE = PE(NewData, Tau1, d, WS);
[MD, rePE1] = rePE(NewData, Tau2, d, WS, Thr1, Thr2);
for j = 1:WS:length(ePE)-WS
    if(numel(NewData(abs(NewData(j:j+WS))<0.04))<10)
        AVR_EPE(1, Cnt) = min(ePE(1, j:j+WS));
        AVR_EPE(1, Cnt+1) = max(ePE(1, j:j+WS));
        AVR_REPE(Cnt) = min(rePE1(1, j:j+WS));
        AVR_REPE(Cnt+1) = max(rePE1(1, j:j+WS));
    else
        AVR_EPE(1, Cnt) = ErValue;
        AVR_EPE(1, Cnt+1) = ErValue;
        AVR_REPE(Cnt) = ErValue;
        AVR_REPE(Cnt+1) = ErValue;
    end
    Cnt = Cnt+2;
end
end
disp('Time of processing 89 files')
toc;

tic;
for i = 100:102
    disp(['file ' num2str(i) ' from 102']);
    if(i == 101)
        sPlot(Cnt+round(2*211787/WS):Cnt+round(2*267313/WS)) = SeizureMark;
    end

    EEG = pop_loadbv(DataPath, ['79502102_0' num2str(i) '.vhdr'], [], Channel);
    NewData = EEG.data;

    if( Filter == 1)
        % high-pass filtering
        [ S_numer, S_denom ] = butter( 3, 2 / FREQ, 'high' );
        NewData = filter( S_numer, S_denom, NewData);
        disp('Filtering 1 has finished');

        % low-pass filtering
        [ S_numer, S_denom ] = butter( 3, 42 / FREQ );
        NewData = filter( S_numer, S_denom, NewData);
        disp('Filtering 2 has finished');
    end

    ePE = PE(NewData, Tau1, d, WS);
    [MD, rePE1] = rePE(NewData, Tau1, d, WS, Thr1, Thr2);
    for j = 1:WS:length(ePE)-WS
        if(numel(NewData(abs(NewData(j:j+WS))<0.04))<10)
            AVR_EPE(1, Cnt) = min(ePE(1, j:j+WS));
            AVR_EPE(1, Cnt+1) = max(ePE(1, j:j+WS));
            AVR_REPE(Cnt) = min(rePE1(1, j:j+WS));
            AVR_REPE(Cnt+1) = max(rePE1(1, j:j+WS));
        else
            AVR_EPE(1, Cnt) = ErValue;
            AVR_EPE(1, Cnt+1) = ErValue;
            AVR_REPE(Cnt) = ErValue;
            AVR_REPE(Cnt+1) = ErValue;
        end
    end
end

```

```

        end
        Cnt = Cnt+2;
    end
end
disp('Time of processing 3 files')
toc;

AVR_EPE = AVR_EPE(1, 1:Cnt-2);
AVR_REPE = AVR_REPE(1:Cnt-2);
sPlot = sPlot(1:Cnt-2);
FS = 10; Gray = [0 0.7 0]; mSize = 2;
TimeAxis(1:Cnt-2) = (1:Cnt-2)*(WS/256)/3600;

aThreshold1 = 0.6;
aThreshold2 = 0.2;
a1 = zeros(1, length(AVR_EPE))+4;
a1((AVR_EPE)<aThreshold1) = AVR_EPE(AVR_EPE<aThreshold1);
a2 = zeros(1, length(AVR_REPE))+4;
a2((AVR_REPE)<aThreshold2) = AVR_REPE((AVR_REPE)<aThreshold2);

figure;
subplot(2, 1, 1);
plot(TimeAxis(1:3:end), AVR_EPE(1,1:3:end), 'k.', 'MarkerSize', mSize); hold on;
plot(TimeAxis, sPlot(1:Cnt-2)-0.4, 'color', Gray, 'LineWidth', 1); hold on;
plot(TimeAxis, a1, 'rx', 'LineWidth', 1, 'MarkerSize', mSize*3); hold on;
axis([TimeAxis(1) TimeAxis(end) 0 1.2]);
XTickArray = TimeAxis(1:18000:end);
XLabelArray= {'15:13','01:13','11:13','21:13','07:13','17:13','03:13','13:13','
23:13','09:13'};
set(gca,'XTick', XTickArray,'XTickLabel',XLabelArray,'fontsize',FS);
title(['EMPIRICAL PERMUTATION ENTROPY ePE(' num2str(d) ',' num2str(Tau1) ','
num2str(WS) ')'], 'fontsize', FS);
xlabel('Time (h:min)', 'fontsize', FS);
ylabel('Entropy (nats)', 'fontsize', FS);

subplot(2, 1, 2);
plot(TimeAxis(1:3:end), AVR_REPE(1, 1:3:end), 'k.', 'MarkerSize', mSize); hold on;
plot(TimeAxis, sPlot(1:Cnt-2)-0.4, 'color', Gray, 'LineWidth', 1); hold on;
axis([TimeAxis(1) TimeAxis(end) 0 1.2]);
plot(TimeAxis, a2, 'rx', 'LineWidth', 1, 'MarkerSize', mSize*3); hold on;
set(gca, 'XTick', XTickArray, 'XTickLabel', XLabelArray, 'fontsize', FS);
title(['robust EMPIRICAL PERMUTATION ENTROPY rePE(' num2str(d) ',' num2str(Tau2)
',' num2str(WS) ',' num2str(Thrd1) ',' num2str(Thrd2) ')'], 'fontsize', FS);
xlabel('Time (h:min)', 'fontsize', FS);
ylabel('Entropy (nats)', 'fontsize', FS);

set( gcf, 'PaperUnits', 'centimeters' );
set( gcf, 'PaperPosition', [ 0 0 21 21/2 ] );
print( '-depsc', '-r200', 'EEG795.eps' );

```


Bibliography

- [AFK⁺05] U.R. Acharya, O. Faust, N. Kannathal, T. Chua, S. Laxminarayan. Non-linear analysis of EEG signals at various sleep stages. *Computer methods and programs in biomedicine*, 80(1):37–45, 2005.
- [AHE⁺06] D. Abásolo, R. Hornero, P. Espino, D. Alvarez, J. Poza. Entropy analysis of the EEG background activity in Alzheimer’s disease patients. *Physiological measurement*, 27(3):241, 2006.
- [AJK⁺06] U.R. Acharya, K.P. Joseph, N. Kannathal, C.M. Lim, J.S. Suri. Heart rate variability: a review. *Medical and Biological Engineering and Computing*, 44(12):1031–1051, 2006.
- [AK13] J.M. Amigó, K. Keller. Permutation entropy: One concept, two approaches. *The European Physical Journal Special Topics*, 222(2):263–273, 2013.
- [AKK05] J.M. Amigó, M.B. Kennel, L. Kocarev. The permutation entropy rate equals the metric entropy rate for ergodic information sources and ergodic dynamical systems. *Physica D: Nonlinear Phenomena*, 210(1):77–95, 2005.
- [AKM14] A. Antoniouk, K. Keller, S. Maksymenko. Kolmogorov-Sinai entropy via separation properties of order-generated σ -algebras. *Discrete and Continuous Dynamical Systems*, 34(5):1793–1809, 2014.
- [ALM⁺01] R.G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, C.E. Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6):061907, 2001.
- [Ami10] J.M. Amigó. *Permutation Complexity In Dynamical Systems: Ordinal Patterns, Permutation Entropy and All That*. Springer, Berlin Heidelberg, 2010.
- [Ami12] J.M. Amigó. The equality of Kolmogorov-Sinai entropy and metric permutation entropy generalized. *Physica D: Nonlinear Phenomena*, 241(7):789–793, 2012.

- [AZS08] J.M. Amigó, S. Zambrano, M.A.F. Sanjuán. Combinatorial detection of determinism in noisy time series. *EPL (Europhysics Letters)*, 83(6):60005, 2008.
- [Ban14] C. Bandt. Autocorrelation type functions for big and dirty data series. *e-print – Arxiv preprint 1411.3904*, 2014.
- [BD00] B.E. Brodsky, B.S. Darkhovsky. *Non-parametric statistical diagnosis. Problems and methods*. Dordrecht: Kluwer Academic Publishers, 2000.
- [BK83] M. Brin, A. Katok. On local entropy. In *Geometric dynamics*. Springer, Berlin Heidelberg, 1983.
- [BKP02] C. Bandt, G. Keller, B. Pompe. Entropy of interval maps via permutations. *Nonlinearity*, 15(5):1595–1602, 2002.
- [BMC⁺05] N. Burioka, M. Miyata, G. Cornélissen, F. Halberg, T. Takeshima, D.T. Kaplan, H. Suyama, M. Endo, Y. Maegaki, T. Nomura, et al. Approximate entropy in the electroencephalogram during wake and sleep. *Clinical EEG and neuroscience*, 36(1):21–24, 2005.
- [Bon14] The Bonn EEG Database. Available online: <http://epileptologie-bonn.de>, (accessed on 16 December 2014).
- [Bor98] S.A. Borovkova. *Estimation and prediction for nonlinear time series*. PhD thesis, University Groningen, 1998.
- [BP02] C. Bandt, B. Pompe. Permutation entropy: A natural complexity measure for time series. *Phys. Rev. Lett.*, 88(174102), 2002.
- [BQMS12] C. Bian, C. Qin, Q.D.Y. Ma, Q. Shen. Modified permutation-entropy analysis of heartbeat dynamics. *Physical Review E*, 85(2):021906, 2012.
- [BRH00] J. Bruhn, H. Röpcke, A. Hoeft. Approximate entropy as an electroencephalographic measure of anesthetic drug effect during desflurane anesthesia. *Anesthesiology*, 92(3):715–726, 2000.
- [BRR⁺00] J. Bruhn, H. Röpcke, B. Rehberg, T. Bouillon, A. Hoeft. Electroencephalogram approximate entropy correctly classifies the occurrence of burst suppression pattern as increasing anesthetic drug effect. *Anesthesiology*, 93:981–985, 2000.
- [BT11] H.H.W. Broer, F. Takens. *Dynamical systems and chaos*, volume 172. Springer, 2011.

- [CGP05] M. Costa, A.L. Goldberger, C.K. Peng. Multiscale entropy analysis of biological signals. *Phys. Rev. E*, 71(2):021906, 2005.
- [Cho00] H.C. Choe. *Computational ergodic theory*. Springer, Berlin Heidelberg, 2000.
- [CTG⁺04] Y. Cao, W.-W. Tung, J.B. Gao, V.A. Protopopescu, L.M. Hively. Detecting dynamical changes in time series using the permutation entropy. *Phys. Rev. E*, 70:046217, 2004.
- [CZYW09] W. Chen, J. Zhuang, W. Yu, Z. Wang. Measuring complexity using FuzzyEn, ApEn, and SampEn. *Medical Engineering & Physics*, 31(1):61–68, 2009.
- [DK86] M. Denker, G. Keller. Rigorous statistical procedures for data from dynamical systems. *Journal of Statistical Physics*, 44(1-2):67–93, 1986.
- [Dow11] T. Downarowicz. Entropy in dynamical systems. *AMC*, 10:12, 2011.
- [ELW11] M. Einsiedler, E. Lindenstrauss, T. Ward. Entropy in Dynamics (unpublished). <http://maths.dur.ac.uk/~tpcc68/entropy/welcome.html>, 2011.
- [Epi14] The European Epilepsy Database. *Available online: <http://epilepsy-database.eu/>*, (accessed on 16 December 2014).
- [ER85] J.-P. Eckmann, D. Ruelle. Ergodic theory of chaos and strange attractors. *Reviews of modern physics*, 57(3):617–656, 1985.
- [FPSH06] B. Frank, B. Pompe, U. Schneider, D. Hoyer. Permutation entropy improves fetal behavioural state classification based on heart rate analysis from biomagnetic recordings in near term fetuses. *Medical and Biological Engineering and Computing*, 44(3):179–187, 2006.
- [GGK12] B. Graff, G. Graff, A. Kaczkowska. Entropy measures of heart rate variability for short ECG datasets in patients with congestive heart failure. *Acta Physica Polonica B Proceedings Supplement*, 5, 2012.
- [GLW14] C. Guo, W. Luk, S. Weston. Pipelined reconfigurable accelerator for ordinal pattern encoding. In *Application-specific Systems, Architectures and Processors (ASAP), 2014 IEEE 25th International Conference on*, pages 194–201. IEEE, 2014.
- [GP83a] P. Grassberger, I. Procaccia. Characterization of strange attractors. *Physical Review Letters*, 50(5):346, 1983.

- [GP83b] P. Grassberger, I. Procaccia. Estimation of the Kolmogorov entropy from a chaotic signal. *Phys. Rev. A*, 28(4):2591–2593, 1983.
- [GP84] P. Grassberger, I. Procaccia. Dimensions and entropies of strange attractors from a fluctuating dynamics approach. *Physica D: Nonlinear Phenomena*, 13(1):34–54, 1984.
- [HAEG09] R. Hornero, D. Abasolo, J. Escuredo, C. Gomez. Nonlinear analysis of electroencephalogram and magnetoencephalogram recordings in patients with Alzheimer’s disease. *Phil. Trans. R. Soc. A*, 367:317–336, 2009.
- [Har91] G.H. Hardy. *Divergent series*, volume 334. American Mathematical Soc., 1991.
- [HN11] T. Haruna, K. Nakajima. Permutation complexity via duality between values and orderings. *Physica D: Nonlinear Phenomena*, 240(17):1370–1377, 2011.
- [HP83] H.G.E. Hentschel, I. Procaccia. The infinite number of generalized dimensions of fractals and strange attractors. *Physica D: Nonlinear Phenomena*, 8(3):435–444, 1983.
- [HPC⁺05] D. Hoyer, B. Pompe, K.H. Chon, H. Hardraht, C. Wicher, U. Zwiener. Mutual information function assesses autonomic information flow of heart rate dynamics at different time scales. *Biomedical Engineering, IEEE Transactions on*, 52(4):584–592, 2005.
- [IFDT⁺12] M. Ihle, H. Feldwisch-Drentrup, C.A. Teixeira, A. Witon, B. Schelter, J. Timmer, A. Schulze-Bonhage. EPILEPSIAE – A European epilepsy database. *Computer methods and programs in biomedicine*, 106(3):127–138, 2012.
- [JB12] C.C. Jouny, G.K. Bergey. Characterization of early partial seizure onset: Frequency, complexity and entropy. *Clinical Neurophysiology*, 123(4):658–669, 2012.
- [JSK⁺08] D. Jordan, G. Stockmanns, E.F. Kochs, S. Pilge, G. Schneider. Electroencephalographic order pattern analysis for the separation of consciousness and unconsciousness: an analysis of approximate entropy, permutation entropy, recurrence rate, and phase coupling of order recurrence plots. *Anesthesiology*, 109(6):1014–1022, 2008.
- [KCAS05] N. Kannathal, M.L. Choo, U.R. Acharya, P.K. Sadasivan. Entropies for detection of epilepsy in EEG. *Computer methods and programs in biomedicine*, 80(3):187–194, 2005.

- [Kel12] K. Keller. Permutations and the Kolmogorov-Sinai entropy. *Discrete Contin. Dyn. Syst.*, 32(3):891–900, 2012.
- [KL03] K. Keller, H. Lauffer. Symbolic analysis of high-dimensional time series. *International Journal of Bifurcation and Chaos*, 13(09):2657–2668, 2003.
- [KLS07] K. Keller, H. Lauffer, M. Sinn. Ordinal analysis of EEG time series. *Chaos and Complexity Letters*, 2:247–258, 2007.
- [KS05] K. Keller, M. Sinn. Ordinal analysis of time series. *Physica A: Statistical Mechanics and its Applications*, 356(1):114–120, 2005.
- [KS09] K. Keller, M. Sinn. A standardized approach to the Kolmogorov-Sinai entropy. *Nonlinearity*, 22(10):2417–2422, 2009.
- [KS10] K. Keller, M. Sinn. Kolmogorov-Sinai entropy from the ordinal viewpoint. *Physica D*, 239(12):997–1000, 2010.
- [KSE07] K. Keller, M. Sinn, J. Emonds. Time series from the ordinal viewpoint. *Stoch. Dyn.*, 7(2):247–272, 2007.
- [KUU12] K. Keller, A.M. Unakafov, V.A. Unakafova. On the relation of KS entropy and permutation entropy. *Physica D*, 241(18):1477–1481, 2012.
- [KUU14] K. Keller, A.M. Unakafov, V.A. Unakafova. Ordinal patterns, Entropy, and EEG. *Entropy*, 16(12):6212–6239, 2014.
- [LE98] K. Lehnertz, C.E. Elger. Can Epileptic Seizures be Predicted? Evidence from Nonlinear Time Series Analysis of Brain Electrical Activity. *Physical Review Letters*, 80:5019–5022, 1998.
- [Lee14a] K. Lee. Fast Approximate Entropy. MATLAB Central File Exchange. Available online, (accessed on 16 December 2014).
- [Lee14b] K. Lee. Fast Sample Entropy. MATLAB Central File Exchange. <http://www.mathworks.com/matlabcentral/fileexchange/35784-sample-entropy>, (accessed on 16 December 2014).
- [Leh08] K. Lehnertz. Epilepsy and nonlinear dynamics. *Journal of biological physics*, 34(3-4):253–266, 2008.
- [Lib12] M.H. Libenson. *Practical approach to electroencephalography*. Elsevier Health Sciences, 2012.
- [LLL⁺10] D. Li, X. Li, Z. Liang, L.J. Voss, J.W. Sleight. Multiscale permutation entropy analysis of EEG recordings during sevoflurane anesthesia. *Journal of neural engineering*, 7(4):046010, 2010.

- [LOR07] X. Li, G. Ouyang, D.A. Richards. Predictability analysis of absence seizures with permutation entropy. *Epilepsy research*, 77(1):70–74, 2007.
- [LPS93] J.C. Lagarias, H.A. Porta, K.B. Stolarsky. Asymmetric tent map expansions. I. Eventually periodic points. *Journal of the London Mathematical Society*, 2(3):542–556, 1993.
- [LRGM02] D.E. Lake, J.S. Richman, M.P. Griffin, J.R. Moorman. Sample entropy analysis of neonatal heart rate variability. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 283(3):R789–R797, 2002.
- [LYLO14] J. Li, J. Yan, X. Liu, G. Ouyang. Using permutation entropy to measure the changes in EEG signals during absence seizures. *Entropy*, 16(6):3049–3061, 2014.
- [Lyu02] M. Lyubich. Almost every real quadratic map is either regular or stochastic. *Annals of Mathematics. Second Series*, 156(1):1–78, 2002.
- [MAEL07] F. Mormann, R.G. Andrzejak, C.E. Elger, K. Lehnertz. Seizure prediction: the long and winding road. *Brain*, 130(2):314–333, 2007.
- [MBAJ09] R. Monetti, W. Bunk, T. Aschenbrenner, F. Jamitzky. Characterizing synchronization in time series using information measures extracted from symbolic representations. *Physical Review E*, 79(4):046207, 2009.
- [MDSBA08] R. Meier, H. Dittrich, A. Schulze-Bonhage, A. Aertsen. Detecting epileptic seizures in long-term human EEG: a new approach to automatic online and real-time detection and classification of polymorphic seizure patterns. *Journal of Clinical Neurophysiology*, 25(3):119–131, 2008.
- [MF84] M. Mor, A.S. Fraenkel. Cayley permutations. *Discrete mathematics*, 48(1):101–112, 1984.
- [Mis03] M. Misiurewicz. Permutations and topological entropy for interval maps. *Nonlinearity*, 16(3):971–976, 2003.
- [MLLF⁺12] F.C. Morabito, D. Labate, F. La Foresta, A. Bramanti, G. Morabito, I. Palamara. Multivariate multi-scale permutation entropy for complexity analysis of Alzheimer’s disease EEG. *Entropy*, 14(7):1186–1202, 2012.
- [MN00] M. Martens, T. Nowicki. *Invariant measures for typical quadratic maps*. Paris: Astérisque, 2000.

- [MWWM99] H.R. Moser, B. Weber, H.G. Wieser, P.F. Meier. Electroencephalograms in epilepsy: analysis and seizure prediction within the framework of Lyapunov theory. *Physica D: Nonlinear Phenomena*, 130(3-4):291–305, 1999.
- [NG11] N. Nicolaou, J. Georgiou. The use of permutation entropy to characterize sleep electroencephalograms. *Clinical EEG and Neuroscience*, 42(1):24–28, 2011.
- [NG12] N. Nicolaou, J. Georgiou. Detection of epileptic electroencephalogram based on permutation entropy and support vector machines. *Expert Systems with Applications*, 39(1):202–209, 2012.
- [Oca09] H. Ocak. Automatic detection of epileptic seizures in EEG using discrete wavelet transform and approximate entropy. *Expert Systems with Applications*, 36(2):2027–2036, 2009.
- [ODRL10] G. Ouyang, C. Dang, D.A. Richards, X. Li. Ordinal pattern based similarity analysis for EEG recordings. *Clinical Neurophysiology*, 121(5):694–703, 2010.
- [OSD08] E. Olofsen, J.W. Sleight, A. Dahan. Permutation entropy of the electroencephalogram: a measure of anaesthetic drug effect. *British journal of anaesthesia*, 101(6):810–821, 2008.
- [Ouy14] G. Ouyang. Permutation Entropy. MATLAB Central File Exchange. <http://www.mathworks.com/matlabcentral/fileexchange/37289-permutation-entropy>, (accessed on 16 December 2014).
- [PBL⁺12] U. Parlitz, S. Berg, S. Luther, A. Schirdewan, J. Kurths, N. Wessel. Classifying cardiac biosignals using ordinal pattern statistics and symbolic dynamics. *Computers in Biology and Medicine*, 42(3):319–327, 2012.
- [Pes97] Ya.B. Pesin. *Dimension theory in dynamical systems: contemporary views and applications*. Chicago: University of Chicago Press, 1997.
- [Pin91] S.M. Pincus. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88(6):2297–2301, 1991.
- [Pin95] S.M. Pincus. Approximate entropy (ApEn) as a complexity measure. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 5(1):110–117, 1995.
- [Pom98] B. Pompe. Ranking and entropy estimation in nonlinear time series analysis. In *Nonlinear analysis of physiological data*, pages 67–90. Springer, 1998.

- [PR11] B. Pompe, J. Runge. Momentary information transfer as a coupling measure of time series. *Physical Review E*, 83:051122, 2011.
- [PWLL11] Y.-H. Pan, Y.-H. Wang, S.-F. Liang, K.-T. Lee. Fast computation of sample entropy and approximate entropy in biomedicine. *Computer methods and programs in biomedicine*, 104(3):382–396, 2011.
- [Rén61] A. Rényi. On measures of entropy and information. In *Proceedings of the fourth Berkeley Symposium on Mathematics, Statistics and Probability 1960.*, pages 547–561, 1961.
- [Rén70] A. Rényi. Probability theory. 1970. *North-Holland Ser. Appl. Math. Mech*, 1970.
- [RK68] A. Rechtschaffen, A. Kales. *A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects*. Washington: Public Health Service US Government Printing Office, 1968.
- [RM00] J.S. Richman, J.R. Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6):2039–2049, 2000.
- [RMW13] M. Riedl, A. Müller, N. Wessel. Practical considerations of permutation entropy. *The European Physical Journal Special Topics*, 222(2):249–262, 2013.
- [Rue78] D. Ruelle. An inequality for the entropy of differentiable maps. *Bulletin of the Brazilian Mathematical Society*, 9(1):83–87, 1978.
- [Rue90] D. Ruelle. The Claude Bernard Lecture, 1989. Deterministic chaos: the science and the fiction. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 427(1873):241–248, 1990.
- [Spr03] J.C. Sprott. *Chaos and time-series analysis*, volume 69. Oxford University Press Oxford, 2003.
- [Sta05] C.J. Stam. Nonlinear dynamical analysis of EEG and MEG: review of an emerging field. *Clinical Neurophysiology*, 116(10):2266–2301, 2005.
- [Tak81] F. Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.
- [TT09] S. Tong, N.V. Thakor. *Quantitative EEG analysis methods and clinical applications*. Artech House, 2009.

- [TTF09] A.T. Tzallas, M.G. Tsipouras, D.I. Fotiadis. Epileptic seizure detection in EEGs using time-frequency analysis. *IEEE Transactions on Information Technology in Biomedicine*, 13(5):703–710, 2009.
- [TV98] F. Takens, E. Verbitskiy. Generalized entropies: Rényi and correlation integral approach. *Nonlinearity*, 11:771–782, 1998.
- [TV02] F. Takens, E. Verbitskiy. Rényi entropies of aperiodic dynamical systems. *Israel journal of mathematics*, 127:279–302, 2002.
- [UK13] V.A. Unakafova, K. Keller. Efficiently measuring complexity on the basis of real-world data. *Entropy*, 15(10):4392–4415, 2013.
- [UK14] A.M. Unakafov, K. Keller. Conditional entropy of ordinal patterns. *Physica D*, 269:94–102, 2014.
- [Una14] V.A. Unakafova. Fast Permutation Entropy. MATLAB Central File Exchange. <http://www.mathworks.com/matlabcentral/fileexchange/44161-fast-permutation-entropy>, (accessed on 02 May 2014).
- [Una15] A.M. Unakafov. *Ordinal-patterns-based segmentation and discrimination of time series with applications to EEG data*. PhD thesis, University of Luebeck, 2015.
- [UUK13] V.A. Unakafova, A.M. Unakafov, K. Keller. An approach to comparing Kolmogorov-Sinai and permutation entropy. *The European Physical Journal ST*, 222:353–361, 2013.
- [Ver00] E.A. Verbitskiy. *Generalized entropies in dynamical systems*. PhD thesis, University of Groningen, 2000.
- [Wal00] P. Walters. *An introduction to ergodic theory*. New York, NY: Springer, 2000.
- [YHS⁺13] J.M. Yentes, N. Hunt, K.K. Schmid, J.P. Kaipust, D. McGrath, N. Stergiou. The appropriate use of approximate entropy and sample entropy with short data sets. *Annals of biomedical engineering*, 41(2):349–365, 2013.
- [YMS83] T. Yoshida, H. Mori, H. Shigematsu. Analytic study of chaos of the tent map: band structures, power spectra, and critical behaviors. *Journal of statistical physics*, 31(2):279–308, 1983.
- [You03] L.-S. Young. Entropy in dynamical systems. In A. Greven, Keller G., G. Warnecke, editors, *Entropy*. Princeton University Press, Princeton, NJ, USA, 2003.

- [You13] L.-S. Young. Mathematical theory of Lyapunov exponents. *Journal of Physics A: Mathematical and Theoretical*, 46(25):254001, 2013.
- [ZZRP12] M. Zanin, L. Zunino, O.A. Rosso, D. Papo. Permutation entropy and its main biomedical and econophysics applications: a review. *Entropy*, 14(8):1553–1577, 2012.

Curriculum Vitae

Personal details

Name:	Valentina A. Unakafova (Zavizion)
Date of Birth:	07/06/1989
E-mail:	UnakafovaValentina@yandex.ru
Nationality:	Russian
Gender	female

Educational profile:

- Diploma (equiv. Dipl. Inf) in Software Engineering (July 2011), Institute of Technology, Southern Federal University, Taganrog, Russia.
- Bachelor of Sciences in Information Science and Computer Engineering (July 2010), Institute of Technology, Southern Federal University, Taganrog, Russia.

Work experience:

- University of Lübeck, Institute of Mathematics, Lübeck, Germany, September 2011 to March 2015, **PhD student**.
- Medicom MTD Ltd, Taganrog, Russia, September 2009 to August 2011, **Laboratory assistant**.

Publications

1. J.M. Amigó, K. Keller, V.A. Unakafova, Ordinal symbolic analysis and its application to biomedical recordings. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 373(2034): 20140091, 2015.
2. K. Keller, A.M. Unakafov, V.A. Unakafova, Ordinal patterns, entropy, and EEG. *Entropy*, 16(12): 6212–6239, 2014.
3. V.A. Unakafova, K. Keller. Efficiently measuring complexity on the basis of real-world data. *Entropy*, 15(10): 4392–4415, 2013.
4. V.A. Unakafova, A.M. Unakafov, K. Keller, An approach to comparing Kolmogorov-Sinai and permutation entropy. *European Physical Journal Special Topics*, 222(2): 353–361, 2013.
5. K. Keller, A.M. Unakafov, V.A. Unakafova, On the relation of KS entropy and permutation entropy. *Physica D*, 241(18): 1477–1481, 2012.

Conference talks

1. Relationship between permutation entropy and Kolmogorov-Sinai entropy with examples from interval maps. The 10th AIMS Conference on Dynamical Systems, Differential Equations and Applications, Madrid, Spain, 2014.
2. The relationship of permutation entropy and Kolmogorov-Sinai entropy. Mathematics for Life Sciences Workshop, Kiev, Ukraine, 2012.