UNIVERSITÄT ZU LÜBECK
INSTITUTE OF INFORMATION SYSTEMS

From the Institute of Information Systems
of the University of Lübeck
Director: Prof. Dr. rer. nat. habil. Ralf Möller

Learning from Ups and Downs: Multivariate Ordinal Pattern Representations for Time Series

Dissertation
for Fulfilment of
Requirements
for the Doctoral Degree
of the University of Lübeck

from the Department of Computer Sciences and Technical Engineering

Submitted by

Marisa Mohr
from Itzehoe

Lübeck 2022

First referee:	Prof. Dr. rer. nat. habil. Ralf Möller
Second referee:	Prof. Dr. rer. nat. habil. Alexander Schnurr
Date of oral examination:	September 15, 2022
Approved for printing.	Lübeck, September 28, 2022

Abstract

The prediction quality of an artificial intelligence (AI) or machine learning (ML) model is primarily determined by the data used to train it. The data must contain relevant and non-redundant information that leads to a solution of the prediction problem. Conversely, a model cannot represent information that it does not contain. The research area of feature extraction or feature learning is concerned with finding efficient mappings from raw data to (low-dimensional) representations that capture information appropriate for the learning task. Information-theoretic entropies are promising representations through an encoding that preserves information content. Specifically, permutation entropy (PE) encodes information about the intrinsic "up and down" movements, or more precisely ordinal patterns, in a time series as a representation of the original time series. As many real-world applications deal with multivariate time series, in this work we study the problem of learning *ordinal pattern representations* for *multivariate time series*. We present three new multivariate ordinal pattern representations that allow to consider ordinal neighbourhood dependencies not only in time but also in space. We categorise the contributions of this work into three parts: In the first part, we elaborate on existing research strategies for determining multivariate permutation entropy (MPE) before we present three new MPE representations and highlight their advantages. In the second part, we investigate MPE in the context of multivariate fractional Brownian motion (mfBm), a time-dependent dynamical system that describes properties of long-range dependence and self-similarity. We show that different MPE representations uncover different information of mfBm such as information about the self-similarity parameter, also called Hurst parameter H, or cross-correlations of the spatial variables. In the third part, we apply MPE to obtain *good* representations in the context of an unsupervised as well as a supervised learning task. The experiments conducted are based on real-world data sets that do not necessarily fulfil the properties of long-range dependence or self-similarity. In the context of an unsupervised learning task, we introduce MOP_4SA, an approach to approximate symmetries between multivariate time series based on multivariate ordinal pattern encodings and spectral clustering. Understanding so-called symmetric behaviour has several advantages. For example, we use MOP_4SA to avoid model splits in dynamic probabilistic relational models and thus achieve runtime advantages in lifted inference. In the context of various supervised classification tasks, we show that the newly introduced MPE representations outperform the existing MPE representations. The experiments support that MPE representations considering interdependencies between spatial variables improve separability and discriminability and thus have higher effectiveness.

Kurzfassung

Die Vorhersagequalität eines Modells der künstlichen Intelligenz (KI) oder des maschinellen Lernens (ML) wird weitgehend von den Daten bestimmt, die zum Trainieren des Modells verwendet werden. Die Daten müssen relevante und nicht-redundante Informationen enthalten, die zur Lösung des Vorhersageproblems führen. Umgekehrt kann ein Modell keine Informationen darstellen, die es nicht hat. Das Forschungsgebiet der Merkmalsextraktion oder des Merkmalslernens befasst sich damit, effiziente Abbildungen von Rohdaten auf (niedrigdimensionale) Darstellungen zu finden, die für die Lernaufgabe geeignete Informationen erfassen. Informationstheoretische Entropien sind vielversprechende Darstellungen durch eine Kodierung, die den Informationsgehalt bewahrt. Konkret kodiert die Permutationsentropie (PE) Informationen über die intrinsischen Auf- und Abwärtsbewegungen oder genauer gesagt Ordinalmuster in einer Zeitreihe als Darstellung der ursprünglichen Zeitreihe. Da viele reale Anwendungen mit multivariaten Zeitreihen verbunden sind, untersuchen wir in dieser Arbeit das Lernen von ordinalen Musterdarstellungen im Kontext von multivariaten Zeitreihen. Wir stellen drei neue multivariate ordinale Musterdarstellungen vor, die die Berücksichtigung von ordinalen Nachbarschaftsabhängigkeiten nicht nur in der Zeit, sondern auch im Raum ermöglichen. Die Beiträge dieser Arbeit sind in drei Teile gegliedert: Im ersten Teil werden verschiedene Strategien zur Bestimmung der multivariaten Permutationsentropie (MPE) in der bestehenden Forschung erläutert, bevor wir drei neue MPE-Darstellungen vorstellen und ihre Vorteile hervorheben. Im zweiten Teil dieser Arbeit untersuchen wir MPE im Zusammenhang mit der multivariaten fraktionalen Brownschen Bewegung (mfBm), einem zeitabhängigen dynamischen System, das Eigenschaften Langzeitkorrelation und Selbstähnlichkeit beschreibt. Wir zeigen, dass verschiedene MPE-Darstellungen unterschiedliche Informationen von mfBm aufdecken, wie etwa Informationen über den Selbstähnlichkeitsparameter, auch Hurst-Parameter H genannt, oder Kreuzkorrelationen der räumlichen Variablen. Im dritten Teil wenden wir MPE an, um *gute* Repräsentationen im Kontext einer unbeaufsichtigten sowie einer überwachten Lernaufgabe zu erhalten. Die durchgeführten Experimente stützen sich auf reale Datensätze, die nicht unbedingt Daten enthalten, die einer Langzeitkorrelation oder Selbstähnlichkeit unterliegen. Im Kontext einer unüberwachten Lernaufgabe stellen wir MOP_4SA vor, ein Ansatz zur Approximation von Symmetrien zwischen multivariaten Zeitreihen, der auf multivariaten ordinalen Musterkodierungen und spektralem Clustering basiert. Das Verständnis von symmetrischem Verhalten hat mehrere Vorteile. Zum Beispiel verwenden wir MOP_4SA, um Modellsplits in dynamischen probabilistischen relationalen Modellen zu vermeiden und so Laufzeitvorteile bei der aufgehobenen Inferenz zu erzielen. Im Kontext verschiedener überwachter Klassifikationsaufgaben zeigen wir, dass die neu eingeführten

MPE-Repräsentationen die bestehenden MPE-Repräsentationen übertreffen. Die Experimente belegen, dass MPE-Darstellungen, die Interdependenzen zwischen räumlichen Variablen berücksichtigen, die Trennbarkeit und Diskriminierbarkeit verbessern und somit eine höhere Effektivität aufweisen.

Acknowledgements

Although the dissertation is supposed to be a one-to-one piece of work, the goal of this work would hardly have been achieved without the participation of the people I would like to thank. First of all, I would like to thank my supervisor Ralf Möller, whose help, enthusiasm, and constant patience have helped me succeed in all phases of this project. Further, I would like to thank Karsten Keller, who inspired me on many aspects of this work and always lent me a sympathetic ear. Many thanks to my employer, inovex GmbH, who gave me the necessary freedom to dedicate myself to this project. A special thanks to Florian, who mentored me all these years at inovex, who always gave me valuable feedback, no matter how spontaneous, and showed me how to survive in the scientific world.

These last years are not only shaped by a supervisor. Anyone who knows my passion for inspiring women and girls to pursue careers or science in STEM knows that female role models have also played a significant role for me when creating this work. Especially, I would like to thank Kristin, my maths teacher at high school, without whom I probably would never have studied maths. But my thanks also go to all the other female role models who have inspired and motivated me over the years. I hope that I can give a lot of this back in the future, so that we can show the world together, even or especially in the 21st century, that not only men can be successful in this field.

Another thank you goes to my parents, who have supported me in everything I have planned for myself. Further, I would like to thank all my friends, whose support I've also felt during these years. I especially like to remember the numerous car trips between Hamburg and Lübeck with Felix, full of funny conversations or technical discussions. Thank you, Felix, for always motivating me when I thought I could no longer get ahead. Thanks to the IFIS colleagues who always had an open ear (also for mathematical) discussions, but also for relaxed walks or cake fights and ginger shots at the bakery Junge. Finally, I'm very grateful that the shared office with Nils has turned into something much more meaningful than sharing an office. Thank you, Nils, for the many weekend mornings we started together on our couch with our laptops and a big cup of coffee, writing our dissertations. Thanks for not leaving me alone when my theory, code, or LaTeX didn't go my way again. Thank you for making sure I don't lose heart to be able to write these last lines of work today.

Marisa
Lübeck, January 2022

Contents

List of Algorithms

List of Abbreviations

AF atrial fibrillation

AI artificial intelligence

AIS automatic identification system

ANN artificial neural network

cfGn coarse-grained fractional Gaussian noise

cmfBm coarse-grained multivariate fractional Brownian motion

CNN convolutional neural network

DFT discrete Fourier transform

DPRM dynamic probabilistic relational model

ECG electrocardiogram

EEG electroencephalography

fBm fractional Brownian motion

FFT fast Fourier transform

fGn fractional Gaussian noise

FT Fourier transform

GP Gaussian process

GPR Gaussian process regression

KLD Kullback Leibler divergence

kNN k-nearest neighbour

logvar logical variable

MEG magnetoencephalography

mfBm multivariate fractional Brownian motion

ML machine learning

MLR multiple linear regression

MMSPE multivariate multi-scale permutation entropy

MOP multivariate ordinal pattern

MOPPE multivariate ordinal pattern permutation entropy

MPE multivariate permutation entropy

MPE-EUCL multivariate permutation entropy based on Euclidian distance

MPE-MANH multivariate permutation entropy based on Manhattan distance

MPE-NORM multivariate permutation entropy based on normalisation

MPE-PCA multivariate permutation entropy based on principle component analysis

MSE multi-scale entropy

MvPE multivariate permutation entropy

MSPE multi-scale permutation entropy

MTSC multivariate time series classification

MWPE multivariate weighted permutation entropy

parfactor parametric factor

PC principal component

PCA principal component analysis

PCC Pearson correlation coefficient

PE permutation entropy

PPE pooled permutation entropy

PPE-PCA pooled permutation entropy based on principal component analysis

PRV parameterized random variable

randvar random variable

RMSE root mean square error

RUL remaining useful lifetime

SAX Symbolic Aggregate approXimation

SVR support vector regression

SVM support vector machine

UEA University of East Anglia

WPE weighted permutation entropy

WMPE weighted multi-scale permutation entropy

List of Symbols

Ω	state space
ω	state
$\mathcal{A}$	σ-algebra
μ	probability distribution
$\mathcal{T}$	transition map
X	random variable
x	observation, realisation of a random variable
T	length of a time series
m	number of variables in a multivariate time series
H	Hurst parameter
σ	standard deviation
ρ	correlation coefficient
η	time-reversibility parameter
γ	autocovariance function
d	order of ordinal pattern
τ	delay between successive time points
ϵ	scaling factor
B_H	fractional Brownian motion
$\mathbf{B}_H^m$	multivariate fractional Brownian motion
R	randvar
$\mathbf{R}$	set of randvars
L	logvar
$\mathbf{L}$	set of logvars
A	PRV
$\mathcal{A}$	sequence of PRVs
ϕ	potential function
C	constraint
$g, \phi(\mathcal{A})_{\mid C}$	parfactor
G	model
$\mathbf{E}$	evidence, set of events
$\mathcal{X}$	symbolic representation
$\mathcal{W}$	similarity graph
$\mathcal{L}$	graph Laplacian matrix
$\mathcal{C}$	symmetry cluster

Chapter 1

Introduction

Time series data is part of many real-world applications concerning space bodies and planet Earth (weather records, temperature changes, wind or asteroids speed), health evolution metrics (brain and heart activities, muscle tensions, data from wearables like pulse, sleep, and stress indices), or economy and politics (stock prices, website clicks, server metrics, industrial sensor data, political indicators) – just to name a few. From a mathematical point of view, time series can be understood as a sequence of observed (real-valued) random variables generated by a time-dependent system, also called a dynamical system. Handling of time series or dynamical systems involves not only modelling of random variables but at the same time also temporal relationships, which naturally increases the complexity of the handling.

In order to model time series, e.g., to predict the future, numerous tools and possibilities exist, which have grown even further through the efforts of recent years in the field of artificial intelligence (AI) or machine learning (ML). However, the choice of models and appropriate algorithms is not the most important thing when deploying ML solutions in daily business – what matters the most is the data and their representations. This is not only true for ML but also for everyday life. We follow an intuitive example from Goodfellow *et al.* (2016) in which a person has the task of dividing 210 by 6. With the help of the long division, the solution should be easy for most people. However, if one is confronted with the same task in a different representation, such as the Roman numeral representation, it becomes considerably more difficult to solve the task, i.e., divide CCX by VI. Most people would first convert the numbers into the Arabic number representation, which allows for the familiar long division procedure. Thus, a *good* representation then simplifies a subsequent task.

More specifically, representations can be selections of raw data or (usually lower-dimensional) values derived from raw data and should be non-redundant and informative. Based on new representations, models are used and algorithms are performed, which leads to *better* predictions. This also means that the choice of appropriate representations influences the quality of a model and its predictions. For example, in classical mathematical modelling, various real-world phenomena are modelled by complex dynamical systems that represent the global behaviour of variables over time resulting from interactions. Such a dynamical system contains parameters that control, for example, long-term behaviour or periodicity of the system. When applying a dynamical system to a real problem, these parameters have to be estimated from observed realisations. For this purpose, certain representations of time series reflecting

these properties of the parameter can be used. The approach of extracting information from observed data and then using this information for modelling a general dynamical system is also called inverse problem.

As another example, we consider supervised learning algorithms from classical ML that learn functions or mappings between input and output to solve a selected task. Unlike time-independent random variables, time series data does not have the right recurrent representation that such algorithms require, so informative scalar-valued representations (or features) have to be extracted. Only then can models like k-nearest neighbour, support vector machine (SVM) or random forest be used. Similarly, the extraction of representations offers an interesting opportunity for unsupervised and semi-supervised learning. Representations of unlabelled data can discover factors that explain relationships or separate data allowing reuse in supervised learning tasks. Summarising, *good* representations "help the learner to discover and disentangle some of the underlying (and a priori unknown) factors of variation" (Bengio *et al.*, 2013).

Ideally, a representation is chosen without domain knowledge of specialists and yet contain as much information about the underlying dynamical process as possible. One approach for the automatic extraction of representations is using deep neural networks, which as a side effect learn appropriate representations of time series in their hidden layers (Franceschi *et al.*, 2019). However, deep learning is usually computationally intensive, difficult to interpret, and requires many data samples for model training, which are rarely available (Chollet, 2017). In symbolic time series analysis, a non-parametric mapping into a sequence of symbols that uncovers the underlying dynamics of the dynamical system is considered (Traversaro *et al.*, 2018). Ordinal pattern symbolisation introduced by Bandt and Pompe (2002) encodes the "up and down" movements in a time series, i.e., the total order of neighbouring time points. The distribution of this ordinal pattern symbolisation is then represented using information-theoretic entropies, specifically referred to as permutation entropy (PE). The class of all representations based on ordinal pattern symbols and entropies is denoted as ordinal pattern representations. The successful use of ordinal pattern representations in learning tasks has been demonstrated in various applications, from mammographic density estimation (Antonelli *et al.*, 2019) to fault diagnosis for rotating machinery (Li *et al.*, 2020). As many real-world challenges deal with multivariate time series, the problem this work focuses on is

finding intrinsic *ordinal pattern representations*
in the context of *multivariate time series*.

Multivariate time series have variables that are both time and space (i.e., on other variables) dependent. Representations of multivariate time series are usually based on univariate representations of the individual spatial variables. This means that a representation is extracted in time-space for each spatial variable before all derived representations are combined in a common measure. In contrast, this work aimes to derive multivariate representations that include dependencies in a multivariate time series in both time and space. This allows for co-movements of several spatial variables over time to be considered, making this kind of representation more efficient.

1.1 Related Work

In this section, we present related work in the field of time series representation based on entropy and ordinal pattern symbolisation. Bandt and Pompe (2002) are the first to combine both ideas in what is called PE, which is the fundamental basis of this work. In general, PE quantifies the complexity of the time series through the distribution of the ordinal patterns, i.e., the up and down movements, in a scalar-valued representation. While the associated PE is low for a deterministic time series, it approaches its maximum value in case of randomness or high complexity. To distinguish between randomness and complexity, Morabito *et al.* (2012) introduce multi-scale permutation entropy (MSPE), an extension of PE that captures the complexity of time series on different time scales. Another limitation of PE is the inability to distinguish between different patterns of a given motif in amplitudes. Fadlallah *et al.* (2013) introduce weighted permutation entropy (WPE), which takes into account patterns that differ in amplitudes by assigning weights to each extracted pattern. WPE and MSPE are based on PE but emphasise different aspects of the time series. Based on these three basic representations, this work focuses on

a) the extension of PE to the multivariate context.

Based on these results, we investigate

b) the behaviour of different multivariate ordinal pattern representations on multivariate fractional Brownian motion, a special stochastic process, and

c) the use of multivariate ordinal pattern representations for time series from real-world challenges to enable or simplify a subsequent learning task.

Multivariate Extensions of PE. To extend the concept of PE to multivariate time series, Keller and Lauffer (2003) introduce pooled permutation entropy (PPE) by pooling each univariate ordinal pattern of all spatial variables into a common multivariate representation. By analogy with MSPE and PPE, Morabito *et al.* (2012) present multivariate multi-scale permutation entropy (MMSPE). To fill the gap, we introduce multivariate weighted permutation entropy (MWPE) as a canonical extension based on WPE and PPE, allowing the consideration of amplitudes by including weighted ordinal patterns in multivariate time series (Mohr *et al.*, 2021b). While the previous measures examine the ups and downs in time-space, He *et al.* (2016) present multivariate permutation entropy (MvPE), which examines the ups and downs in phase-space, i.e., ups and downs between spatial variables at a fixed time point. To account for time and phase space simultaneously, we propose multivariate ordinal pattern permutation entropy (MOPPE) by extending univariate ordinal pattern to multivariate ordinal pattern (MOP). To reduce the complexity of combinatorial possibilities, Rayan *et al.* (2019) propose to first reduce the number of spatial variables in a multivariate time series to one dimension by applying distance measures before calculating univariate PE by default. To account for interdependencies or correlations

between variables in both time and space, we propose multivariate permutation entropy based on principle component analysis (MPE-PCA), using principal component analysis (PCA) as a dimensionality reduction procedure (Mohr *et al.*, 2020b).

PE and Fractional Brownian Motion. For many real-world applications, modelling of time-dependent dynamical systems requires specific properties such as *long-range dependence* or *self-similarity*, for which fractional Brownian motion (fBm) is commonly used. The fundamentals of fBm are based on sound theory and have successfully applied to many real-life challenges (Dietrich and Newsam, 1997; Arianos and Carbone, 2009; Didier and Pipiras, 2011). Lavancier *et al.* (2009) introduce multivariate fractional Brownian motion (mfBm) as a generalisation of fBm. Amblard and Coeurjolly (2011) study the covariance structure and the spectral structure of this multivariate stationary process and obtain a characterisation of mfBm through its covariance function on which some of the results of this work are based. Moreover, Bandt and Shiha (2007); Zunino *et al.* (2008); Sinn and Keller (2011); Dávalos *et al.* (2018) investigate the distribution of ordinal patterns for the univariate case, i.e., in fBm. The authors show that the self-similarity parameter, also known as Hurst parameter H, is directly related to the distribution of ordinal patterns, which nominates PE as a candidate for solving an inverse problem, among others. We take up the property of the distribution of univariate ordinal patterns and investigate the behaviour of existing and newly introduced multivariate ordinal pattern representations in the multivariate context, from which interesting applications and recommendations can be derived (Mohr *et al.*, 2020a, 2021b; Mohr and Möller, 2021b,a).

PE for Learning Tasks in Different Applications. For other real-world applications, it is of interest to detect similarities between time series, e.g., to perform classification tasks or to obtain sparse (lifted) representations using symmetries in the data, which reduce complexity and achieve good performance. Similarity detection in time series is a vast area of research, with approaches to detecting similarities in a set of time series generally are either value-based or symbol-based. By comparing the values of each point in a time series with the values of each other point in a different time series (warping), value-based approaches are able to take into account shifts and frequencies, which makes them successful in application. Kramer (2020) provides a detailed overview of popular algorithms such as dynamic time warping (DTW) introduced by Kruskal and Liberman (1983), its multivariate extension dependent multivariate dynamic time warping (DMDTW) by Petitjean *et al.* (2012) or matrix profile by Yeh *et al.* (2016). Nevertheless, the flexibility of warping in value-based approaches leads to a high computational effort. Even if, e.g., Salvador and Chan (2004) or Silva and Batista (2016) achieve a runtime improvement by limiting the warping path (FastDTW) or reducing the number of data points (PrunedDTW), these approaches are difficult to apply in applications involving large amounts of data. While value-based approaches are inefficient because they are usually performed on raw data, symbol-based approaches encode the time series observations as sequences

of symbolic abstractions that match the shape of the time series. The symbolic abstraction can be understood as (lower-dimensional) representation, based on which classical similarity or learning algorithms are applied. Chiu *et al.* (2003) introduce the probably best known and most used symbolic representation, namely Symbolic Aggregate approXimation (SAX), where the data range is partitioned, and the partitions are assigned to the symbols. While the search for the optimal partitioning is complex, the ordinal symbolisation approach by Bandt and Pompe (2002) dispenses with partitioning and uses the order between successive neighbours. The use of neighbourhood behaviour allows us to successfully apply (multivariate) ordinal patterns in different learning tasks such as classification (Mohr *et al.*, 2020b) or symmetry approximation (Finke and Mohr, 2021; Finke *et al.*, 2022).

1.2 Research Objectives and Scientific Contributions

In Section 1.1, we have presented important work in the domain of ordinal pattern representations and its applications. In general, many efforts have been made in the last decade to extract ordinal pattern representations from time series in a multivariate context and to beneficially use these representations for analysis or prediction. Compared to the research of the last decade, the aim of this work is to consider not only temporal interdependencies but also interdependencies of spatial variables, i.e., the potentially simultaneous up and down movement of two or more variables. Since the representation of a time series is related to the model used, we examine and discuss existing and especially the new representations in different contexts and selected applications.

In this work, we make several contributions to multivariate ordinal pattern representations. We summarise the contributions as follows.

1) **New approaches in multivariate ordinal pattern representations:** Since this work focuses on multivariate ordinal pattern representations, we examine existing multivariate extensions, point out their limitations and fill the gaps in existing theory. In addition, we provide new definitions that address the major drawback of existing extensions, namely the fact that existing extensions do not consider mutual dependencies in both time and space.

    a) **Review of existing multivariate extensions for ordinal pattern representations:** After PE was introduced by Bandt and Pompe (2002) and successfully applied in numerous real-world challenges, several authors have proposed multivariate extensions of PE. We summarise these and categorise them according to their approach to give an overview of the possible applications and facilitate the integration of the following contributions.

    b) **Introduction of MWPE:** As PE is multivariate extended in a canonical way by PPE, so is MSPE extended by MMSPE. We fill the gap and extend WPE by multivariate weighted permutation entropy (MWPE).

c) **Introduction of MOPPE:** While canonical multivariate extensions such as PPE, MMSPE and MWPE are based on the spatial variable-by-variable use of univariate ordinal patterns, we extend the definition of univariate ordinal patterns to multivariate ordinal patterns (MOPs). Thus, we allow for the consideration of interdependencies between variables in both time and space. Based on the definition of MOP, we introduce multivariate ordinal pattern permutation entropy (MOPPE) in analogy to univariate ordinal patterns and PE.

d) **Introduction of MPE-PCA:** Since MOPPE has some weaknesses in application due to exponential combinatorial possibilities of MOPs, we introduce multivariate permutation entropy based on principle component analysis (MPE-PCA). By reducing the dimension of the phase-space via PCA, the necessary information or interdependencies can be extracted from the phase-space and then included in the PE calculation in the time-space.

2) **Investigation of multivariate extensions of PE on mfBm:** The univariate behaviour, i.e., PE on fBm, is well known. This contribution comprises i) detailed investigations of the behaviour of existing and newly introduced multivariate measures on mfBm in the variation of its self-similarity or Hurst parameter H, and ii) a detailed comparison of all multivariate measures examined, as well as recommendations for their use. The determined relationships are particularly suitable for use in solving the inverse problem.

a) **Investigation of PPE on mfBm:** The distribution of ordinal patterns of certain length, and thus also PPE, is monotonically related to the Hurst parameter H, i.e., PPE decreases with increasing H. Thus, considering the estimation of H of mfBm as an inverse problem, the PPE of an observed multivariate time series can be used for its solution.

b) **Investigation of MMSPE on mfBm:** Since scaling does not change the structure of mfBm, we show that the behaviour of MMSPE of any scale ϵ is identical to that of PPE, and analyses with MMSPE do not provide any additional insight than with PPE.

c) **Investigation of MWPE on mfBm:** In contrast to PPE, MWPE is substantially influenced by strictly ascending and descending ordinal patterns. For this reason, MWPE decreases faster than PPE as H increases, providing more expressive representations that may promise better discriminability, e.g., in classification tasks.

d) **Investigation of MOPPE on mfBm:** Since MOPs involve the dependence of several spatial variables on mfBm, MOPPE is promising for estimating the Hurst parameter H and correlations from one source, which the other measures cannot.

e) **Investigation of MPE-PCA on mfBm:** We show that the entropies of higher orders of the principal components again are monotonically dependent

on the Hurst parameter H, i.e., entropy decreases as H increases. Thus, MPE-PCA is appropriate for solving inverse problems, i.e., given an observed realisation of mfBm, the calculation of MPE-PCA provides information about the level of H and thus parameters of the generating mfBm. Moreover, we show that, unlike PPE, MPE-PCA can uncover large cross-correlations of variables at large Hurst parameter H.

3) **Further applications of the introduced representations:** According to Goodfellow *et al.* (2016), we consider the introduced representations to be *good* if they make a subsequent learning task easier. Therefore, we use the introduced approaches in selected real-world challenges and show their efficiency in the context of machine learning.

   a) **Symmetry Approximation:** We propose MOP$_4$SA, an approach for approximating symmetries based on MOP and clustering. Exploiting symmetries in data is an important topic to obtain sparse (lifted) representations, reduce complexity, and achieve good performance in a wide variety of different challenges in ML or AI in general. In an experiment, we show that runtime advantages in lifted inference can be obtained while accuracy remains good.

   b) **Supervised Learning:** Classical ML models for regression or classification such as multiple linear regression or k-nearest neighbour can't process time series directly. It is necessary to extract *good* representations (or features) from time series before using these algorithms. In a regression task from manufacturing, we show that the use of entropy features is useful due to their low complexity and good performance compared to other prominent feature classes. In several classification tasks, we evaluate which of the multivariate ordinal pattern representations presented in this work can be used flexibly and reliably and lead to good predictions on different real-world data sets. We show that new representations introduced in this work outperform existing ones.

1.3 Structure

After this introduction, **Chapter 2** starts with three introductory examples that illustrate the potential of using ordinal pattern representations in mathematical modelling or ML in general. The examples are followed by **Chapter 3** on preliminaries, which formalises the world of dynamical systems and time series, before (multivariate) fractional Brownian motion as the special object of this work is presented. We then formalise univariate ordinal pattern symbolisation and the corresponding concepts of permutation entropy as analytical tools to study dynamical systems and time series for qualitative properties. After the preliminaries, the main part of this work begins, divided into three parts representing the contributions of this work.

In the **first part**, existing approaches are discussed, and new approaches in multivariate ordinal pattern representations are introduced as the main contribution. In **Chapter 4**, we present a review of existing multivariate extensions of ordinal pattern representations by categorising them according to their approach, while also discussing limitations (Contribution 1a). In **Chapter 5**, we introduce three new multivariate extensions of PE, namely MWPE, MOPPE and MPE-PCA (Contribution 1b-d). The contents of the first part were mainly published in the following two conference papers, the first introducing MWPE and the second MPE-PCA and MOPPE.

Marisa Mohr, Florian Wilhelm, and Ralf Möller. On the Behaviour of Weighted Permutation Entropy on Fractional Brownian Motion in the Univariate and Multivariate Setting. *The International FLAIRS Conference Proceedings*, 34, 2021

Marisa Mohr, Florian Wilhelm, Mattis Hartwig, Ralf Möller, and Karsten Keller. New Approaches in Ordinal Pattern Representations for Multivariate Time Series. In *Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference (FLAIRS-33)*, pages 124–129, 2020

In the **second part**, we investigate existing and newly introduced multivariate extensions of PE when applied to multivariate fractional Brownian motion (mfBm). To this end, in **Chapter 6** and **Chapter 7** we present new studies on the behaviour of the existing representations PPE and MMSPE, respectively (Contribution 2a and 2b). Thereafter, in **Chapter 8**, **Chapter 9** and **Chapter 10** we investigate the behaviour of the newly introduced representations MWPE, MOPPE and MPE-PCA (Contribution 2c-e). Each chapter contains a theoretical analysis, an empirical evaluation as well as a short interim conclusion. **Chapter 11** concludes with a comparison of all representations and a discussion of possible applications in the context of mfBm. The second part is based on the following four publications, the first examining the behaviour of PPE and MMSPE, the second of MWPE, the third of MPE-PCA and the fourth a summary of all studies with additional study of MOPPE.

Marisa Mohr, Nils Finke, and Ralf Möller. On the Behaviour of Permutation Entropy on Fractional Brownian Motion in a Multivariate Setting. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2020 (APSIPA-ASC)*, pages 189–196, 2020

Marisa Mohr, Florian Wilhelm, and Ralf Möller. On the Behaviour of Weighted Permutation Entropy on Fractional Brownian Motion in the Univariate and Multivariate Setting. *The International FLAIRS Conference Proceedings*, 34, 2021

Marisa Mohr and Ralf Möller. Ordering Principal Components of Multivariate Fractional Brownian Motion for Solving Inverse Problems. In *Proceedings of*

the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2021 (APSIPA-ASC), 2021

Marisa Mohr and Ralf Möller. A Summary of Canonical Multivariate Permutation Entropies on Multivariate Fractional Brownian Motion. *Advances in Science, Technology and Engineering Systems Journal*, 6(5):107–124, 2021

In the **third part**, we discuss the introduced multivariate representations in the context of real-world applications. In **Chapter 12**, we present MOP_4SA, an approach for symmetry approximation based on multivariate ordinal patterns, in a reference formalism of dynamic probabilistic relational models (DPRMs) (Contribution 3a). In **Chapter 13**, we present an empirical evaluation of multivariate permutation entropies as representations or features in a classification task (Contribution 3b). This third part was mainly published in the following three conference papers. The first and second papers deal with MOP_4SA in the univariate and multivariate cases respectively, while the third paper deals with classification.

Nils Finke and Marisa Mohr. A Priori Approximation of Symmetries in Dynamic Probabilistic Relational Models. In Stefan Edelkamp, Ralf Möller, and Elmar Rueckert (Eds.), *KI 2021: Advances in Artificial Intelligence*, pages 309–323. Springer, 2021

Nils Finke, Ralf Möller, and Marisa Mohr. Multivariate Ordinal Patterns for Symmetry Approximation in Dynamic Probabilistic Relational Models. In Guodong Long, Xinghuo Yu, and Sen Wang (Eds.), *AI 2021: Advances in Artificial Intelligence*, pages 543–555. Springer International Publishing, 2022

Marisa Mohr, Florian Wilhelm, Mattis Hartwig, Ralf Möller, and Karsten Keller. New Approaches in Ordinal Pattern Representations for Multivariate Time Series. In *Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference (FLAIRS-33)*, pages 124–129, 2020

Chapter 14 concludes with a summary of contributions and ends with broad future work for ordinal pattern representations in the context of multivariate time series.

Chapter 2

Learning from Ups and Downs by Example

This work deals with a scalar-valued representation of time series, which is based on ordinal pattern symbols that reflect the "up and down" movements within a time series. The distribution of ordinal pattern symbols is represented by using information-theoretic entropies, in particular PE, as a measure of the complexity of the time series. Before diving straight into the details of multivariate time series and multivariate ordinal pattern representations, we illustrate the potential of ordinal pattern representations by examples.

In **Section 2.1**, we motivate the use of ordinal pattern symbols and PE to solve the so-called inverse problem, i.e., retrieving causing parameters of a dynamical system that generate the observed data. We elaborate the inverse problem in the context of multivariate fractional Brownian motion and conclude in **Part II** with several theorems and experimental results that are suitable for solving the inverse problem.

In **Section 2.2**, we motivate the use of ordinal patterns to determine symmetries in time series data. Exploiting symmetries is an important topic to obtain sparse (lifted) representations, reduce complexity and achieve good performance in ML in general. The entire algorithm for learning symmetries as well as its benefits are discussed in depth in the context of dynamic probabilistic relational models in **Chapter 12**.

In **Section 2.3**, we motivate the use of ordinal pattern symbols and PE as features in a supervised learning task. Since a good prediction depends on both the choice of features and the choice of a suitable model, we compare different features as well as different supervised models for regression as well as classification in **Chapter 13**.

2.1 Using Ups and Downs for Solving Inverse Problems

In this first example, we motivate the use of univariate ordinal pattern representations for solving inverse problems. An inverse problem is a problem in which, based on an observed effect of a system, the underlying cause of the effect is determined. The opposite of an inverse problem is a direct problem (sometimes called a forward problem), where starting from an underlying cause of a system, its effect is determined. The inverse problem is thus the inverse of the forward problem, in which the model parameters leading to the recorded observed data need to be determined.

In the following, we consider complex dynamical systems for modelling real phenomena that represent the global behaviour of variables over time resulting from interactions. Some real phenomena require specific properties such as *long-term dependence* or *self-similarity*. For example, long-term memory is fundamental for financial data (Elliott and van der Hoek, 2001). If the profit series on an financial investment, i.e., the return series, exhibit long-term dependence, this indicates that returns over time are not independent, which means that past returns can help to predict future returns (Nath and Mousumi, 2012). In general, long-range dependence (also called long-term memory or long-range persistence) means that the statistical dependence between two points in time decreases less than exponentially with increasing time interval or spatial distance between the points, typically as a power decay. Note that in many publications, the terms "long-range dependence" and "self-similarity" are used interchangeably even though they are not the same. Self-similar processes, simply put processes with recurring patterns, are the simplest way to model far-reaching dependent processes, as they can be described by only one (self-similarity) parameter – the so-called Hurst parameter. The Hurst parameter is thus a factor of the dynamical system that causes the effect of long-range dependence. Here, $0 < H < 1$ applies, whereby the closer H is to 1, the higher the degree of self-similarity as well as the degree of long-range dependence. A value of 0.5 means that there is no long-range dependence. As the opposite of long-range dependence, $H < 0.5$ corresponds to antipersistence, which indicates a strong negative correlation, so that the process fluctuates violently. Pipiras and Taqqu (2017) provide further details regarding long-range dependence and self-similarity. Note that time-dependent phenomena that depend on long-range dependence or self-similarity are usually described by fractional Brownian motion (fBm), the stochastic process, which we focus on in Part II of this work. To this end, definitions and details of dynamical systems or fBm are introduced in Section 3.2.

Having discussed dynamical systems and their special properties, we come back to the inverse problem. Assuming that the observed time series data are realisations of a dynamical system like fBm, why is it of interest to solve the inverse problem? To retrieve information about the Hurst parameter H the generating fBm provides information about the cause of the self-similarity. When knowing the Hurst parameter H, the dynamical model can be used for general purposes such as simulations or predictions. For example, financial returns can be predicted and peaks exploited. The problem of retrieving information from measurement data has already been studied for a long time. Over the years, several graphical or analytical methods such as rescaled (R/S) range analysis (Li, 2003), detrended fluctuation analysis (DFA) (Setty and Sharma, 2015) or classical statistical inversion methods (D'Ambrogi-Ola, 2009) have been proposed for this purpose. More recent approaches from the field of machine learning are the application of PCA (Oezkurt and Akgül, 2005; Oezkurt *et al.*, 2006; Li, 2009), Bayesian estimation (Makarava, 2012), or wavelet filters (Hmood and Hamza, 2021) for Hurst parameter estimation.

Since the Hurst parameter reflects the self-similarity of a stochastic process, i.e., the recurrence of certain patterns with regard to scaling, it is promising to study the distribution of (ordinal) symbol patterns within the process. Indeed, in Section 6.1 we formulate a direct relation between the distribution of ordinal patterns and the Hurst parameter H in a closed formula. This property enables the Hurst parameter H to be determined via the distribution of ordinal patterns. Furthermore, the use of ordinal pattern representations to estimate the Hurst parameter promises an additional advantage. While the measurement data is usually noisy and value-based approaches have difficulty dealing with noise, shape-based approaches are robust to noise.

The use of ordinal pattern representations to solve the inverse problem in the context of fBm is elaborated in Part II of this work. In particular, we derive several theorems and experimental results that are suitable for solving the inverse problem.

2.2 Using Ups and Downs for Exploiting Symmetries

In this second example, we motivate the use of ordinal pattern representations to identify symmetries in time series data, i.e., time series encoding the same or approximately the same behaviour of an object over time. As the German mathematician Hermann Klaus Hugo Weyl (1885 – 1955) said, "symmetric means something like well-proportioned, wellbalanced, and symmetry denotes that sort of concordance of several parts by which they integrate into a whole. [...] Beauty is bound up with symmetry" (Weyl, 1952). In everyday life, we usually easily recognise when shapes exhibit symmetry. They are probably easiest to recognise in the three well-known transformations mirroring, rotation and translation. Also the fractional Brownian motion that we presented in the previous section contains symmetries, namely in the form of self-similarity. Self-similar objects look the same when enlarged or reduced. In a sense, they are composed of smaller copies of themselves. This property is often referred to as scale symmetry or scale invariance. Self-similar objects, also called fractals, are found both as theoretical mathematical constructions and in nature. The former, such as the Mandelbrot set or Koch's snowflake, usually exhibit exact self-similarity. In contrast to mathematical fractals, naturally occurring fractals are usually roughly or statistically self-similar, i.e., they look more or less the same at different scales or have a similar distribution of elements when magnified. The natural phenomenon of self-similarity can be observed in clouds, coastlines, snowfields or waterfalls, or even ferns, among other beautiful objects in life. The connection between beauty and symmetry mentioned by Hermann Weyl is still cited again and again in literature today. Why is this connection so significant in science, and what role does symmetry play in the field of time series research in particular, or in AI and ML in general?

In the age of Big Data, historical time series represent a large part of the data

stored in computers. One example is historical share prices in finance with important indicators such as opening, closing, high and low prices as well as trading volume. The ability to retrieve time series of stock prices from a database with specific shapes is a requirement that is essential in many applications. In financial analysis, for example, it is interesting to examine all closing price series for so-called head-and-shoulders patterns. A head-and-shoulders pattern consists of three peaks, with the two outer peaks close together and the middle peak being the highest. In order to retrieve time series with specific shapes, such as head-and-shoulders patterns, the shapes of interest need to be determined. Agrawal *et al.* (1993) introduces a shape definition language, also denoted as $\mathcal{SDL}$. For example, a trend can be called an upward trend even if there have been some downward transitions within time, as long as they are limited to a certain number. The shape of the curve can be described by its own alphabet, such as "rising" and "falling", i.e., encodings of the total order of two values as so-called ordinal patterns. A symbol of this alphabet describes a transition between two successive values in the time series. The general framework of $\mathcal{SDL}$ allows (i) searching for sub-sequences with a certain shape and their positions in a time series (database), (ii) searching for unknown frequently occurring sub-sequences, i.e., shapes or patterns, and their positions in a time series (database), as well as (iii) identifying the most similar time series in a time series database based on their shapes. Roughly speaking, all the tasks described involve identifying (parts of a) time series that behave the same or approximately similar over time. In financial analysis, certain recurring shapes have proven to be important indicators over time. For example, the head-and-shoulders pattern is considered one of the most reliable trend reversal patterns, predicting a trend reversal from bull to bear. While the search for similar sequences or certain patterns, e.g. head and shoulder patterns, in time series (databases) is mainly studied in the research area of knowledge discovery in databases (KDD), it can also be assigned to the research area of symmetries. Typically, important shapes emerge as a result of observations over a long period of time. The forms which are observed frequently and repeatedly and are usually due to a specific and relevant behaviour for an underlying behaviour or task. Therefore, shapes are a form of symmetrical behaviour, i.e., symmetries that are due to their recurrence over time.

Besides symmetry analysis as part of a knowledge discovery process, symmetries can also be exploited to speed up inference of any probabilistic model, e.g., when performing inference based on historical stock prices. Continuing with the example, a temporal, or dynamical, probabilistic model can be set up to infer future share prices. Let's say we want to infer whether a share is a good investment or not in terms of its political circumstances. To do this, we use the opening and closing prices as well as the highs and lows. Share price changes can depend, for example, on the economic situation, i.e., the industry, or on political factors of a country. This relational information can be described in a dynamic probabilistic relational model (DPRM) that uses multiple random variables to encode uncertainty while decomposing the model into independent parts. Especially when many entities are represented in a model,

e.g., different shares and their relations, the DPRM quickly grows in size. In order to achieve a sparse representation, shares that show similar behaviour, e.g., in the context of policy restrictions, can be treated in a group. This means that in the DPRM, instead of using a random variable for each individual stock, only one (parameterised) random variable is used to describe the common symmetric behaviour of all of them. This technique is also known as *lifting* and is not only applied in the context of DPRMs but is widely used in the field of AI (Agostini and Celaya, 2009; Dieleman *et al.*, 2016; Satorras *et al.*, 2021). Lifting also speeds up inference, as when answering queries, calculations only need to be performed for a group of similarly behaving entities, rather than performing the same calculation for every single random variable in the model. Kersting (2012) provides a detailed overview of techniques from the field of lifted probabilistic inference that exploit symmetries in graphical models to speed up inference by orders of magnitude.

While the detection of exact symmetric behaviour, i.e., two or more entities having exactly the same behaviour, is not realistic, often approximately symmetric behaviour with a bounded error is desired, i.e., the behaviour of several entities is approximately the same over time. In this case, shape-based rather than value-based similarity measures are of interest, i.e., measures such as ordinal patterns that focus on the overarching behaviour as the general up and down movements in a time series. In Chapter 12, we present an approach to approximate symmetries based on multivariate ordinal patterns in DPRMs. We substantiate computational as well as other advantages in an experimental setting.

2.3 Using Ups and Downs as Features in Supervised Learning

Ordinal pattern representations play a role not only for solving the inverse problem, or for identifying symmetries in time series data. As a third and final example, we motivate the use of ordinal pattern representations in supervised learning tasks. Supervised learning describes a particular type of ML. In supervised learning, a function is learned that maps an input to an output, based on training pairs (observations or samples). A training pair always consists of an input object (raw data or representations of the raw data, also called features, predictor, or independent variable) and a desired output value (target, label or dependent variables). Depending on the type of target variable that supervised ML algorithms predict, algorithms can be divided into two groups: If the output variable is numerical (or continuous), it is called a regression, while in classification the variable is categorical (or discrete). Classifications are either right or wrong in a single prediction, while regressions have an error on a continuous scale. In addition, some classification models are able to predict not only the class but also the probability to belong to a class. This is particularly related to the chosen error measure.

There are a variety of models and algorithms for supervised learning, such as

multiple linear regression (MLR), support vector machine (SVM), linear discriminant analysis, decision trees, k-nearest neighbour algorithm, naive Bayes, or artificial neural network (ANN), all of which have their strengths and weaknesses. Some models are suitable only for regression, some only for classification, and others for both. The *no free lunch theorem* states that no single learning algorithm is best suited for all supervised learning problems (Wolpert and Macready, 1997). The prediction quality of any supervised learning algorithm depends on several factors as described in the following (Hastie *et al.*, 2009). The expected error of a learning algorithm with respect to a given problem can be represented as the sum of three terms: The bias, the variance and an irreducible error, where the irreducible error results from the noise of the problem itself. Bias is an error that arises from faulty assumptions in the learning algorithm. A high bias can cause an algorithm to fail to recognise the relevant relationships between predictors and target (underfitting). Variance is the error from sensitivity to small variations in the training data. A high variance can cause that the noise in the training data is modelled instead of the intended output (overfitting). A learning algorithm with low bias needs to be "flexible" so that it can adapt well to the data. However, if the learning algorithm is too flexible, it adapts differently to each training data set and therefore has a high variance. The choice of learning algorithm should therefore be based on a trade-off between bias and variance.

Furthermore, the flexibility or complexity of the learning algorithm depends on the amount of training data available. If the underlying "true" model function is simple, an "inflexible" learning algorithm with high bias and low variance can learn the model function from a small amount of training data. However, if the true function is complex, i.e., several predictor interactions need to be modelled, then only a "flexible" learning algorithm with low bias and high variance can learn the model function from a large amount of training data. Not only the number of training data is important to build an accurate prediction model, but also the properties of the training data set. A good training data set should be heterogen, but include no redundancies A good training data set should contain heterogeneous data, as well as interactions and nonlinearities, but no redundancies. Most importantly, the dimensionality of the input space is crucial. As input, classical ML-algorithms usually expect numerical random variables that are stochastically independent, while the individual values of a time series are only independent under certain conditions (Duboue, 2020). Therefore, it is necessary to extract scale-valued representations (features) from a time series before a learning algorithm can be applied. In addition, the number of extracted features plays an important role. If the vectors of the input features have a very high dimension, while the true function depends only on a small number of these features, poor prediction results may occur. This is because the additional dimensions can lead to a high variance and the learning algorithm overfits. With appropriate feature selection algorithms the optimal subset of suitable features can be determined. However, such approaches only lead to a good predictive model with high accuracy if the features contain information that contributes to the solution of the problem.

Ordinal pattern are interesting candidates as features in a supervised learning task

because they encode intrinsic up and down movements in a time series that follows a natural logic of its generating system. In combination with information-theoretic entropies, ordinal patterns are promising because the encoding preserves the information content and can therefore be used well for prediction, classification or further learning tasks (Amigó, 2010). Although details of explicit values or original amplitude information are lost in the ordinal pattern symbolisation approach, meaningful quantifications of the underlying system dynamics are still possible. Roughly speaking, ordinal pattern representations enable a reduction of complex systems to their intrinsic basic structure, on the basis of which classifications or predictions are supported. In Chapter 13, we elaborate on the use of ordinal pattern representations in supervised learning on the basis of various regression and classification tasks.

In conclusion, there is one major idea that links all three motivational examples in this chapter. To solve a data-driven problem, whether it is estimating a parameter to create a generalised system, identifying approximate symmetries for lifting models, or usually classifying data, it is always a matter of finding a suitable characteristic, representation or feature from which a learning algorithm can learn. And what could be more intuitive than describing the up and down movements in a time series to describe a temporal behaviour of an object, just as we would describe it to a friend?

Chapter 3

Preliminaries

In this section, we introduce the world of dynamical systems and time series and then present fractional Brownian motion (fBm) as a special case of it. To examine dynamical systems and time series for qualitative properties, we introduce univariate ordinal pattern representations and permutation entropy as an analytical tool. Finally, we recapitulate univariate results from the evaluation of ordinal patterns, i.e., up and down movements, in fBm and incorporate them into the overall contribution of this work, namely multivariate investigations and extensions.

Basic work related to entropy or time series is first presented in the following blog posts.

Marisa Mohr. The Mystery of Entropy: Measuring Unpredictability in Machine Learning. *inovex Blog*, May 2019. `https://www.inovex.de/blog/the-mystery-of-entropy-how-to-measure-unpredictability-in-machine-learning/`

Marisa Mohr. The Mystery of Time Series: Why Dealing with Time Series is Difficult. *inovex Blog*, July 2021. `https://www.inovex.de/de/blog/the-mystery-of-time-series/`

3.1 From Dynamical Systems to Time Series

Time series data are part of many applications in practice, as enumerated at the beginning of this work. More precisely, a time series is a temporal sequence of any data, e.g., a series of measurements (with possibly continuous range). Handling of time series requires a precise mathematical framework, which we introduce as follows.

Time series can be understood as a sequence of (real-valued) random variables of a time-dependent system also called *dynamical system* $(\Omega, \mathcal{A}, \mu, \mathcal{T})$.

- The set Ω, also called *state space*, includes all possible states of the system. As with non-time-dependent variables, states include latent variables, i.e., variables whose values are difficult or impossible to measure. For example, one's income is often dependent on one's reputation, which is determined individually by the recruiter and thus cannot be measured.

- Events are taken from a *σ-algebra* $\mathcal{A}$ on Ω. Then $(\Omega, \mathcal{A})$ is a measurable space.

- We assume that all states follow a probability distribution μ on the measurable space $(\Omega, \mathcal{A})$.
- The function $\mathcal{T}$ is a map describing the change or dynamics of the system at times $t \in \mathbb{N}$.

At time t the system is in state w, at time $t+1$ the system is in state $\mathcal{T}(w)$ and at time $t+s$ the system is in state $\mathcal{T}^{\circ s}(w)$, where

$$\mathcal{T}^{\circ s}(\omega) := \mathcal{T}(\dots \mathcal{T}(\mathcal{T}(\omega))) \tag{3.1}$$

is the s-fold concatenation of $\mathcal{T}$. Moreover, we assume that the function $\mathcal{T} : \Omega \to \Omega$ is μ-preserving, i.e., for all $A \in \mathcal{A}$ is $\mu(\mathcal{T}^{-1}(A)) = \mu(A)$. This ensures an identical distribution of the values of a variable over time, which means that each point in time provides information about the same data-generating process. A sequence of random variables or vectors, all defined on the same probability space $(\Omega, \mathcal{A}, \mu)$, is called a *stochastic process*. In the case of real-valued random variables, a stochastic process is a function

$$X : \Omega \times \mathbb{N} \to \mathbb{R} \tag{3.2}$$

$$X(\omega, t) := X_t(\omega). \tag{3.3}$$

A stochastic process depends on both coincidence and time. Note that in the most simple case, Ω matches with $\mathbb{R}$ and X with the identity map. Then the observations are directly related to iterations of some ω, i.e., there is no latency, and the X itself is redundant. Over time, the values of an individual variable $X_t(\omega)$ of this stochastic process get observed values (expression of a characteristic or several characteristics, observations), so-called realisations $(x(t))_{t\in\mathbb{N}}$. The sequence of realisations is called *time series*. With the formalism from above, and fixing of some $\omega \in \Omega$, a time series is given by

$$(X(\omega), X(\mathcal{T}(\omega)), X(\mathcal{T}^{\circ 2}(\omega)), \dots) = (x(t))_{t\in\mathbb{N}}. \tag{3.4}$$

In many cases, the terms *time series* and *stochastic process* are used interchangeably. The subtle difference is shown in Table 3.1.

	t fixed	t variable
ω fixed	$X_t(\omega)$ is a **real-valued number** or **vector** denoted as x	$X_t(\omega)$ is a sequence of real-valued numbers (path, **time series**) denoted as $(x(t))_{t\in\mathbb{N}}$
ω variable	$X_t(\omega)$ is a **random variable** denoted as X	$X_t(\omega)$ is a **stochastic process** denoted as $(X(t))_{t\in\mathbb{N}}$

Table 3.1: Difference between time series and stochastic processes.

Thus, for stochastic processes we use the capitalisation $(X(t))_{t\in\mathbb{N}}$, while for observations, i.e., paths or time series, we use the small notation $(x(t))_{t\in\mathbb{N}}$. In the case $X(t) \in \mathbb{R}$ or $x(t) \in \mathbb{R}$, the stochastic process or time series, is called univariate, while in the case $X(t) \in \mathbb{R}^m$ or $x(t) \in \mathbb{R}^m$ it is called multivariate.

3.2 Fractional Brownian Motion

As motivated in Section 2.1, modelling time-dependent real-world phenomena by dynamical systems requires properties such as *long-range dependence* or *self-similarity*. Such properties play an important role in different research fields such as physics, geophysics, hydrology, climatology, internet traffic, telecommunications, network engineering, linguistics, finance or econometric (Beran *et al.*, 2013). Fractional Brownian motion, a special class of stochastic processes, describes the properties of long-range dependence and self-similarity and is subject of this work, on which we focus in Part II.

> We believe that fBms provide useful models for a host of natural time series and that their curious properties deserve to be presented to scientists, engineers and statisticians.
>
> Mandelbrot and Ness (1968)

3.2.1 Univariate Fractional Brownian Motion

A stochastic process or, more generally, a mathematical object that is similar to itself at all scales is called a fractal. When you zoom in on a fractal, it resembles or looks exactly like the original shape. The mathematical property to describe such phenomena is called self-similarity and is expressed in honour of Harold Edwin Hurst by the so-called *Hurst exponent*, *Hurst index* or *Hurst parameter* and denoted as H (Mandelbrot and Wallis, 1968).

Definition 3.2.1 (Self-similar process)**.** A stochastic process $(X(t))_{t\in\mathbb{R}}$ is called *self-similar*, if there exists $H \in (0, 1)$ such that for any real $a > 0$ it holds that

$$X(at) \sim a^H X(t), \tag{3.5}$$

where $\sim$ denotes the equality of probability distribution.

Intuitively, self-similarity describes the phenomenon that certain process properties are preserved regardless of scaling in time or space. At the same time, long-range dependence means that the behaviour of a time-dependent process exhibits statistically significant correlations over large time scales. Self-similar processes are the simplest way to model long-dependent processes.

An important class of stochastic processes that fulfil self-similarity and is used to model corresponding phenomena is defined as follows.

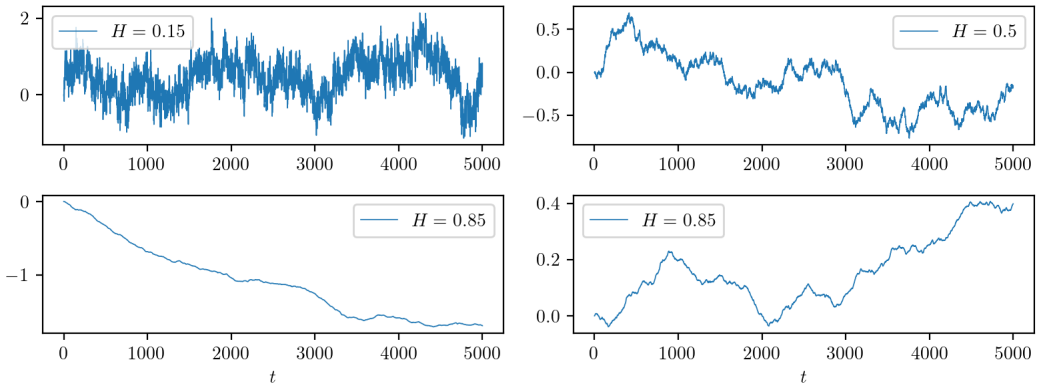


Figure 3.1: Four realisations of fBm with different Hurst parameters H.

Definition 3.2.2 (Fractional Brownian motion)**.** A Gaussian process with Hurst parameter $H \in (0,1)$ is called *fractional Brownian motion* (fBm) and denoted as $B_H(t), t \in \mathbb{R}$ if it holds

(i) $B_H(0) = 0$

(ii) $\mathrm{E}(B_H(t)) = 0, t \in \mathbb{R}$, and

(iii) $\mathrm{Cov}(B_H(t), B_H(s)) = \frac{1}{2}(|s|^{2H} + |t|^{2H} - |t-s|^{2H}), t, s \in \mathbb{R}$

As shown for example by Mishura and Mišura (2008), it follows that $B_H(t)$ is H-self-similar. In case $H = 1/2$, fBm corresponds to the ordinary Brownian motion. In case $H > 1/2$, the process has a persistence property and positively correlated increments, i.e., an upward jump is more likely followed by another upward jump and vice versa, and the process exhibits long-range dependence. For $H \to 1$, the process becomes smoother, less irregular and more trendy. In case $H < 1/2$, the process has negatively correlated increments and an anti-persistence property. Figure 3.1 shows three paths of fBms with different Hurst parameters.

In addition, it can be shown that fBm has stationary increments, i.e., $B_H(t) - B_H(s) \sim B_H(t-s)$, see also Mishura and Mišura (2008). Stationarity refers to the fact that the distribution of the process doesn't change in time, which has important consequences. In particular, the $B_H(t)$ are identically distributed, i.e., the expectation values and variances of components do not depend on time t. Furthermore, the joint distribution of $(B_H(t), B_H(s))$ depends only on $t - s$, so the correlations of the components also depend only on $t - s$.

3.2.2 Multivariate Fractional Brownian Motion

Multivariate generalisations of fBm are introduced by Lavancier *et al.* (2009) and Didier and Pipiras (2011). While the definition from Didier and Pipiras (2011) concerns a far-reaching generalisation of fBm, where self-similarity becomes a mathematical operator for the multivariate case, Lavancier *et al.* (2009) restrict themselves to joint self-similarity. For the sake of simplicity, we focus on the definition introduced by Lavancier *et al.* (2009). Thus, in the following, we extend self-similarity to multivariate processes first.

Definition 3.2.3 (Joint self-similarity (Lavancier *et al.*, 2009))**.** An m-multivariate stochastic process $X(t) = ((X^i(t))_{i=1}^m)_{t\in\mathbb{R}}$ is called *jointly self-similar*, if for any real $a > 0$ it holds that

$$X(at) \overset{fidi}{=} a^H X(t), \tag{3.6}$$

where $H = diag(H_1, \ldots, H_m)$ with $H_i \in (0, 1)$ for $i = 1, ..., m$ and a^H is intended in the matrix sense. The notion $\overset{fidi}{=}$ stands for equality of all the finite-dimensional probability distributions.

Using the properties of fBm, i.e., self-similarity and stationary increments, multivariate fractional Brownian motion (mfBm) can be defined as in the following definition. Note that mfBm is not to be confused with multifractional Brownian motion (mBm) or n-th-order fractional Brownian motion (n-fBm). The mBm is a natural extension of fBm where the Hurst parameter H changes over time, i.e., it is parameterised by a function $t \mapsto H(t)$ (Benassi *et al.*, 1997; Ayache and Véhel, 1999), while n-fBm is an fBm whose increments of order n are stationary (Perrin *et al.*, 2001).

Definition 3.2.4 (Multivariate fractional Brownian motion (Amblard and Coeurjolly, 2011))**.** An m-multivariate stochastic process $((X^i(t))_{i=1}^m)_{t\in\mathbb{R}}$ is called *multivariate fractional Brownian motion (mfBm)* with Hurst parameter $H = (H_1, \ldots, H_m)$, where $H_i \in (0, 1)$ for $i = 1, ..., m$, and denoted as $\mathbf{B}_H^m(t)$, if it is

(i) zero-mean Gaussian distributed,

(ii) jointly self-similar with Hurst parameter H, and it has

(iii) stationary increments, i.e., $\mathbf{B}_H^m(t) - \mathbf{B}_H^m(s) \sim \mathbf{B}_H^m(t - s)$.

As in the univariate case, the self-similarity property is associated with the covariance structure of mfBm. Moreover, joint self-similarity imposes many constraints on the covariance structure of mfBm derived from Amblard and Coeurjolly (2011). In the following, we first introduce corresponding parameters before repeating the covariance function of mfBm. In general, the parameters $\sigma_i > 0$, $\rho_{ij} \in (-1, 1)$ and $\eta_{ij} \in \mathbb{R}$ for $i, j = 1, ..., m$ allow for two variables $i, j \in \{1, \ldots, m\}$ to be more or less correlated and the process to be reversible in time or not.

- Scaling parameter $\sigma_i > 0$ is the standard deviation of the i-th variable at time 1.

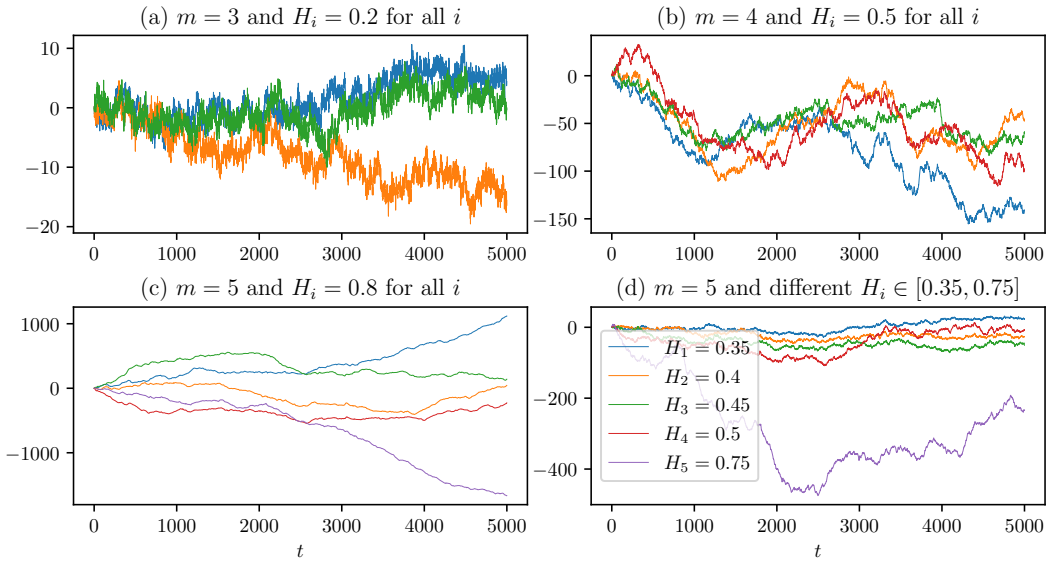


Figure 3.2: Four realisations of mfBm of length $T = 5000$ with different numbers of variables m and Hurst parameters H.

- Parameter $\rho_{ij} = \rho_{ji}$ is the correlation coefficient between the variables i and j at time 1.

- Parameter $\eta_{ij} = -\eta_{ji}$ is antisymmetric, and they are linked with the time-reversibility of mfBm. Time reversibility in a deterministic process means that the time-reversed process satisfies the same dynamic equations as the original process. In other words, the equations are invariant or symmetric when the sign of time changes. Furthermore, time reversibility for a stochastic process means that the statistical properties of the process are the same as the statistical properties for time-reversed data of the same process. In general, the antisymmetry parameters $\eta_{ij} = -\eta_{ji}$ are unrestricted. If the process is reversible in time, they are all zero. If the process allows for a causal (or anti-causal) representation, they are a function of ρ_{ij}, H_i and H_j, as exemplify in the simulation in Fig. 3.2 discussed later.

Multivariate fractional Brownian motion can be characterised by its covariances and cross-covariances of its variables as follows.

Lemma 3.2.1 (Covariance Function of mfBm (Amblard and Coeurjolly, 2011))**.** *The mfBm $\mathbf{B}_H^m(t)$ is marginally an fBm, such that the covariance function of the i-th variable $B_{H_i}^i$ of mBfm is as in the univariate case*

$$\mathrm{Cov}(B_{H_i}^i(s), B_{H_i}^i(t)) = \frac{\sigma_i^2}{2}(|s|^{2H_i} + |t|^{2H_i} - |t-s|^{2H_i}). \tag{3.7}$$

where $\sigma_i^2 = \mathrm{Var}(B_{H_i}^i(1))$. *The cross-covariances of mfBm for all* $(i,j) \in \{1,...,m\}^2$ *and* $i \neq j$ *are given by*

$$\mathrm{Cov}(B_{H_i}^i(s), B_{H_j}^j(t)) = \frac{\sigma_i \sigma_j}{2}(w_{ij}(-s) + w_{ij}(t) - w_{ij}(t-s)), \tag{3.8}$$

where the function $w_{ij}(h)$ *is defined by*

$$w_{ij}(h) = \begin{cases} (\rho_{ij} - \eta_{ij}\,\mathrm{sign}(h))|h|^{H_i+H_j} & \text{if } H_i + H_j \neq 1, \\ \rho_{ij}|h| + \eta_{ij} h \log|h| & \text{if } H_i + H_j = 1. \end{cases} \tag{3.9}$$

A setting of $\rho_{ij} = 1$ and $\eta_{ij} = 0$ in Eqs. (3.8) and (3.9) matches with the univariate case in Definition 3.2.2 (iii). Note that in the literature the covariance between the values of a stochastic process $X(t)$ at different times t_1 and t_2 is also called *autocovariance function* given by

$$\gamma(t_1,t_2) = \mathrm{Cov}(X(t_1), X(t_2)) = \mathrm{E}[(X(t_1) - \mu(t_1))(X(t_2) - \mu(t_2)], \tag{3.10}$$

where $\mu(t) = \mathrm{E}[X(t)]$ is the mean function of $X(t)$ and $\gamma(t_1,t_2) \in \mathbb{R}$. The *autocorrelation function* ρ of the stochastic process is defined as the normalised autocovariance function, i.e.,

$$\rho\,(t_1,t_2) = \frac{\gamma\,(t_1,t_2)}{\sigma(t_1)\sigma(t_2)}, \tag{3.11}$$

where $\sigma(t)$ is the standard deviation of $X(t)$ and $-1 \leq \rho(t_1,t_2) \leq +1$.

Although mfBm depends not only on a number of variables $m \in \mathbb{N}$ and a Hurst parameter $H \in \mathbb{R}^m$ with $H_i \in (0,1)$, but also on a scaling parameter $\sigma \in \mathbb{R}^m$, correlation coefficient $\rho_{ij} \in \mathbb{R}^{m \times m}$ and time-reversibility parameter $\eta_{ij} \in \mathbb{R}^{m \times m}$, for simplicity we write $\mathbf{B}_H^m(t)$ for short. Actually, the 5-tuple $(m, H, \sigma, \rho, \eta)$ is needed to define an mfBm and to realise it, for example, using the properties described above. As an example, Fig. 3.2 shows four realisations of mfBm of length $T = 5000$ with different numbers of variables m, different Hurst parameters H, $\sigma_i = 1$, $\rho_{ij} = 0.3$, and $\eta_{i,j} = 0.1/(1 - H_i - H_j)$.

A popular task, as previously introduced in Section 2.1, is the estimation of an mfBm or its descriptive parameters from observed values, also known as inverse problem. Other prediction tasks related to mfBm are classical tasks in the sense of time series problems, such as predicting future values or classifying and clustering multiple realisations. As described in Section 2.3 motivated, such learning tasks can be solved e.g. with known algorithms of supervised, unsupervised or semi-supervised learning.

For all these learning tasks, it is necessary to identify intrinsic representations that describe certain properties of mfBm. In the following, we consider the mapping of mfBm in particular or of time series in general into the ordinal pattern space.

3.3 Ordinal Pattern Representations

After an introduction to the concepts of dynamical systems, stochastic processes and time series and a deep dive into fBm, a special class of stochastic processes, we introduce ordinal pattern representations. Ordinal pattern representations are scalar-valued representations for time series based on time series symbolisation and entropy and are used to describe qualitative characteristics of a time series. While classical time series analysis examines the values of the time series themselves, symbolic time series analysis considers a non-parametric mapping or encoding into a sequence of symbols. The origins of symbolisation are controversial, ranging from 1898, when the French mathematician Jacques Salomon Hadamard (1898) studies geodesics on surfaces with negative curvature, to a paper by Hedlund (1944), in which he introduces the notion of an orbit for a discrete dynamical system, opening the door to symbolic dynamics (Coven and Nitecki, 2008). Apart from that, the idea of time series symbolisation is promising, especially in the case of dynamical systems such as fBm. The overall dynamics of the generating system are revealed by analysing the system's realisations in terms of an inverse problem (Traversaro *et al.*, 2018).

3.3.1 Time Series Symbolisation

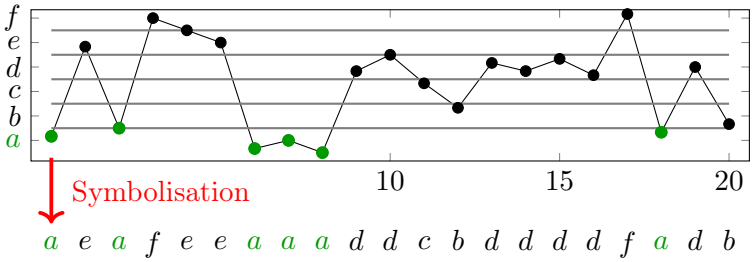
In the mathematical theory of symbolic dynamics, a dynamical system is modelled by a discrete space consisting of infinite (sequences of) abstract symbols, each of which corresponds to a state of the system. These sequences of symbols are the subject of advanced analytics and predictive models. As far as current research is concerned, there are two general approaches to encode the sequence of real-valued measurements into a sequence of symbols as visualised in Figure 3.3. On the one hand, classical symbolisation partitions the data range according to specified mapping rules to encode a numerical time series into a sequence of discrete symbols from a predefined alphabet Σ. An example of the partitioning of the data range and can be found in Fig. 3.3a, while the assignment of the symbols (encoding) to the time series is visualised in Fig. 3.3b. A corresponding and well-known algorithm for determining data range partitions is Symbolic Aggregate approXimation (SAX) introduced by Chiu *et al.* (2003). On the other hand, ordinal pattern symbolisation is another approach that, independent of the data range of the time series, encodes the total order between two or more neighbours into symbols. This ordinal pattern approach is based on an idea of Bandt and Pompe (2002) and visualised in Figs. 3.3c and 3.3d. Since we use the ordinal symbolisation scheme in the remainder of this work, we present its formalism and advantages in detail as follows.

Ordinal patterns describe the total order between two or more neighbours in a path or time series encoded by permutations.

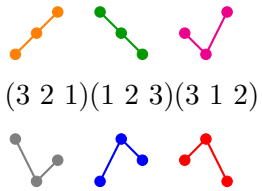
Definition 3.3.1 (Univariate Ordinal Pattern)**.** A vector $(x_1, \ldots, x_d) \in \mathbb{R}^d$ has *ordinal pattern* $(r_1, \ldots, r_d) \in \mathbb{N}^d$ of order $d \in \mathbb{N}$ if $x_{r_1} \geq \cdots \geq x_{r_d}$ and $r_{l-1} > r_l$ in the case $x_{r_{l-1}} = x_{r_l}$.

$a = (-\infty, 10]$
$b = (10, 20]$
$c = (20, 30]$
$d = (40, 50]$
$e = (50, 60]$
$f = (60, \infty]$

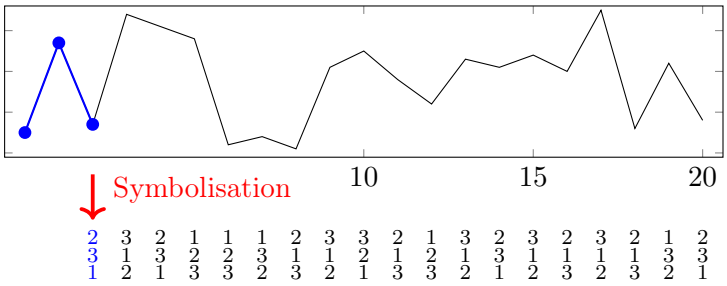
(a) All symbols from an alphabet Σ of size $|\Sigma| = 6$.



(b) Partitioning of the data range and classical symbolisation determination in a univariate time series.



(c) All possible ordinal patterns of order $d = 3$.



(d) Ordinal pattern determination of order $d = 3$ and time delay $\tau = 1$ at any time point $t \in [d\tau - \tau + 1, T]$.

Figure 3.3: Two approaches for symbolising a univariate time series: (a) and (b) classical symbolisation and (c) and (d) ordinal pattern symbolisation. Best viewed in colour.

Figure 3.3c shows all possible ordinal patterns of order $d = 3$ of a vector (x_1, x_2, x_3). To symbolise a time series $(x_1, x_2, ..., x_T) \in \mathbb{R}^T$ each point in time $t \in \{d, ..., T\}$ is assigned its ordinal pattern of order d. The order d is chosen to be much smaller than the total length T of the time series to look at smaller sub-sequences, i.e., ordinal pattern windows of order d, within the series and the distributions of respective "up and down" movements. To assess the overarching trend, delayed behaviour is of interest. The time delay $\tau \in \mathbb{N}$ is the delay between successive values in the tine series. Different delays show different details of the structure of a time series. Figure 3.3d visualises the ordinal pattern determination of order $d = 3$ and time delay $\tau = 1$ of three different time points in a univariate time series marked in blue, orange and magenta. Note that ordinal patterns are usually determined at an arbitrary point in time, as illustrated below the graph in form of vectors.

The ordinal approach has notable advantages in its practical application. First of all, the method is conceptually simple as it reflects man's natural thought of up and down movements and is therefore open to interpretation. Second, prior knowledge of the data range or the type of time series is not necessary. The concept can be applied to any time series as long as the range of values is ordered, e.g., $x_t \in \mathbb{R}$. Third, the

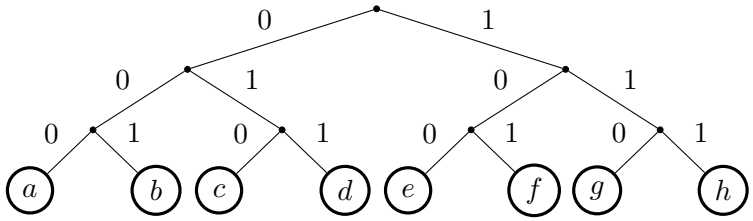


Figure 3.4: Encoding of symbols with bits.

ordinal approach supports robust and fast implementations (Keller *et al.*, 2017; Piek *et al.*, 2019). Fourth, it allows for an easier estimation of a good symbolisation scheme compared to the classical symbolisation approaches, as no data range partitioning is necessary (Keller *et al.*, 2015; Stolz and Keller, 2017).

3.3.2 Symbol Distributions as a Measure of (Dis)order

By symbolising the time series, the sequence of values, usually $x_t \in \mathbb{R}$, is encoded into a sequence containing low-dimensional representations or symbols $x_t \mapsto z \in \mathbb{N}$. Often it is not the symbols or ordinal patterns themselves, but their distributions in different parts of a time series $(x_t)_{t=1}^{T}$ that are of interest. To assess the (dis)order of the identified symbols in the system or in the time series, we use a dispersion measure from the field of statistics, namely entropy. The statistical interpretation of entropy corresponds to Shannon's information entropy used in information theory or computer science.

Shannon's Information Entropy

Let's start with an information-theoretical view. Knowing that a computer computes with bits, we can ask "How many bits does the computer need to display or transmit an information, e.g., a string of characters?" If the computer needs few bits, the information is less complex. If the computer needs many bits, the information has a high complexity. If one wants to encode $n = 8$ symbols from the Latin alphabet $X = \{a, b, c, e, f, g, h\}$ in bits, for example, to display them on the screen, then sequences of length 3 bits are needed, i.e., $a = 000$, $b = 001$, $c = 010$ etc., to uniquely encode all $n = 8$ symbols. Figure 3.4 shows the corresponding encoding tree in which each individual symbol is encoded by a unique sequence of bits. The display of a symbol can be interpreted as a statistical event occurring, e.g., with the same probability $p = 1/n$. The *minimum number of bits* needed to represent a symbol can therefore also be formulated as the height of the coding tree or its statistical significance given by

$$\log_2(n) = \log_2\left(\frac{1}{p}\right) = \log_2(1) - \log_2(p) = -\log_2(p). \tag{3.12}$$

The base 2 for the logarithm corresponds to the unit of bits (binary digits).

Since for all positive real numbers $m, n > 1$, there exists some real number k, so that for all positive real numbers x

$$\log_m(x) = k \cdot \log_n(x) \tag{3.13}$$

applies, the choice of the logarithm does not matter. For reasons of simplicity, we choose the natural logarithm in the following.

Furthermore, symbols or events do not always have to be uniformly distributed, i.e., $p \neq 1/n$. For example, in every language, characters, such as e or a, occur more frequently than others. For non-uniformly distributed characters, we are interested in the *expected number of bits* to represent a message, which can be determined by

$$\mathrm{I}((p_1, p_2, \ldots, p_n)) = \mathrm{E}(-\ln(p_i)) = -\sum_{i=1}^{n} p_i \cdot \ln p_i \tag{3.14}$$

and is called (Shannon's information) entropy, where p_i is the frequency of symbol i. The greater the disorder in the symbols, the closer they are uniformly distributed, the higher the entropy.

Depending on the area of research, entropy is a measure for quantifying inhomogeneity, impurity, complexity and uncertainty or unpredictability. The information-theoretical (left-hand side of Eq. (3.12)) and statistical (right-hand side of Eq. (3.12)) coherence becomes particularly clear in Eq. (3.12) and thus also what the words inhomogeneity or impurity have in common with uncertainty or unpredictability. For more details related to entropy, we refer to basic work published by Mohr (2019).

Permutation Entropy

Applying the well-known formula of (Shannon) entropy from Eq. (3.14) to the distribution of ordinal patterns leads directly to the definition of permutation entropy. For this purpose, each pattern is previously identified with exactly one of the *ordinal pattern symbols* $j = 1, 2, ..., d!$.

Definition 3.3.2 (PE (Bandt and Pompe, 2002))**.** The *permutation entropy* (PE) of order $d \in \mathbb{N}$ and delay $\tau \in \mathbb{N}$ of a time series $\mathbf{x} = (x_t)_{t=1}^{T}$, $T \in \mathbb{N}$ is defined as

$$\mathrm{PE}_{d,\tau}(\mathbf{x}) = -\sum_{j=1}^{d!} p_j^{d,\tau} \ln p_j^{d,\tau}, \tag{3.15}$$

where

$$p_j^{d,\tau} = \frac{\sum_{t \leq T} [(x_{t-\tau(d-1)}, \ldots, x_{t-\tau}, x_t) \text{ has pattern } j]}{T - \tau(d-1)} \tag{3.16}$$

is the frequency of ordinal pattern j in the time series, where $[x] = 1$ if x is true, 0 otherwise.

In time series with maximally random ordinal pattern symbols, the ordinal patterns tend to be equally distributed so that PE is $\ln(d!)$. For a time series with a regular pattern, e.g., in the case of strict monotony, PE is equal to zero (Amigó, 2010). The search for an optimal order d and time delay τ is a challenging problem in research, which we leave to hyperparameter optimisation in this work, see also Riedl *et al.* (2013); Myers and Khasawneh (2020). As a guideline, $d = 2, \ldots, 7$ is recommended.

Multi-Scale-Permutation Entropy

While the associated entropy is low for a deterministic time series, it approaches its maximum value in case of complete randomness or high complexity. To distinguish between randomness and complexity, an additional measure is necessary. As a complement, Costa *et al.* (2002) introduce multi-scale entropy (MSE), an approach that captures the complexity of time series on different time scales. On higher time scales, random noise tends to cancel out, resulting in a low entropy measurement, where complex signals retain a high entropy. In this manner, it is possible to gain a deeper insight into the randomness and complexity of a system. Morabito *et al.* (2012) extend the concept of MSE to univariate ordinal patterns and introduce MSPE.

For the consideration of different scales of the time series and an associated definition, a coarse-grained procedure is used: From the original time series, several consecutive time data points are averaged within a non-overlapping time window of scaling length ϵ, also called *scaling factor*. Each element of the coarse-grained time series $\mathbf{y} = (y_l^{(\epsilon)})_{l=1}^{T/\epsilon}$ is calculated as

$$y_l^{(\epsilon)} = \frac{1}{\epsilon} \sum_{t=(l-1)\epsilon+1}^{l\epsilon} x_t \tag{3.17}$$

for $1 \leq l \leq \frac{T}{\epsilon}$. Note that if the scaling factor ϵ is 1, the coarse-grained time series is equal to the original time series. After coarse-grained processing, the original algorithm of PE from Eq. (3.15) is applied to the new sequence with length T/ϵ, which leads to the following definition.

Definition 3.3.3 (MSPE (Morabito *et al.*, 2012))**.** The *multi-scale permutation entropy* (MSPE) of order $d \in \mathbb{N}$ and delay $\tau \in \mathbb{N}$ of a univariate time series $\mathbf{x} = (x_t)_{t=1}^{T}$, $T \in \mathbb{N}$ is defined as PE of its coarse-grained time series $\mathbf{y} = (y_l^{(\epsilon)})_{l=1}^{T/\epsilon}$, that is

$$\mathrm{MSPE}_{d,\tau,\epsilon}(\mathbf{x}) = \mathrm{PE}_{d,\tau}(\mathbf{y}). \tag{3.18}$$

For purposes of analysis, MSPE is plotted as a function of scale factor ϵ. For example, Liu *et al.* (2017) analyse the MSPE of electrocardiogram (ECG) signals from people suffering from congestive heart failure (CHF) and healthy young and elderly people, using MSPE with a scaling factor from 2 to 100. When the scaling factor is between 10 and 32, the complexities of CHF patients, elderly and young people are in

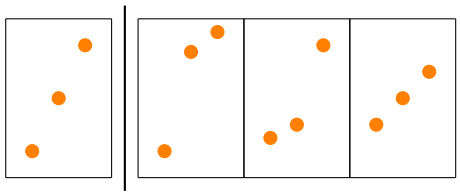


Figure 3.5: Same ordinal pattern of order $d = 3$ (left) for three different motifs (right).

ascending order, and therefore allow discriminations consistent with normal human characteristics and diagnoses. As another example, Guo *et al.* (2021) successfully use MSPE as a measure of the effects of sleep medication on the brain dynamics of patients with insomnia. In addition, Su *et al.* (2016) provide a comparison and detailed discussion of MSPE to five other related ordinal pattern representations in the context of an anaesthesia depth monitoring application to quantify anaesthesia effect on real-time electroencephalography (EEG) recordings.

Weighted Permutation Entropy

Another shortcoming in the above Definition 3.3.2 of PE is that no other information is preserved during the extraction of the ordinal patterns except for the order structure. Information about the amplitude in a time series, i.e., different motifs as visualised in Fig. 3.5, is lost. However, ordinal patterns with large differences in amplitude should contribute in different ways to the calculation of PE. Weighted permutation entropy introduced by Fadlallah *et al.* (2013) allows for the weighting of ordinal patterns by exploiting amplitude information resulting from small fluctuations in the time series due to the effect of noise to be weighted less than ordinal patterns with large amplitudes.

Definition 3.3.4 (WPE (Fadlallah *et al.*, 2013))**.** The *weighted permutation entropy (WPE)* of an univariate time series $\mathbf{x} = (x_t)_{t=1}^T$, $T \in \mathbb{N}$, with order $d \in \mathbb{N}$ and delay $\tau \in \mathbb{N}$ is defined as

$$\mathrm{WPE}_{d,\tau}(\mathbf{x}) = -\sum_{j}^{d!} p_{wj}^{d,\tau} \ln p_{wj}^{d,\tau} \tag{3.19}$$

with

$$p_{w_j}^{d,\tau} = \frac{\sum_{t \leq T} w_t \cdot [(x_{t-(d-1)\tau}, ..., x_{t-\tau}, x_t) \text{ has pattern } j]}{\sum_{t \leq T} w_t} \tag{3.20}$$

where

$$w_t = \frac{1}{d} \sum_{k=1}^{d} (x_{t-(k-1)\tau} - \overline{x}_t^{d,\tau})^2 \tag{3.21}$$

is the empirical variance of the sub-sequence and $\overline{x}_t^{d,\tau}$ denotes the arithmetic mean that is $\overline{x}_t^{d,\tau} = \frac{1}{d} \sum_{k=1}^{d} x_{t-(k-1)\tau}$ and $[x] = 1$ if x is true, 0 otherwise.

Fadlallah *et al.* (2013) perform various analyses on synthetic data and also human EEG recordings using WPE in the context of epilepsy detection. The EEG recordings were previously narrow-band filtered to account for the variance of the patterns mentioned above. When examining two separate EEG signals on different regions of the brain, WPE proved consistent in distinguishing different regimes and assigning similar complexities to analogue portions. Note that WPE is suited for signals that contain significant amplitude information. Otherwise, PE might be the better choice.

Furthermore, Yin and Shang (2014) introduce weighted multi-scale permutation entropy (WMPE) by combining WPE and MSPE to analyse financial time series of the US and Chinese stock markets. Since the calculation of MSPE consists of two separate steps, namely a coarse-grained procedure followed by an ordinary entropy calculation, it is reasonable to combine both approaches, i.e., WPE and MSPE into WMPE. Through the coarse-grained procedure and the subsequent weighting introduced in Eqs. (3.20) and (3.21), both potentials, that of scaling and that of amplitude information, can be exploited. Xia *et al.* (2016), for example, successfully use WMPE to investigate the complexity of different traffic congestion indices series collected every 15 minutes by the Beijing Transportation Research Center. In particular, WMPE uncovers the differences between weekday and weekend time series.

Part I

New Approaches in Multivariate Ordinal Pattern Representations

Chapter 4

Review of Multivariate Extensions for Permutation Entropy

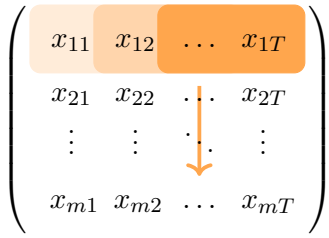
In real-world applications, one often is faced with multivariate time series, e.g., with medical measurements, which are stored as ECG data that is usually not determined from a single electrode but from multiple electrodes. Since ordinal pattern representations are designed for univariate time series, the concept has to be extended to the multivariate case in order to apply it for multivariate time series as well. Numerous studies introduce multivariate extensions of PE under the generic name "Multivariate Permutation Entropy", each claiming to be the most general approach. Nevertheless there are several differences between them, resulting in different strengths and weaknesses. In this work, by *multivariate permutation entropy* (MPE) we denote the class of all multivariate extensions of PE. We divide them into four strategies. The first three are based on the concept of univariate ordinal patterns, while the fourth strategy is based on a new concept of multivariate ordinal patterns. In this chapter we elaborate on the different characteristics of the strategies, before introducing and formalising multivariate ordinal patterns for MPE as a contribution of this work in the next chapter. The categorisation of MPE strategies was first presented or elaborated in the following conference paper or journal article, respectively.

Marisa Mohr, Florian Wilhelm, Mattis Hartwig, Ralf Möller, and Karsten Keller. New Approaches in Ordinal Pattern Representations for Multivariate Time Series. In *Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference (FLAIRS-33)*, pages 124–129, 2020

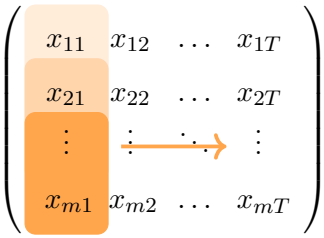
Marisa Mohr and Ralf Möller. A Summary of Canonical Multivariate Permutation Entropies on Multivariate Fractional Brownian Motion. *Advances in Science, Technology and Engineering Systems Journal*, 6(5):107–124, 2021

4.1 Multivariate Extensions for Permutation Entropy

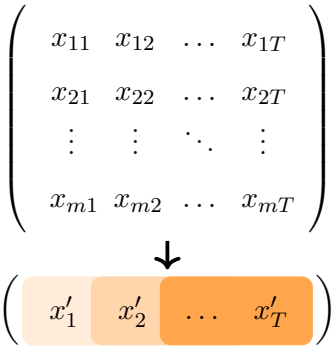
A multivariate time series $((x_t^i)_{i=1}^m)_{t=1}^T$ has more than one time-dependent variable. Each variable x^i for $i = 1, ..., m$ not only depends on its past values but also has dependencies on the values of other variables at the same time point, which we call *spatial variables*. Considering two time points $(x_t^i)_{i=1}^m$ and $(x_{t+1}^i)_{i=1}^m$ with m spatial



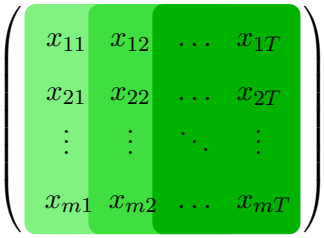
(a) Univariate ordinal pattern in time.



(b) Univariate ordinal pattern in phase space.



(c) Dimensionality reduction.



(d) Multivariate ordinal pattern.

Figure 4.1: Four strategies of MPE determination.

variables, simply put two vectors, it is not possible to establish a total order between them. A total order is only possible if $x_t^i > x_{t+1}^i$ or $x_t^i < x_{t+1}^i$ for all $i \in 1, ..., m$. Therefore, there is no trivial generalisation of the PE algorithm to the multivariate case. Nevertheless, there are several strategies in research to extend PE to the multivariate case. A multivariate time series is

a) projected into univariate ordinal space by determining univariate ordinal patterns *between neighbouring values in time space* (orange row in Fig. 4.1a) in each single variable i, and then pooled over all m variables for multidimensionality, or

b) projected into univariate ordinal space by determining univariate ordinal patterns *between values of all m variables* (orange column in Fig. 4.1b) for each single time point t, and then pooled over all T time points for multidimensionality, or

c) projected onto a single-dimensional reduction (bottom in Fig. 4.1c), and then transformed into ordinal space by determining univariate ordinal patterns, or

d) projected into multivariate ordinal space by determining the multivariate ordinal pattern *between d vectors of variable dimension m* (green box in Fig. 4.1d) for $d = 2$).

In the following, we provide definitions and algorithms to establish MPE in detail. Note that strategy d) is presented as contribution 1c) of this work in the next chapter and serves only as an outlook at this point. Similarly, strategies a) and c) are adapted in the next chapter.

4.1.1 Canonical Extensions

Procedure a), presented in Figure 4.1a, is a canonical extension of PE to the multivariate case, i.e., univariate ordinal patterns are determined for each spatial variable of the multivariate time series before each pattern is pooled over all variables. This idea is introduced by Keller and Lauffer (2003) and referred to as *pooled permutation entropy* (PPE). By analogy with MSPE, Morabito *et al.* (2012) introduce *multivariate multi-scale permutation entropy* (MMSPE), where based on PPE different scales are studied.

Pooled Permutation Entropy

The idea of PPE is to use marginal frequencies of $d!$ ordinal patterns regarding all m spatial variables as input for entropy computation. To determine marginal frequencies, an auxiliary matrix is created first:

1. For each variable $i = 1, .., m$ and for each ordinal pattern $j = 1, ..., d!$, count all time steps $s \in [\tau(d-1)+1, T]$, for which the variable-time pair (i, s) has the ordinal pattern j.
2. Divide the counts by $m \cdot \delta$, where $\delta := T - \tau(d-1)$ is the total count of ordinal patterns each variable has.
3. Store the results, i.e., frequencies $p_{ij}^{d,\tau}$ in a so-called *pooling matrix* $P \in (0,1)^{m \times d!}$, which reflects the distribution of ordinal patterns in the multivariate time series across its m variables.

It holds $\sum_{i=1}^{m} \sum_{j=1}^{d!} p_{ij}^{d,\tau} = 1$. For computational reasons, the marginal frequencies $p_{\cdot j}^{d,\tau} = \sum_{i=1}^{m} p_{ij}^{d,\tau}$ must not vanish for $j = 1, ..., d!$. If they vanish, set respective values close to zero.

Definition 4.1.1 (PPE (Keller and Lauffer, 2003))**.** The *pooled permutation entropy* (PPE) of a multivariate time series $\mathbf{X} = ((x_t^i)_{i=1}^m)_{t=1}^T$ is defined as PE of the marginal frequencies $p_{\cdot j}^{d,\tau} = \sum_{i=1}^{m} p_{ij}^{d,\tau}$ for $j = 1, ..., d!$ describing the distribution of the ordinal pattern and is calculated by

$$\mathrm{PPE}_{d,\tau}(\mathbf{X}) = -\sum_{j}^{d!} p_{\cdot j}^{d,\tau} \ln p_{\cdot j}^{d,\tau}. \tag{4.1}$$

Algorithm 1 provides pseudocode for computing PPE.

Algorithm 1: Computation of PPE

Input: Multivariate Time Series $X^{m\times T}$, Order d, Delay τ

Function `pooling(`X`,`d`,`τ`):`

- $P^{m\times d!} \leftarrow$ pooling matrix initialised with zeros
- **for** *every time series variable* $i = 1, \ldots, m$ *in* X **do**
  - **for** *each ordinal pattern* $j = 1, \ldots, d!$ **do**
    - $c \leftarrow$ number of time steps $s \in [\tau(d-1)+1, T]$ with pattern j
    - $P_{ij} \leftarrow$ divide c by the number of total time steps $\delta \cdot m$ in X
- **return** P

Function `marginalisation(`P`):`

- $p^{1\times d!} \leftarrow$ vector of marginalisation initialised with zeros
- **for** *every column* $j = 1, \ldots, d!$ *in* $P^{m\times d!}$ **do**
  - $p_j \leftarrow$ sum up p_{ij}
- **return** p

PPE $\leftarrow$ PE(`marginalisation(pooling(`X`,`d`,`τ`))`) `// for PE see Eq. (3.15)`

A major advantage of PPE is that Alg. 1 not only allows for a multivariate calculation, but by a simple intermediate step, it can still return the univariate PEs of each spatial variable. For this purpose, the calculation of PE from Eq. (3.15) is applied to each row of the pooling matrix P. Considering univariate and multivariate representations together can support the analysis in applications.

For example, PE and PPE are successfully used to analyse EEG signals by extracting individual as well as cross-channel regularities between spatially distant variables, i.e., on different hemispheres and/or in different areas (Keller and Lauffer, 2003). Furthermore, Keller *et al.* (2017) apply PPE to study the 19-channel scalp EEG reflecting changes in brain dynamics of a boy with lesions predominantly in the left temporal lobe caused by connatal toxoplasmosis. Nicolaou and Georgiou (2011) use PPE to characterise sleep EEG signals from more than 80 hours of nocturnal sleep recordings and classify sleep stages. They find that each sleep stage is characterised by statistically different PPE values and that the observed pattern of PPE is consistent with the physiological properties of the EEG in each sleep stage. Note that limitations of PPE are discussed in Section 4.2 in the context of further MPEs.

Multivariate Multi-Scale Permutation Entropy

With the introduction of MSPE in Definition 3.3.3, an extension of PE was introduced to investigate different time scales of a univariate time series. While in classical PE maximum randomness and high complexity both have a high entropy value, MSPE enables their distinction. Random noise tends to cancel out on high time scales, resulting in a low entropy value, while high complexity results in a high entropy value.

Algorithm 2: Computation of MMSPE

Input: Multivariate Time Series $X^{m\times T}$, Order d, Delay τ, Scale Factor ϵ

$Y^{m\times(T/\epsilon)} \leftarrow$ coarse-grained multivariate time series initialized with zeros
for *every time series variable* $i = 1, \ldots, m$ *in* X **do**
  $y_i \leftarrow$ calculate coarse-grained univariate time series `// see Eq.(4.2)`
MMSPE $\leftarrow$ PPE(Y, d, τ) `// see Algorithm 1`

As with MSPE, the multivariate analogue, i.e., multivariate multi-scale permutation entropy (MMSPE), is calculated on different time scales of the multivariate time series by processing a coarse-grained multivariate time series depending on the scale factor ϵ. Per variable i in a multivariate time series, several consecutive time data points are averaged within a non-overlapping time window of the scaling length ϵ. Each element of the coarse-grained time series $\mathbf{Y} = ((y_{i,l}^{(\epsilon)})_{l=1}^{T/\epsilon})_{i=1}^{m}$ is calculated as

$$y_{i,l}^{(\epsilon)} = \frac{1}{\epsilon} \sum_{t=(l-1)\epsilon+1}^{l\epsilon} x_{i,t} \tag{4.2}$$

for all $i = 1, ..., m$ and $1 \leq l \leq \frac{T}{\epsilon}$.

Definition 4.1.2 (MMSPE (Morabito *et al.*, 2012))**.** The *multivariate multi-scale permutation entropy* (MMSPE) of order $d \in \mathbb{N}$ and delay $\tau \in \mathbb{N}$ of a multivariate time series $\mathbf{X}$ is defined as PPE of its coarse-grained time series $\mathbf{Y}$, that is

$$\mathrm{MMSPE}_{d,\tau,\epsilon}(\mathbf{X}) = \mathrm{PPE}_{d,\tau}(\mathbf{Y}). \tag{4.3}$$

Algorithm 2 provides pseudocode for computing MMSPE. Note that the definition does not include variation across different scales. Nevertheless, the simultaneous utilisation of a multi-scale approach and the consideration of multiple spatial variables of the time series facilitates assessing the complexity of the underlying dynamical system. For example, Morabito *et al.* (2012) compare the discriminability of 3-channel EEG data for three different categories of individuals: HC (Healthy elderly Control), MCI (Mild Cognitively Impaired patients) and AD (Alzheimer Disease patient). They highlight the potential utility of the MMSPE as a complementary approach to the assessment of ageing processes in Alzheimer Disease.

Note that Morel and Humeau-Heurtier (2021) propose the bidimensional multi-scale permutation entropy, MPE2D for short. Since this approach is specifically designed for 2D image data and texture evaluations, we refrain from considering it in general discussion.

4.1.2 Extensions via Spatial Dependencies

The previous approach uses univariate ordinal patterns in time, while the interactions of different spatial variables at a fixed time point are not considered. While canonical approaches measure the complexity of each univariate spatial variable in time space, procedure b), that is visualised in Figure 4.1b, analyses the complexity of the spatial dependencies. A variant of this procedure is introduced by He *et al.* (2016) as *multivariate permutation entropy* (MvPE). To ensure the uniqueness of this measure in relation to the general class of multivariate permutation entropy (MPE), we always use the abbreviation MvPE in the following when referring to this representation.

Let $\mathbf{X} = ((x_t^i)_{i=1}^m)_{t=1}^T$ be a multivariate time series. As the data range of each spatial variable in $\mathbf{X}$ may be different, each variable i has to be normalised first. Thus, the min-max scaling is applied, i.e.,

$$\tilde{x}_t^i = \frac{x_t^i - \min((x_t^i)_{t=1}^T)}{\max((x_t^i)_{t=1}^T) - \min((x_t^i)_{t=1}^T)} \tag{4.4}$$

where $\max(\cdot)$ is the maximum function and $\min(\cdot)$ is the minimum function. By doing so, all values of the time series are transformed into the range $[0, 1]$.

Based on the normalised multivariate time series, MvPE is defined as follows.

Definition 4.1.3 (MvPE (He *et al.*, 2016))**.** The MvPE of order $d \in \mathbb{N}$ of a multivariate time series $\mathbf{X} = ((x_t^i)_{i=1}^m)_{t=1}^T$ is defined as

$$\mathrm{MvPE}_d(\mathbf{X}) = -\sum_{j=1}^{d!} p_j^d \ln p_j^d, \tag{4.5}$$

where

$$p_j^d = \frac{\sum_{t \leq T} [(\tilde{x}_t^1, ..., \tilde{x}_t^m) \text{ has pattern } j]}{T - (d-1)} \tag{4.6}$$

with $[x] = 1$ if true, and 0 otherwise, is the frequency of univariate ordinal patterns established over spatial variables.

As discussed in Section 3.3.2 by introducing WPE, changes in time series cannot always be detected by using ordinal patterns due to different amplitudes in the motifs. To get more details from the time series, He *et al.* (2016) propose the use of subpatterns. As shown in Fig. 4.2, the original ordinal patterns from Fig. 3.3c are subdivided into 18 possible (sub)motifs by means of thresholds, i.e., by the definition of

$$\begin{cases} Bot_t = \frac{2}{3}\min((\tilde{x}_t^i)_{i=1}^m) + \frac{1}{3}\max((\tilde{x}_t^i)_{i=1}^m) \\ Top_t = \frac{1}{3}\min((\tilde{x}_t^i)_{i=1}^m) + \frac{2}{3}\max((\tilde{x}_t^i)_{i=1}^m) \end{cases}. \tag{4.7}$$

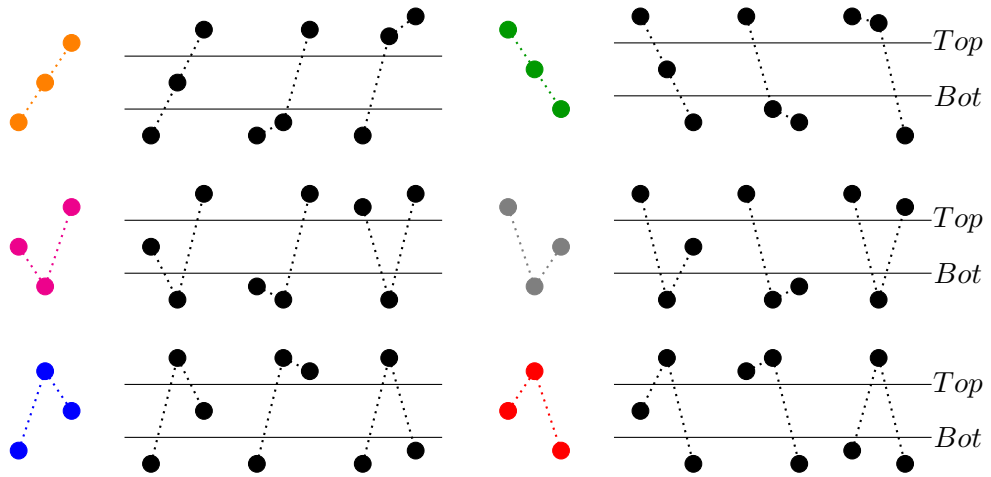


Figure 4.2: Ordinal patterns for a multivariate time series with $i = 3$ variables with 18 sub-patterns by MvPE.

To the best of our knowledge, the effectiveness of MvPE has only been demonstrated on synthetic data, i.e., in the context of the hyperchaotic Henon map or the simplified fractional-order Lorenz system, but not on real-world data. Since we discuss MvPE in this work only as a theoretical concept, and do not investigate it in any experimental setting, we refrain from presenting a corresponding algorithm.

4.1.3 Extensions via Dimensionality Reduction

In the previously presented approach, the extension via spatial dependencies, neighbourhood relations of the spatial variables are preserved. However, the neighbourhood relations in time – an advantage of the ordinal approach that makes its application so successful – are discarded by using the approach. The procedure c) presented in Figure 4.1c follows the idea of first reducing the number of spatial variables m to a single dimension by applying an arbitrary dimensionality reduction method to then encode the spatial neighbourhood relations in a single dimension. More specifically,

$$f : \mathbb{R}^{m \times T} \quad \rightarrow \quad \mathbb{R}^{1 \times T} \tag{4.8}$$

$$\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1T} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mT} \end{pmatrix} \mapsto \begin{pmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \dots & \tilde{x}_{1T} \end{pmatrix}. \tag{4.9}$$

Consequently, Definition 3.3.2 can be used for PE calculation directly. Note that the length T of each spatial variable of the multivariate time series must be the same to perform dimensionality reduction.

Rayan *et al.* (2019) proposes several approaches to reduce the dimensionality of the multivariate time series to a one-dimensional projection, transforming the problem of multivariate PE calculation into a one-dimensional evaluation. In particular, they propose to apply three different distance metrics specified by a reference point q, namely

1. Euclidean distance with reference point $q = (x_1^i)_{i=1}^m$, i.e.,

$$\tilde{x}_t = \sqrt{(x_t^1 - x_1^1)^2 + (x_t^2 - x_1^2)^2 + \cdots + (x_t^m - x_1^m)^2} \tag{4.10}$$

2. Manhattan distance with reference point $q = (x_1^i)_{i=1}^m$, i.e.,

$$\tilde{x}_t = |x_t^1 - x_1^1| + |x_t^2 - x_1^2| + \cdots + |x_t^m - x_1^m| \tag{4.11}$$

3. Euclidean distance with reference point $q = \mathbf{0}$, i.e.,

$$\tilde{x}_t = \sqrt{(x_t^1 - 0)^2 + (x_t^2 - 0)^2 + \cdots + (x_t^m - 0)^2} \tag{4.12}$$

at each time point t. After applying a distance measure to the original multivariate time series, the well-known univariate PE algorithm from Eq. (3.15) can be applied to the resulting single-dimensional time series $(\tilde{x}_t)_{t=1}^T$. We refer to three different possibilities of MPE in the following as

- multivariate permutation entropy based on Euclidian distance (MPE-EUCL),
- multivariate permutation entropy based on Manhattan distance (MPE-MANH), and
- multivariate permutation entropy based on normalisation (MPE-NORM).

For the evaluation of the proposed approaches, Rayan *et al.* (2019) use measurement data from body-worn motion sensors, each with 12 3D accelerometers and 4 3D localisation information recorded for human activity detection. Thus, these sensors provide a 36-dimensional time series that is used for motif discovery, i.e., the discovery of approximately recurring short patterns in long time-series. The localisation modes, consisting of four classes (standing = 1, walking = 2, sitting = 3, lying = 4), define four different motifs that are repeated in the time series, but in slightly different ways, corresponding to the definition of approximately recurring motifs. The authors show that all three proposed MPE variants outperform the classification accuracy using e.g. MMSPE. Moreover, the proposed MPE-variants tend to be twice as fast as MMSPE.

4.2 Limitations

In the previous section we introduced three procedures, in total six algorithms, of existing multivariate extensions of PE each with its representatives, namely

a) canonical: PPE, and MMSPE,

b) via spatial dependencies: MvPE,

c) via dimensionality reduction: MPE-EUCL, MPE-MANH, and MPE-NORM.

First of all, the field of canonical multivariate extensions has gaps in current research. In the same way that PE is extended by PPE and MSPE by MMSPE in an analogous way, we fill the gap and extend WPE by introducing multivariate weighted permutation entropy (MWPE) in the next chapter. As in the univariate case, amplitude information can be uncovered using MWPE, resulting in advantages in subsequent analyses.

All representations are based on the idea of univariate ordinal patterns. This means that univariate ordinal patterns are established either in time (procedure a)) or in space (procedure b)), so that only either neighbourhood relations in time (procedure a)) or in space (procedure b)) are taken into account. Neighbourhood relations of the respective second dimension (information from space or time) are given up accordingly in the entropy calculation, which is a major limitation. The *(simultaneous) co-movement pattern of several variables over time* is important information that should be encoded in a mathematical representation, however. For example, each ECG and arterial blood pressure (ABP) signal contains information about cardiac status, which can be used for the diagnosis of diseases. Depolarisation of ventricles and contraction of the large ventricular muscles of the human heart (so-called QRS complex) can be observed during an ECG signal by the highest rash, the most visually striking part of the upper recording II in Fig. 4.3. Shortly after the electrical activity, the blood pressure rises, as shown in the second signal in Fig. 4.3. Both movements of the spatial variables depend on each other and should not be considered separately in an analysis, but in co-existence (red boxes in Fig. 4.3).

Procedure c) allows both spatial and temporal information to be taken into account by first encoding the spatial information of all m variables in a single dimension and then taking the temporal neighbourhood relationship into account in the univariate entropy calculation. MPE-EUCL, MPE-MANH and MPE-NORM consider spatial information in the form of spatial distances, but do not consider that the spatial variables may be correlated. Therefore, we propose to perform dimensionality reduction using principal component analysis (PCA), a well-known method converting a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables by an orthogonal transformation, in the next chapter.

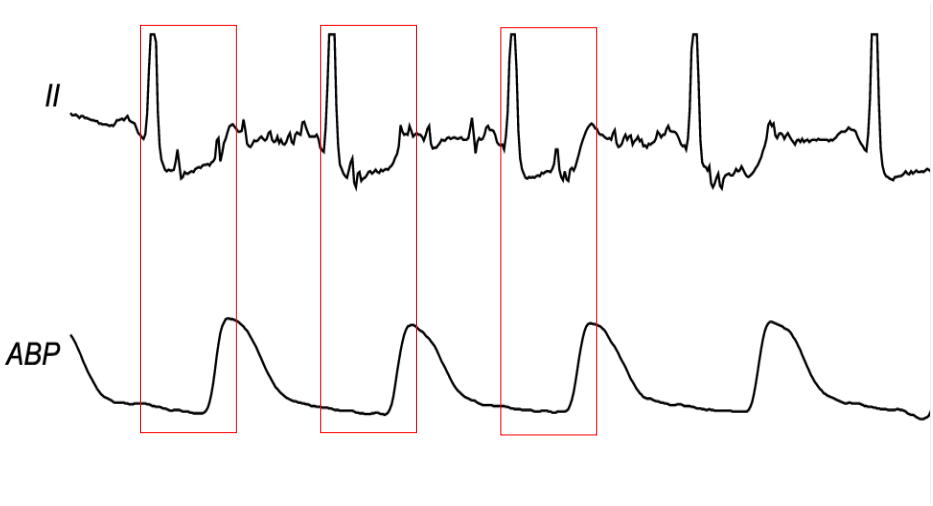


Figure 4.3: Medical multivariate time series from a patient of the MIMIC III waveform database with identification id 3900006_0029 published by Johnson *et al.* (2016).

Although possible correlations of the spatial variables are involved in procedure c), the ordering properties of neighbouring spatial variables still do not play a role. To ensure this, univariate ordinal patterns must be generalised to the multivariate case as already motivated with procedure d). In the next chapter, we introduce a corresponding generalisation.

Chapter 5

New Approaches in Multivariate Ordinal Pattern Representations

In this chapter, we introduce new approaches to multivariate variants of permutation entropy. In **Section 5.1** we fill the gap in the field of canonical extensions by introducing multivariate weighted permutation entropy. The idea is first presented in the following conference paper.

> Marisa Mohr, Florian Wilhelm, and Ralf Möller. On the Behaviour of Weighted Permutation Entropy on Fractional Brownian Motion in the Univariate and Multivariate Setting. *The International FLAIRS Conference Proceedings*, 34, 2021

The canonical variants investigate various aspects such as the influence of scales or amplitudes, but no correlations of spatial variables. Thus, in **Section 5.2** we present a new approach that allows for decorrelation of spatial variables before computing MPE. In **Section 5.3** we introduce another approach by naturally extending the definition of univariate ordinal patterns to the multivariate domain to account for both variable and temporal dependencies. Both approaches allow for the consideration of interdependencies between spatial variables in different ways and are presented for the first time in the following conference paper.

> Marisa Mohr, Florian Wilhelm, Mattis Hartwig, Ralf Möller, and Karsten Keller. New Approaches in Ordinal Pattern Representations for Multivariate Time Series. In *Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference (FLAIRS-33)*, pages 124–129, 2020

Note that the approach to be introduced in Section 5.2 is furthermore deepened and generalised in the following article.

> Marisa Mohr and Ralf Möller. Ordering Principal Components of Multivariate Fractional Brownian Motion for Solving Inverse Problems. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2021 (APSIPA-ASC)*, 2021

Algorithm 3: Computation of MWPE

Input: Multivariate Time Series $X^{m\times T}$, Order d, Delay τ
Function `weightedPooling(`X`,`d`,`τ`):`
  $P^{m\times d!} \leftarrow$ weighted pooling matrix initialised with zeros
  for *every time series variable* $i = 1, \ldots, m$ *in* X **do**
    for *each ordinal pattern* $j = 1, \ldots, d!$ **do**
      $w_{ij} \leftarrow$ add up weights w_t `// see Equation (3.21)`
      $p_{ij} \leftarrow$ divide w_{ij} by the total sum of all $m \cdot \delta$ weights
  return P
MWPE $\leftarrow$ PE (`marginalisation(weightedPooling(`X`,`d`,`τ`))`) `// see Alg. 1`

5.1 Multivariate Weighted Permutation Entropy (MWPE)

In analogy of PE and PPE or MSPE and MMSPE, we provide a canonical definition of multivariate weighted permutation entropy (MWPE) based on WPE. Again, an auxiliary matrix has to be established first for the determination of MWPE, but this time using weights:

1. For each variable $i = 1, .., m$ and for each ordinal pattern $j = 1, ..., d!$, select all time steps $s \in [\tau(d-1)+1, T]$, for which the variable-time pair (i, s) has univariate ordinal pattern j.

2. Add up the weights w_t, i.e., $\mathbf{w_{ij}} = \sum_{t=d\tau-\tau+1}^{T} w_t$ for all selected ordinal pattern vectors j and for each variable $i = 1, ..., m$. Note that the total count of weights w_t^i for each variable i is $\delta := T - (d\tau - \tau)$.

3. Divide the weighted sum $\mathbf{w_{ij}}$ by the total sum of all $m \cdot \delta$ weights to obtain the weighted frequencies for every ordinal pattern j.

4. Store the results, i.e., weighted frequencies $p_{w_{ij}}^{d,\tau}$ in a so-called *weighted pooling matrix* $P_w^{d,\tau} \in \mathbb{R}^{m\times d!}$, which reflects the weighted distribution of ordinal patterns in the multivariate time series across its m variables.

Based on the weighted pooling matrix $P_w^{d,\tau}$, MWPE is defined as follows.

Definition 5.1.1 (MWPE)**.** *Multivariate weighted permutation entropy* (MWPE) of a multivariate time series $\mathbf{X} = ((x_t^i)_{i=1}^m)_{t=1}^T$ is defined as PE of the marginal weighted frequencies $p_{w_{\cdot j}}^{d,\tau} = \sum_{i=1}^m p_{w_{ij}}^{d,\tau}$ for $j = 1, ..., d!$ and is defined by

$$\mathrm{MWPE}_{d,\tau}(\mathbf{X}) = -\sum_{j}^{d!} p_{w_{\cdot j}}^{d,\tau} \ln p_{w_{\cdot j}}^{d,\tau}, \tag{5.1}$$

where $p_{w_{ij}}^{d,\tau}$ is the weighted frequency of ordinal pattern j in variable i.

Algorithm 3 provides pseudocode for computing MWPE.

5.2 Multivariate Permutation Entropy Based on Principal Components (MPE-PCA)

As introduced in Section 4.1.3, one way to realise MPE is to first transform a multivariate time series $((x_t^i)_{i=1}^m)_{t=1}^T$ into a univariate time series $(x'_t)_{t=1}^T$ and then calculate the PE from Definition 3.3.2 on the reduced time series as usual. To account for possible correlations between spatial variables, we propose a decorrelation by principal component analysis and use the (first) principal component to calculate PE.

5.2.1 Principal Component Analysis

Principal component analysis (PCA) converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables by an orthogonal transformation. For m-dimensional data X, there are basically m basis vectors that are orthogonal. The variance of data points along each basis vector is the total variance of the data. In particular, applying PCA to centred data $X \in \mathbb{R}^{m\times T}$ means finding a linear mapping $V \in \mathbb{R}^{m\times m}$ onto a new decorrelated representation

$$Z = V^T X \in \mathbb{R}^{m\times T}, \tag{5.2}$$

such that the variance of the projected data

$$\mathrm{Var}(Z) = \frac{1}{n-1}\sum_i \|V^T x_i\|^2 \tag{5.3}$$

is maximal. Then, to find the direction v_1 of maximum variance, we need to solve

$$\max_{v_1} \frac{1}{n-1}\|v_1^T X\|^2 \qquad \text{s.t. } v_1^T v_1 = 1. \tag{5.4}$$

Rewriting the objective, we have

$$\frac{1}{n-1}\|v_1^T X\|^2 = \frac{1}{n-1} v_1^T X X^T v_1 = v_1^T \Sigma v_1, \tag{5.5}$$

where Σ is the covariance matrix of X. Maximisation under constraint $v_1^T v_1 = 1$ means solving the Lagrangian

$$f(v_1) = v_1^T \Sigma v_1 - \lambda_1(v_1^T v_1 - 1) \tag{5.6}$$

with its derivative

$$\mathrm{d}f(v_1) = 2(v_1^T\Sigma - \lambda_1 v_1^T)\mathrm{d}v_1. \tag{5.7}$$

Setting the derivative to zero implies

$$\Sigma v_1 = \lambda_1 v_1. \tag{5.8}$$

From Equation (5.8) we see that v_1 must be an eigenvector of Σ for the largest eigenvalue. In general, PCA is thus based on eigenvalue analysis, i.e., an eigenvector of the covariance matrix $\mathrm{Cov}(X) = \Sigma$ corresponds to a basis vector. Further details can be found in several statistical textbooks, e.g., by Hastie *et al.* (2009).

5.2.2 Dimensionality Reduction

After performing PCA, the full representation $Z \in \mathbb{R}^{m \times T}$ contains m eigenvectors, also called principal components (PCs), which are decorrelated, with the first PC capturing most of the variance in the data. Depending on the application, it is helpful to map the m-dimensional data into low-dimensional data without losing too much information, as this may facilitate a classification or regression problem. For example, data compression can reduce processing time and storage space. Moreover, dimensionality reduction can improve the performance of ML models in terms of avoiding the curse of dimensionality or enable visualisation (2D or 3D data is easier to visualise and interpret). Based on PCA, dimension reduction ignores the dimensions that capture low variance in the data (and thus represent a kind of noise). It is assumed that if the first $r < m$ basis vectors cover a sufficiently large proportion of the total variance, the new r basis vectors are sufficient for the information content of the data. Keeping only the first r PCs of the data X gives the truncated transformation $Z_{1\dots r} = V_r^T X$, where $V \in \mathbb{R}^{m \times r}$ is a matrix of weights whose columns are the eigenvectors of Σ sorted in descending order of the r highest corresponding eigenvalues and is used for dimensionality reduction.

Analogous to procedure c) introduced in Section 4.1.3, a single-dimensional representation is now used in the context of the entropy calculation, as this allows the original univariate Definition 3.3.2 of PE to be used. Since the first PC captures the greatest variance and thus the greatest information in the data, $r = 1$ is chosen. Depending on the application, however, other individual PCs $Z^i \in \mathbb{R}^{1 \times T}$ with $i \in \{1, \dots, m\}$ or combinations of PCs $Z_{1\dots r} \in \mathbb{R}^{r \times T}$ with $r \leq m$ may be of interest for the analysis. Note that Z^i is a vector, while $Z_{1\dots r}$ is a matrix.

5.2.3 Calculation of MPE

Multivariate permutation entropy based on principle component analysis (MPE-PCA) can intuitively be understood as a two-step approach, firstly using PCA to transform the data into a decorrelated (lower) dimensional representation, before secondly applying well-known methods such as PE or PPE.

In the case of an individual PC $Z^i \in \mathbb{R}^{1 \times T}$, MPE-PCA is defined as follows. Note that $i = 1$ is most reasonable.

Algorithm 4: Computation of MPE-PCA/PPE-PCA

Input: Multivariate Time Series $X^{m\times T}$, where X is centred, i.e., mean zero, Order d, Delay τ, Principal Component (i, r) with $i \leq r$

Function `PCA(`X`):`
- $\Sigma^{m\times m} \leftarrow$ compute covariance matrix of X
- $V\Lambda V^{-1} \leftarrow$ compute eigendecomposition of Σ with $V = (v_1, \ldots, v_m)$ and $\lambda_1 \leq \lambda_m$
- $Z^{m\times T} \leftarrow$ compute orthogonal transformation `// see Eq. (5.2)`
- **return** Z

$Z^{m\times T} \leftarrow$ `PCA(`X`)`
if $r == i$ **then**
- **return** PE(Z^i) `// see Definition 3.3.2`

return PPE$(Z_{1\ldots r})$ `// see Alg. 1`

Definition 5.2.1 (MPE-PCA)**.** The *multivariate permutation entropy based on principle component analysis* (MPE-PCA) of order $d \in \mathbb{N}$ and delay $\tau \in \mathbb{N}$ of a multivariate time series $((x_t^i)_{i=1}^m)_{t=1}^T$, $T \in \mathbb{N}$ is defined as

$$\text{MPE-PCA}_{d,\tau,i}(Z^i) = -\sum_{j=1}^{d!} p_j^{d,\tau} \ln p_j^{d,\tau}, \tag{5.9}$$

where $p_j^{d,\tau}$ is the frequency of ordinal pattern j in the i-th principal component $Z^i \in \mathbb{R}^{1\times T}$.

Depending on the application, the first or a single PC may not describe "enough" variance of the data, such that the first $r \in \{2, \ldots, m\}$ PCs $Z_{1\ldots r} \in \mathbb{R}^{r\times T}$ are considered for calculation. For this case, we define a pooled version for MPE calculation as follows.

Definition 5.2.2 (PPE-PCA)**.** The *pooled permutation entropy based on principal component analysis* (PPE-PCA) of a multivariate time series $X = ((x_t^i)_{i=1}^m)_{t=1}^T$ is defined as PE of the marginal frequencies $p_{.j}^{d,\tau} = \sum_{i=1}^m p_{ij}^{d,\tau}$ for $j = 1, ..., d!$ of the first $r \leq m$ principal components $Z_{1\ldots r} \in \mathbb{R}^{r\times T}$ and given by

$$\text{PPE-PCA}_{d,\tau}(Z_{1\ldots r}) = -\sum_{j}^{d!} p_{.j}^{d,\tau} \ln p_{.j}^{d,\tau}. \tag{5.10}$$

Algorithm 4 provides pseudocode for computing MPE-PCA or PPE-PCA.

Ma *et al.* (2021) recently already applied MPE-PCA to the complexity analysis of EEG data. The authors show that a person has higher complexity during a mental arithmetic task as measured by MPE-PCA than before carrying out the task. In addition, they also confirm the necessity of dimensionality reduction via PCA.

$$\begin{array}{cccccc}
\binom{2\,1\,0}{2\,1\,0} & \binom{2\,1\,0}{0\,1\,2} & \binom{2\,1\,0}{1\,2\,0} & \binom{2\,1\,0}{0\,2\,1} & \binom{2\,1\,0}{2\,0\,1} & \binom{2\,1\,0}{1\,0\,2} \\
\binom{0\,1\,2}{2\,1\,0} & \binom{0\,1\,2}{0\,1\,2} & \binom{0\,1\,2}{1\,2\,0} & \binom{0\,1\,2}{0\,2\,1} & \binom{0\,1\,2}{2\,0\,1} & \binom{0\,1\,2}{1\,0\,2} \\
\binom{1\,2\,0}{2\,1\,0} & \binom{1\,2\,0}{0\,1\,2} & \binom{1\,2\,0}{1\,2\,0} & \binom{1\,2\,0}{0\,2\,1} & \binom{1\,2\,0}{2\,0\,1} & \binom{1\,2\,0}{1\,0\,2} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\binom{1\,0\,2}{2\,1\,0} & \binom{1\,0\,2}{0\,1\,2} & \binom{1\,0\,2}{1\,2\,0} & \binom{1\,0\,2}{0\,2\,1} & \binom{1\,0\,2}{2\,0\,1} & \binom{1\,0\,2}{1\,0\,2}
\end{array}$$

Figure 5.1: All $(d!)^m = 36$ possible MOPs of order $d = 3$ with $m = 2$ variables.

5.3 Multivariate Ordinal Pattern Permutation Entropy (MOPPE)

Previous approaches are not able to encode the joint movement of several spatial variables over one or more points in time. The intuitive idea of the following approach is to store the univariate ordinal patterns of all spatial variables at a fixed time point t together into one multivariate pattern. Thus, an extension of univariate ordinal pattern to multidimensionality is given. Note that some theoretical basis for this is set by Keller (2012) and Antoniouk *et al.* (2014).

Definition 5.3.1 (MOP)**.** A matrix $(x_1, ..., x_d) \in \mathbb{R}^{m\times d}$ is associated with *multivariate ordinal pattern* (MOP)

$$\begin{pmatrix} r_{11} & \cdots & r_{1d} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{md} \end{pmatrix} \in \mathbb{N}^{m\times d} \tag{5.11}$$

of order $d \in \mathbb{N}$ if $x_{r_{i1}} \geq ... \geq x_{r_{id}}$ for all $i = 1, ..., m$ and $r_{il-1} > r_{il}$ in the case $x_{r_{il-1}} = x_{r_{il}}$.

Figure 5.1 shows all possible MOPs of order $d = 3$ and number of variables $m = 2$. With the natural extension of Definition 3.3.1 to the multivariate case, it is possible to apply the PE algorithm from Definition 3.3.2 in its original form to multivariate time series.

Definition 5.3.2 (MOPPE)**.** The *multivariate ordinal pattern permutation entropy* (MOPPE) of order $d \in \mathbb{N}$ and delay $\tau \in \mathbb{N}$ of a multivariate time series $\mathbf{X} = ((x_t^i)_{i=1}^m)_{t=1}^T$ is defined by

$$\mathrm{MOPPE}_{d,\tau}(\mathbf{X}) = -\sum_{j=1}^{d!} p_j^{d,\tau} \ln p_j^{d,\tau}, \tag{5.12}$$

where $p_j^{d,\tau}$ is the frequency of MOP j in the multivariate time series $\mathbf{X}$.

Algorithm 5: Computation of MOPPE

Input: Multivariate Time Series $X^{m \times T}$, Order d, Delay τ

$p^{1 \times d!} \leftarrow$ frequencies initialised with zeros
for *every time step* t *in* X **do**
  for *each ordinal pattern* $j = 1, \ldots, d!$ **do**
    $c \leftarrow$ count all time points t with MOP j of order d and delay τ
     `// see Definition 5.3.1`
    $p_j \leftarrow$ divide c by the number of total time steps $T - \tau(d-1)$ in X
MOPPE $\leftarrow$ PE(p) `// see Definition 3.3.2`

Algorithm 5 provides pseudocode for computing MOPPE. Note that all corresponding algorithms for computing MWPE, MPE-PCA, PPE-PCA, and MOPPE can be found on Github[1] and Python Package Index (PyPI)[2].

The number of the possible MOPs increases exponentially with the number of variables m, i.e., $(d!)^m$. Therefore, if d and m are too large, depending on the application, each pattern occurs only rarely or some not at all, resulting in a uniform distribution of ordinal patterns. This has the consequence that subsequent learning procedures may fail in prediction because the representation does not discriminate correctly. Nevertheless, for small order d and sufficiently large length T of the time series, the use of multivariate ordinal patterns can lead to higher accuracy in learning tasks as we show in Chapters 12 and 13 as the representation encode information of interdepencies of several spatial variables. In addition, the challenge of combinatorial possibilities of MOPs can be overcome by reducing the dimensionality of the multivariate time series beforehand, e.g., by PCA. Thus, the advantages of MPE-PCA and MOPPE can be used simultaneously.

[1] https://github.com/marisamohr/mpePy
[2] https://pypi.org/project/mpePy

Part II

Multivariate Permutation Entropy Applied to Multivariate Fractional Brownian Motion

Time-dependent real-world applications involving *long-range dependence* or *self-similarity* are usually modelled by multivariate fractional Brownian motion (mfBm). Examples of mfBm can be found in genetic sequences (Arianos and Carbone, 2009), financial time series (Fleming *et al.*, 2001; Gil-Alana, 2003; Davidson and Hashimzade, 2008; Alvarez, 2021), or functional magnetic resonance imaging of several brain regions (Achard *et al.*, 2008; Rabiei *et al.*, 2021). When modelling, analysing or predicting in relation to mfBm, the study of the qualitative behaviour of its observations in terms of specific characteristics, representations, or parameters is an important topic. For example, as motivated in Section 2.1, by estimating a model parameter from a few observations, the inverse problem can be solved, i.e., (information about) the generating dynamical system that produces the observations can be derived. In the specific case of mfBm, the solution of the inverse problem involves the estimation of the self-similarity or Hurst parameter H.

In this second part, the qualitative behaviour of mfBm is investigated by analysing numerous MPE-variants in variation of the Hurst parameter H. Several theoretical relationships between MPE and the Hurst parameter H of mfBm are derived and constitute contributions 2a-e of this dissertation. The theoretical results can be used, for example, to solve the inverse problem. A summary of all contributions (except for contribution 2d, i.e., the study on MPE-PCA) is first published in the following journal article.

> Marisa Mohr and Ralf Möller. A Summary of Canonical Multivariate Permutation Entropies on Multivariate Fractional Brownian Motion. *Advances in Science, Technology and Engineering Systems Journal*, 6(5):107–124, 2021

We start with individual investigations of the behaviour of the existing canonical approaches PPE, MMSPE and the newly introduced approaches MWPE, MPE-PCA and MOPPE when applied to mfBm in the following Chapters 6 to 10. We then compare all representations and discuss differences as well as possible applications in Chapter 11.

Chapter 6

Pooled Permutation Entropy Applied to Multivariate Fractional Brownian Motion

This chapter investigates the qualitative behaviour of PPE on mfBm in variation of its Hurst parameter H and additional parameters. We first presented this idea in the following conference paper.

> Marisa Mohr, Nils Finke, and Ralf Möller. On the Behaviour of Permutation Entropy on Fractional Brownian Motion in a Multivariate Setting. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2020 (APSIPA-ASC)*, pages 189–196, 2020

We begin with presenting related work in the univariate context needed for the work to be self-consistent. The remainder of this chapter provides contribution 2a) of this work. For this, we provide a theoretical analysis of the behaviour of PPE on mfBm in variation of its Hurst parameter H, before underpinning the results in an experimental setting. We end with an interim conclusion.

6.1 The Univariate Case: PE of fBm

To study the behaviour of PPE on mfBm, we first recapitulate univariate results of Bandt and Shiha (2007), who studied univariate ordinal patterns in the context of Gaussian and autoregressive moving-average processes. Corresponding proofs can be looked up in the original work.

6.1.1 Distribution of Ordinal Pattern Symbols in fBm

We summarise the distribution of univariate ordinal patterns in fBms according to its orders $d = 2, \ldots, 5$.

Order $d = 2$. The ordinal patterns of order $d = 2$ in fBms are equally distributed, more specifically

$$p_{12}^{\tau} = p_{21}^{\tau} = 1/2 \tag{6.1}$$

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

Order $d = 3$. The distribution of ordinal patterns of order $d = 3$ in fBms is given by

$$p^\tau_{123} = \frac{1}{\pi} \cdot \arcsin 2^{H-1} =: u \tag{6.2}$$

for all $\tau \in \mathbb{N}$. In particular, the distribution of ordinal patterns is monotonically dependent on the Hurst parameter H. In addition, for all $\tau \in \mathbb{N}$ it holds that

$$p^\tau_j = \begin{cases} u & \text{if } j = (123), (321), \\ (1-2u)/4 & \text{otherwise.} \end{cases} \tag{6.3}$$

Note that Eq. (6.3) applies to arbitrary Gaussian processes with stationary increments.

Order $d = 4$. The distribution of ordinal patterns of order $d = 4$ and all delays $\tau \in \mathbb{N}$ can also be expressed, albeit in a more complex formula

$$p^\tau_j = \frac{1}{8} + \frac{1}{4\pi} \cdot v_j, \tag{6.4}$$

where $v_j =$

$$\begin{cases} \arcsin \alpha_1 + 2 \arcsin \alpha_2 & \text{if } j = (1234), (4321), \\ \arcsin \alpha_4 - 2 \arcsin \alpha_5 & \text{if } j = (4231), (1324), \\ 2 \arcsin \alpha_6 + \arcsin \alpha_1 & \text{if } j = (2143), (3412), \\ \arcsin \alpha_7 - \arcsin \alpha_1 - \arcsin \alpha_5 & \text{if } j = (1243), (2134), (3421), (4312), \\ \arcsin \alpha_7 - \arcsin \alpha_4 - \arcsin \alpha_5 & \text{if } j = (1423), (4132), (3241), (2314), \\ \arcsin \alpha_3 + \arcsin \alpha_8 - \arcsin \alpha_5 & \text{if } j = (3124), (1342), (4213), (2431), \\ \arcsin \alpha_6 - \arcsin \alpha_8 + \arcsin \alpha_2 & \text{if } j = (1432), (4123), (2341), (3214), \end{cases} \tag{6.5}$$

with

$$\begin{aligned} \alpha_1 &= \frac{1 + 3^{2H} - 2^{2H+1}}{2}, & \alpha_2 &= 2^{2H-1} - 1, & \alpha_3 &= \frac{1 - 3^{2H} - 2^{2H}}{2 \cdot 6^H}, \\ \alpha_4 &= \frac{3^{2H} - 1}{2^{2H+1}}, & \alpha_5 &= 2^{2H-1}, & \alpha_6 &= \frac{-1 - 3^{2H} + 2^{2H}}{2 \cdot 3^H}, \\ \alpha_7 &= \frac{3^{2H} - 2^{2H} - 1}{2^{2H+1}}, & & \text{and} & \alpha_8 &= \frac{2^{2H-1}}{3^H}. \end{aligned} \tag{6.6}$$

Order $d \geq 5$. For the distribution of patterns of order $d = 5$ or greater, no closed formulas exist.

6.1.2 PE of fBm

The distribution of ordinal patterns of different orders d in fBms results in the following behaviour of PE when applied to fBm.

Theorem 6.1.1. *For order $d = 2$ and $H \in (0, 1)$, it holds*

$$\mathrm{PE}_{2,\tau}(B_H(t)) = -\ln(1/2) \tag{6.7}$$

for all delays $\tau \in \mathbb{N}$.

Proof. Follows directly by inserting Eq. (6.1) into Definition 3.3.2 of PE. □

Theorem 6.1.2. *For orders $d = 3$ and $d = 4$, $\mathrm{PE}_{d,\tau}(B_H(t))$ is independent of delays $\tau \in \mathbb{N}$ but monotonically dependent on the Hurst parameter $H \in (0, 1)$, i.e.,*

$$\mathrm{PE}_{3,\tau}(B_H(t)) = -(2 \cdot u \cdot \ln(u) + (1 - 2u) \cdot \ln((1 - 2u)/4) \tag{6.8}$$

$$\mathrm{PE}_{4,\tau}(B_H(t)) = -\sum_{j=1}^{d!} \frac{1}{8} + \frac{1}{4\pi} \cdot v_j \cdot \ln\left(\frac{1}{8} + \frac{1}{4\pi} \cdot v_j\right) \tag{6.9}$$

with u from Eq. (6.2) and v_j from Eq. (6.5) for all delays $\tau \in \mathbb{N}$.

Proof. Follows directly by inserting Eqs. (6.2) and (6.3) or Eqs. (6.4) and (6.5) into Definition 3.3.2 of PE. The monotonic dependence on the Hurst parameter H of the individual distributions of the ordinal patterns transfers to the PE calculation, since addition or multiplication with scalars does not change monotonicity. □

6.2 The Multivariate Case: Theoretical Analysis

Based on the existing work of the last section, we derive in the following contribution 2a) of this work. The univariate case, i.e., the distribution of univariate ordinal patterns in fBm and hence the behaviour of PE of fBm, is well understood and provided in detail the previous section. Since the idea of PPE is to pool the distribution of univariate ordinal patterns of each spatial variable into a multivariate measure, the results of Section 6.1 can be applied to the multivariate case as follows.

In analogy to the univariate case, PPE of order $d = 2$ is constant for all delays τ and Hurst parameters $H \in (0, 1)$ of mfBm. In addition, it is independent of all numbers of variables m of mfBm.

Theorem 6.2.1. *For order $d = 2$, it holds*

$$\mathrm{PPE}_{2,\tau}(\mathbf{B}_H^m(t)) = -\ln(1/2) \tag{6.10}$$

for all delays $\tau \in \mathbb{N}$, Hurst parameters $H \in (0, 1)^m$, and number of variables $m \in \mathbb{N}$ of mfBm.

Proof. To determine the frequencies of ordinal patterns in the pooling matrix P, univariate ordinal patterns per variable $i = 1, \dots, m$ are used. Since mfBm $\mathbf{B}_H^m(t)$ is marginally an fBm, with Eq. (6.1) the distribution of univariate ordinal patterns of order $d = 2$ is equally distributed. For length $T \to \infty$ the distribution of ordinal patterns is independent of T and delay τ, such that the frequencies of the pooling matrix P are still equally distributed, i.e.,

$$p_{ij}^{2,\tau} = p_{ij}^{2,\tau} = \frac{1}{2m} \tag{6.11}$$

for all $i = 1, .., m$ and $j = (0,1), (1,0)$. When calculating the marginal frequencies of the pooling matrix P, the number of variables m is reduced again, that is

$$p_{\cdot j}^{2,\tau} = \frac{1}{2} \tag{6.12}$$

for $j = (0,1), (1,0)$. Hence the claim follows by inserting in Eq. (4.1). □

Theorem 6.2.2. *For orders $d = 3, 4$, $\mathrm{PPE}_{d,\tau}(\mathbf{B}_H^m(t))$ is independent of all delays $\tau \in \mathbb{N}$, but monotonically dependent on the number of variables $m \in \mathbb{N}$ and on the Hurst parameter $H \in (0,1)^m$ of mfBm.*

Corollary 6.2.1. *If $H_i = H_j$ for all i, j, then $\mathrm{PPE}_{d,\tau}(\mathbf{B}_H^m(t))$ is independent of the number of variables $m \in \mathbb{N}$ of mfBm.*

We prove Thm. 6.2.2 and Cor. 6.2.1 together.

Proof. Since mfBm $\mathbf{B}_H^m(t)$ is marginally an fBm, by analogy with the case of order $d = 2$, with Eqs. (6.2) and (6.3) the frequencies of ordinal patterns of order $d = 3$ in each variable $i = 1, ..., m$ in the pooling matrix P are given by

$$p_{ij}^{3,\tau} = \begin{cases} \frac{1}{m\pi} \arcsin 2^{H_i - 1} & \text{if } j = (123), (321) \\ \frac{1}{4m}(1 - \frac{2}{\pi} \arcsin 2^{H_i - 1}) & \text{otherwise} \end{cases} \tag{6.13}$$

for all delays τ.

With the same argumentation as above but with Eq. (6.4), the frequencies of ordinal patterns of order $d = 4$ in each variable $i = 1, ..., m$ in the pooling matrix P are given by

$$p_{ij}^{\tau,4} = \frac{1}{m}\left(\frac{1}{8} + \frac{1}{4\pi} \cdot v_j\right) \tag{6.14}$$

with v_j as in Eq. (6.5), while

$$\begin{aligned} \alpha_1 &= \frac{1 + 3^{2H_i} - 2^{2H_i+1}}{2}, & \alpha_2 &= 2^{2H_i-1} - 1, & \alpha_3 &= \frac{1 - 3^{2H_i} - 2^{2H_i}}{2 \cdot 6^{H_i}}, \\ \alpha_4 &= \frac{3^{2H_i} - 1}{2^{2H_i+1}}, & \alpha_5 &= 2^{2H_i-1}, & \alpha_6 &= \frac{-1 - 3^{2H_i} + 2^{2H_i}}{2 \cdot 3^{H_i}}, \\ \alpha_7 &= \frac{3^{2H_i} - 2^{2H_i} - 1}{2^{2H_i+1}}, & & \text{and} & \alpha_8 &= \frac{2^{2H_i-1}}{3^{H_i}} \end{aligned} \tag{6.15}$$

for all delays τ. In particular, $\mathrm{PPE}_{d,\tau}(\mathbf{B}_H^m(t))$ stays independent of all delays $\tau \in \mathbb{N}$.

For the calculation of marginal frequencies, we distinguish two cases:

I) If $H_i \neq H_l$ for all variables $i = l$, the marginal frequencies are given by

$$p_{\cdot j}^{3,\tau} = \begin{cases} \sum_{i=1}^{m} \frac{1}{m\pi} \arcsin 2^{H_i - 1} & \text{if } j = (0,1,2), (2,1,0) \\ \sum_{i=1}^{m} \frac{1}{4m} (1 - \frac{2}{\pi} \arcsin 2^{H_i - 1}) & \text{otherwise} \end{cases} \tag{6.16}$$

or

$$p_{\cdot j}^{\tau,4} = \sum_{i=1}^{m} \frac{1}{m} \left(\frac{1}{8} + \frac{1}{4\pi} \cdot v_j \right), \tag{6.17}$$

where v_j is as in Eq. (6.5) and $\alpha_1, \ldots, \alpha_8$ as in Eq. (6.15) for all delays τ.

II) If $H_i = H_l := H$ for all variables i and l, the number of variables m is reduced again and the marginal frequencies are the same as the frequencies of univariate ordinal pattern in Eqs. (6.2) to (6.4), i.e.,

$$p_{\cdot j}^{3,\tau} = \begin{cases} \frac{1}{\pi} \arcsin 2^{H-1} & \text{if } j = (0,1,2), (2,1,0) \\ \frac{1}{4} (1 - \frac{2}{\pi} \arcsin 2^{H-1}) & \text{otherwise} \end{cases} \tag{6.18}$$

or

$$p_{\cdot j}^{\tau,4} = \left(\frac{1}{8} + \frac{1}{4\pi} \cdot v_j \right), \tag{6.19}$$

where v_j as in Eq. (6.5) and $\alpha_1, \ldots, \alpha_8$ as in Eq. (6.6) for all delays τ. In particular, they are independent of all numbers of variables m of mfBm.

Monotonic dependence of PPE on the Hurst parameter H_i is also preserved since the transformations carried out, i.e., additions or multiplication with scalars, do not change the monotonicity. □

For orders $d \geq 5$, no closed formulas for the distribution of ordinal patterns in mfBm exist (Bandt and Shiha, 2007). Nevertheless, similar behaviour can be observed in this case, which we evaluate in an experimental setting in the next section.

The consistency concerning the univariate case, i.e., constant PPE for $d = 2$ and monotonic dependence for $d \geq 3$, is not surprising but is related to the computational logic of PPE. PPE can be understood as a canonical extension of the univariate ordinal patterns into a multivariate variant, all univariate ordinal patterns of all variables combined into one measure. Simply put, the variables are more or less appended one after the other and considered as a univariate time series.

However, this simple approach has major advantages. The monotonic dependence of PPE on the Hurst parameter H can be exploited for its estimation. Thus, if one calculates the value of PPE from mfBm, this provides information about the level of the Hurst parameter H via the direct dependency, e.g. in Eqs. (6.16) to (6.19). Furthermore, based on the pooled matrix P and with a small extension of the PPE algorithm (see Section 4.1.1), the univariate PE of the individual variables can also be determined, which provide information about Hurst parameters H_i of the individual variables $i = 1, \ldots, m$. Further advantages and disadvantages, especially in comparison with other representations, are discussed in Chapter 11.

6.3 Experimental Evaluation

In this experimental evaluation, we investigate the behaviour of PPE on mfBm in variation of its Hurst parameter H, the number of variables m and the delay τ underpinning the theoretical results derived in the previous section. Even though fBm and mfBm are continuous processes, only sampled data can be used in practical applications. This sampling leads to a discrete-time version of fBm and mfBm with a sample path of length $T \to \infty$, which we do not refer to separately.

We first introduce the experimental setup before discussing the results.

6.3.1 Experimental Setup

All experimental computations are based on simulations of mfBms using an algorithm according to Lemma 3.2.1 implemented by Amblard *et al.* (2013)[1]. For the simulation of mfBms, the correlation coefficient ρ_{ij} are all set to 0.0 (if not explicitly mentioned otherwise), resulting in a correlation matrix

$$\begin{pmatrix} 1 & 0.0 & \ldots & 0.0 \\ 0.0 & 1 & \ldots & \vdots \\ \vdots & \vdots & \ddots & 0.0 \\ 0.0 & \ldots & 0.0 & 1 \end{pmatrix} \in \mathbb{R}^{m \times m}. \tag{6.20}$$

[1]Thanks to Jean-François Coeurjolly for providing corresponding code.

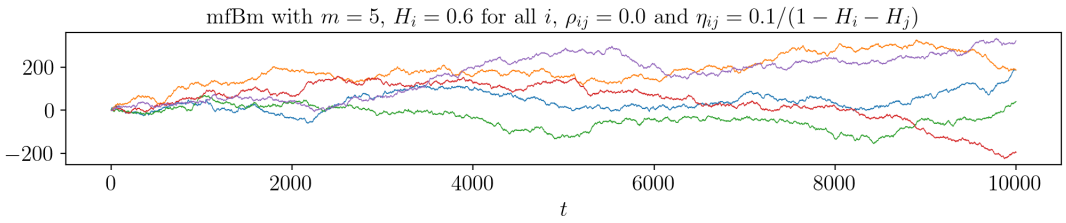


Figure 6.1: Exemplary Simulation of mfBm.

Time-reversibility parameters η_{ij} are set to $0.1/(1 - H_i - H_j)$, resulting in a matrix

$$\begin{pmatrix} 0 & \frac{0.1}{1-H_i-H_j} & \cdots & \frac{0.1}{1-H_i-H_j} \\ \frac{-0.1}{1-H_i-H_j} & 0 & \vdots & \vdots \\ \vdots & \vdots & \ddots & \frac{0.1}{1-H_i-H_j} \\ \frac{-0.1}{1-H_i-H_j} & \cdots & \frac{-0.1}{1-H_i-H_j} & 0 \end{pmatrix} \in \mathbb{R}^{m \times m}. \tag{6.21}$$

Note that the setting of the time-reversibility parameters η_{ij} corresponds to the case where mfBm is well-balanced. Since we do not go into detail about the effects of the time-reversibility parameter in this work, for further information we refer to the works of Stoev and Taqqu (2006); Coeurjolly *et al.* (2010). Furthermore, the length $T = 10,000$ of mfBms is assumed to be large. An example of a simulation of mfBm with $m = 5$ variables, Hurst parameters $H_i = 0.6$ for $i = 1, \ldots, 5$, and correlation coefficients ρ_{ij} and time-reversibility parameter η_{ij} as set above is shown in Figure 6.1. In the following experiment, we mainly consider different parameters H_i and m.

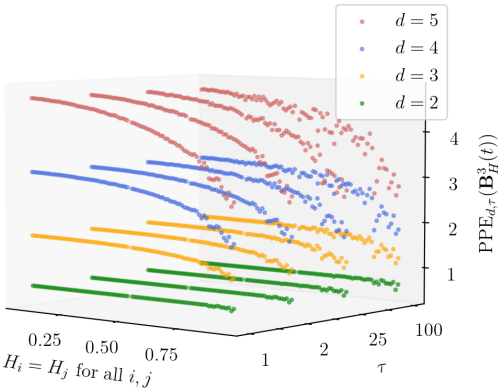
6.3.2 Experimental Results

In this experiment we compute PPE of different orders d and different delays τ for simulations of mfBm with different numbers of variables m and different Hurst parameters H_i for $i = 1, \ldots, m$. The results of the computations of PPE of orders $d = 2$ (green), $d = 3$ (orange), $d = 4$ (blue), and $d = 5$ (red) are visualised in Figure 6.2 and underpin the theorems derived in Section 6.2. In particular, we show the two following aspects:

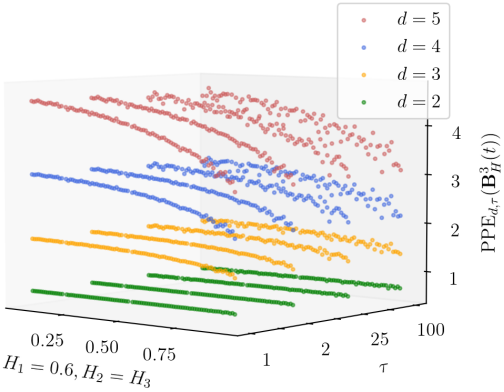
i) independence from the delay τ (Figs. 6.2a and 6.2b),

ii) (in)dependence from the number of variables m (Figs. 6.2c and 6.2d).

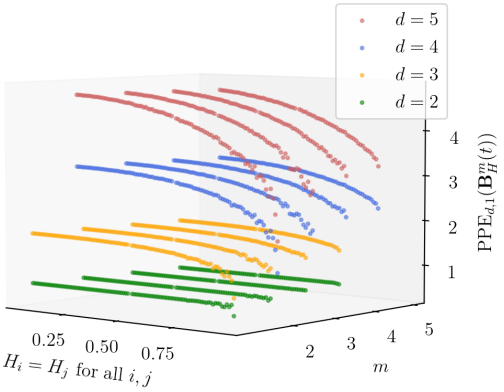
In general, all sub-figures of Fig. 6.2 show a constant PPE for order $d = 2$, namely $-\ln(1/2)$, while for orders $d > 2$ there is a monotonic dependence on the Hurst parameter H, i.e., for increasing Hurst parameter H the entropy decreases. In Figures 6.2a and 6.2b we show the independence of PPE on mfBm from the delay τ.



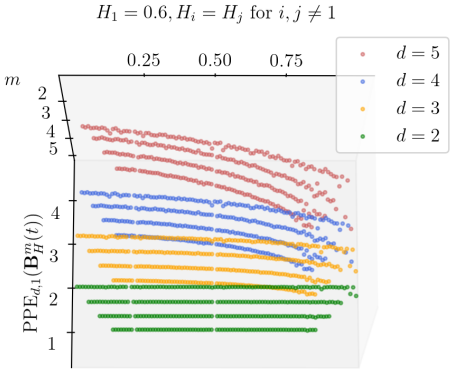
(a) $\tau \in \{1, 2, 25, 100\}$, $m = 3$, $H_i = H_j$.



(b) $\tau \in \{1, 2, 25, 100\}$, $m = 3$, $H_i \neq H_j$.



(c) $\tau = 1$, $m \in \{2, 3, 4, 5\}$, $H_i = H_j$.



(d) $\tau = 1$, $m \in \{2, 3, 4, 5\}$, $H_i \neq H_j$.

Figure 6.2: Experimental computations of PPE of orders $d = 2$ (green), $d = 3$ (orange), $d = 4$ (blue), and $d = 5$ (red) on mfBm in variation of its Hurst parameter $H \in (0, 1)^m$.

Here, the number of variables is set to $m = 3$. Again, all green lines in Figs. 6.2a and 6.2b, i.e., for $d = 2$, are constant for each τ, confirming the independence of delay τ in Eq. (6.10). All other lines, i.e., for $d > 2$, are also the same for any τ, but monotonically dependent on the Hurst parameter H, as given in Thm. 6.2.2. With a sharp eye it can be seen that the lines in Fig. 6.2a fall more sharply than in Fig. 6.2b. This is because in Fig. 6.2a the Hurst parameters H_i are the same, i.e. $H_i = H_j$ for all spatial variables i, j, and in Fig. 6.2b the Hurst parameters H_i are different, i.e. $H_1 = 0.6$ and $H_2 = H_3$. The proportion of variable $i = 1$ with Hurst parameter $H_1 = 0.6$ keeps the curve of PPE upwards.

In Figs. 6.2c and 6.2d we show the dependence of PPE, when applied to mfBm, on the number of variables m and the independence in the case $H_i = H_j$ for all variables i, j. For this, the delay is fixed to $\tau = 1$. Again, the left figure (Fig. 6.2c) considers the special case where all components of the Hurst parameter H_i are equal. Again,

all green lines, i.e., for $d = 2$, are constant $-\ln(1/2)$ for each m, confirming the independence of numbers of variables m in Thm. 6.2.1. All other lines, i.e., for $d > 2$, are also the same for any m, but monotonically dependent on the Hurst parameter H, as given in Cor. 6.2.1. The right figure (Fig. 6.2d) shows the general case where the components of the Hurst parameter H_i are not equal. Specifically, $H_i = H_j$ for $i, j \neq 1$ is variable, while $H_1 = 0.6$ is fixed. We have rotated the view of the figure so that we can see that the lines for $d > 2$ fall more sharply as the number of variables m increases, confirming the dependence on the number of variables m in Thm. 6.2.2. All in all, the experiments underpin our theoretical results from the previous section.

Note that in Fig. 6.2, especially for increasing H, some deviations (jittering) in the data from a theoretically monotonic line are visible. In particular, this is related to the experimental length constraints of $T < \infty$, because the distributions given in Section 6.2 can only be expected to be exact if $T \to \infty$. Thus, the estimates of the values of PPE of mfBm for a small length T differ from the true values based on a hypothetical mfBm of infinite length. Visually, this phenomenon is underpinned by the following three aspects:

i) In general, the distribution of ordinal patterns in mfBm becomes more deterministic or predictable as H increases, i.e., lower entropy applies. In all sub-figures of Figure 6.2, the deviations in the distributions due to the length restriction $T < \infty$ have a stronger effect on the value of PPE in the case $H \to 1$ than in the case $H \to 0$, where an approximately equal distribution holds.

ii) In Figure 6.2a, the deviations become larger as the delay τ increases, because the number of time points to which an ordinal pattern is assigned decreases to $T - (d\tau - \tau)$. That is, the length of the time series on which PPE is technically based decreases further, increasing the number and strength of the deviations in the distributions.

iii) In Figure 6.2c, the deviations increase as the number of spatial variables m increases, because the number of time points to which an ordinal pattern is assigned increases to $m \cdot (T - (d\tau - \tau))$. This means that the length of the time series on which PPE is based becomes greater, which reduces the number and strength of the deviations in the distributions.

Dávalos *et al.* (2018) provide a detailed discussion of the implications of fBm's experimental length constraints on the distribution of ordinal patterns.

6.4 Interim Conclusion: PPE

In this chapter, we have investigated the behaviour of PPE of different orders d and delays τ on mfBm of different numbers of variables m in variation of its Hurst parameter $H \in (0, 1)^m$. For order $d = 2$, PPE is constant, namely $-\ln(1/2)$, independent of the number of variables, the Hurst parameter or the delay. However, for orders $d > 2$, the use of PPE provide interesting insights and possible applications. The

distribution of ordinal patterns of orders $d > 2$, and thus also PPE, are directly related to the Hurst parameter H and can be expressed in formulae. For example, considering the estimation of H as an inverse problem, PPE can be used to solve it, since PPE depends monotonically on H, i.e., the entropy decreases as H increases. In Chapter 11 we discuss PPE in comparison to further representations.

Chapter 7

Multivariate Multi-Scale Permutation Entropy Applied to Multivariate Fractional Brownian Motion

After investigating the behaviour of PPE on mfBm, this chapter investigates the behaviour of MMSPE on mfBm in a theoretical as well as experimental setting. We first presented this idea in the following conference paper.

> Marisa Mohr, Nils Finke, and Ralf Möller. On the Behaviour of Permutation Entropy on Fractional Brownian Motion in a Multivariate Setting. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2020 (APSIPA-ASC)*, pages 189–196, 2020

This chapter provides contribution 2b) of this work. The computations of MSPE and MMSPE are based on a coarse-grained process. Thus, we start with corresponding related work of the coarse-grained process on fBm by Delignières (2015); Dávalos *et al.* (2018) and adapt the results to the multivariate case of mfBm. This is followed by a theoretical analysis and an experimental evaluation of the behaviour of MMSPE on mfBm. We end with an interim conclusion.

7.1 Coarse-Grained Fractional Brownian Motion

As introduced in Section 4.1.1 and Algorithm 2, MMSPE includes the additional step of coarse-gaining or scaling before PPE is performed in its origin. To investigate the behaviour of MMSPE on mfBm, coarse-grained multivariate fractional Brownian motion (cmfBm) has to be derived and analysed first. To calculate the elements of the multivariate coarse-grained time series, we apply Eq. (4.2) to mfBm. That is, cmfBm is defined by

$$B_{H_i}^{i,(\epsilon)}(l) = \frac{1}{\epsilon} \sum_{j=1}^{\epsilon} B_{H_i}^{i}((l-1)\epsilon + j) \tag{7.1}$$

for $l = 1, ..., T/\epsilon$ and all $i = 1, ..., m$. By definition, the coarse-graining procedure is a linear combination of the elements in the scale ϵ, where each variable i marginally

is an fBm. As fBm refers to the incremental process of correlated Gaussian variables, we consider in the following

$$B_H(t) = \sum_{i=0}^{t} G_H(i), \tag{7.2}$$

$$G_H(t) = B_H(t) - B_H(t-1), \tag{7.3}$$

where the fractional Gaussian noise (fGn) $G_H(t)$ refers to the individual step. As Gaussian processes, fGn is completely characterised by the expected value and the autocovariance function (Delignières, 2015), given by

$$\mathrm{E}\left(G_H(t)\right) = 0, \tag{7.4}$$

$$\mathrm{Cov}\left(G_H(t), G_H(t+s)\right) = \frac{\sigma^2}{2}(|t+1|^{2H} + |t-1|^{2H} - 2|t|^{2H}), \tag{7.5}$$

$$\rho_G(k) = \frac{1}{2}(|t+1|^{2H} + |t-1|^{2H} - 2|t|^{2H}), \tag{7.6}$$

where σ^2 is the variance of any individual Gaussian step $G_H(t)$.

We follow the arguments of Dávalos *et al.* (2018), who characterise the coarse-grained fractional Gaussian noise (cfGn) defined as

$$G_H^{\epsilon}(l) = \frac{1}{\epsilon}\sum G_H((l-1)\epsilon + j) = \frac{1}{\epsilon}(B_H(\epsilon l) - B_H(\epsilon(l-1))) \tag{7.7}$$

by its expected value and its autocovariance function. As the expected value of a sum of independent random variables is the sum of the expected value of each variable, it holds $\mathrm{E}(G_H^{\varepsilon}(l)) = 0$. Using Definition 3.2.2 (iii), Dávalos *et al.* (2018) derive the variance and covariance of cfGn as

$$\mathrm{Var}(G_H^{\epsilon}(l)) = \frac{1}{\epsilon^2}\,\mathrm{E}\left((B_H(\epsilon l) - B_H(\epsilon(l-1)))^2\right) \tag{7.8}$$

$$= \sigma_i^2 \epsilon^{2(H-1)} \tag{7.9}$$

and

$$\mathrm{Cov}(G_H^{\epsilon}(l), G_H^{\epsilon}(m)) = \sigma^2 \epsilon^{2(H-1)} \rho_G(k). \tag{7.10}$$

The structure of the autocorrelation function is the same as the original fGn, but with additional information of the scale factor ϵ, i.e., it holds that

$$\rho_{G,\epsilon}(k) = \frac{\sigma^2 \epsilon^{2(H-1)} \rho_G(k)}{\sigma^2 \epsilon^{2(H-1)}} = \rho_G(k). \tag{7.11}$$

It follows that the autocovariance function of the original fGn is invariant to the coarse-graining procedure. Transferring the properties of fGn to fBm and mfBm, whose spatial variables are marginally fBms, also results in invariance for these with respect to the coarse-graining procedure. In the following we use the definition of cmfBm and its properties to investigate the behaviour of MMSPE on mfBm.

7.2 Theoretical Analysis

With previous insights, the transfer of results from PPE to MMSPE is straightforward. In analogy to PPE, MMSPE of order $d = 2$ is constant for all delays τ and Hurst parameters H of mfBm. In addition, it is independent of all numbers of variables m of mfBm.

Theorem 7.2.1. *For order $d = 2$, it is*

$$\mathrm{MMSPE}_{d,\tau,\epsilon}(\mathbf{B}_H^m(t)) = -\ln(1/2) \tag{7.12}$$

for all delays $\tau \in \mathbb{N}$, scales $\epsilon \in \mathbb{N}$, Hurst parameters $H \in (0,1)^m$, and number of variables $m \in \mathbb{N}$ of mfBm.

Proof. For mfBm, each variable $i = 1, \ldots, m$ at the margin is a fBm. Since MSPE and MMSPE are based on PE and PPE, respectively, and the underlying distributions in the coarse-grained procedure of cmfBm follow those of the original fBm, the distribution of ordinal patterns of orders $d = 2$ of cmfBm are the same as in Eq. (6.1). In particular, they are scale-invariant. Hence the claim follows with Thm. 6.2.1. □

For orders $d = 3$ and $d = 4$ there exist a dependence on the number of variables m and on the Hurst parameter H of mfBm.

Theorem 7.2.2. *For orders $d = 3, 4$, $\mathrm{MMSPE}_{d,\tau}(\mathbf{B}_H^m(t))$ is independent of all delays $\tau \in \mathbb{N}$ and scale factors $\epsilon \in \mathbb{N}$ but monotonically dependent on the number of variables $m \in \mathbb{N}$ and on the Hurst parameter $H \in (0,1)^m$ of mfBm.*

Corollary 7.2.1. *If $H_i = H_j$ for all i, j, then $\mathrm{MMSPE}_{d,\tau}(\mathbf{B}_H^m(t))$ is independent of the number of variables $m \in \mathbb{N}$ of mfBm.*

We prove Thm. 7.2.2 and Cor. 7.2.1 in one.

Proof. For mfBm, each variable $i = 1, \ldots, m$ is marginally an fBm. Since the covariance function of fBm is invariant to the coarse-grained procedure in Eq. (7.1), the independence of scales ϵ follows. Then Thm. 6.2.2 is directly applicable, which implies independence from the delay τ, the number of variables m (in the case of $H_i = H_j$) and monotonic dependence on the Hurst parameter H. □

For orders $d \geq 5$, no closed formulas for the distribution of ordinal patterns in mfBm exist (see Section 6.1). Nevertheless, analogous behaviour can be observed in this case as well, which we evaluate in an experimental setting in the next section.

7.3 Experimental Evaluation

In this experiment we compute MMSPE of different orders d, different delays τ and different scale factors ϵ for simulations of mfBm with different numbers of variables m and different Hurst parameters H_i for $i = 1, \ldots, m$. The results of the computations of MMSPE of orders $d = 2$ (green), $d = 3$ (orange), $d = 4$ (blue), and $d = 5$ (red) are visualised in Fig. 7.1 and underpin the theorems derived in Section 7.2. The experimental computations are based on same simulations of mfBms as described in Section 6.3. In particular, in the experiments we underpin the following three aspects

i) independence from the delay τ,

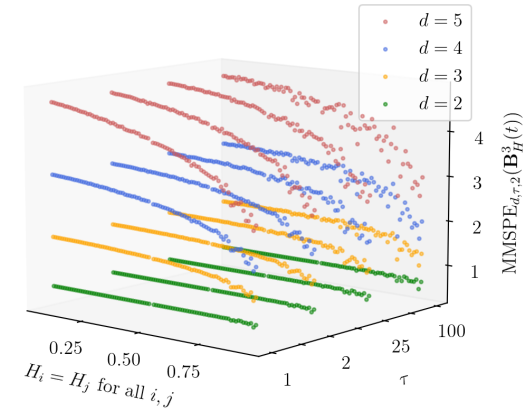
ii) independence from the scale factor ϵ,

iii) (in)dependence from the number of variables m.

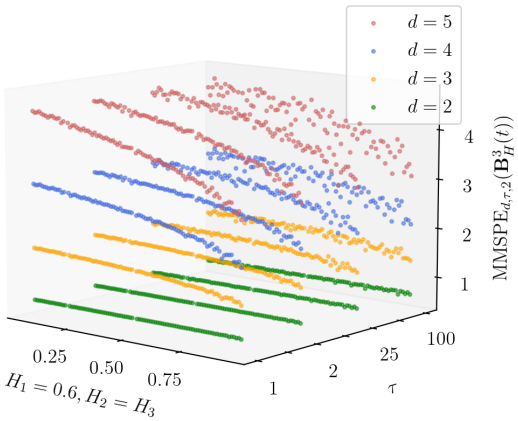
In general, all sub-figures of Fig. 7.1 show a constant MMSPE for order $d = 2$, namely $-\ln(1/2)$, while for orders $d > 2$ there is a monotonic dependence on the Hurst parameter H, i.e., for increasing Hurst parameter H the entropy decreases. The deviations with increasing Hurst parameter H occur for the same reason as in the last experiment in Section 6.3, namely because of experimental length constraints. In Figures 7.1a and 7.1b, we show the independence of MMSPE on mfBm from the delay τ in the same way as in Figs. 6.2a and 6.2b with the difference that the scale factor ϵ is set to 2. The results are the same as in Figs. 6.2a and 6.2b, i.e., in the case of PPE, which is a first indication that the scaling of mfBm has no influence on MMSPE. Rather, however, Figs. 7.1a and 7.1b show the independence of MMSPE of orders $d = 2, \ldots, 5$ from the delay τ as all line are the same for each scale factor ϵ.

Moreover, in Figs. 7.1c and 7.1d, we underpin the independence of MMSPE on mfBm from the scale factor ϵ. For this, the number of variables is fixed to $m = 3$ and the delay to $\tau = 1$. Again, all green lines, i.e., for $d = 2$, are constant with value $-\ln(1/2)$ for each $\epsilon \in \{1, 2, 5, 10\}$, confirming the independence of the scale factor ϵ in Thm. 7.2.1. All other lines, i.e., for $d > 2$, are also the same for any scale $\epsilon \in \{1, 2, 5, 10\}$, but monotonically dependent on the Hurst parameter H, in particular confirming Thm. 7.2.2. Finally, scaling has no influence on the behaviour of MMSPE on mfBm.

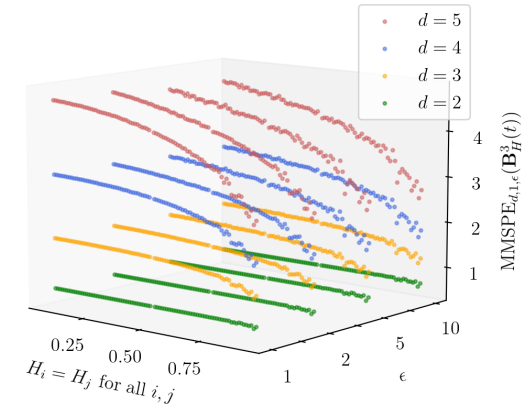
In the end, Figs. 7.1e and 7.1f show the independence of MMSPE, when applied to mfBm, from the number of variables m in the special case $H_i = H_j$ for all variables i, j (left) and the dependence in the general case (right). The delay is set to $\tau = 1$ and the scale factor to $\epsilon = 2$. Again, all green lines, i.e., for $d = 2$, are constant with value $-\ln(1/2)$ for each m, confirming the independence of numbers of variables m in Thm. 7.2.1. All other lines in Fig. 7.1f, i.e., for $d > 2$, are also the same for any m, but monotonically dependent on the Hurst parameter H, as stated in Cor. 7.2.1. On the other hand, in Fig. 7.1e, this time without rotating the view, but with a sharp eye, it can still be seen how the lines for $d > 2$ decrease more rapidly with increasing number of variables m, again confirming the dependence on the number of variables m in Thm. 7.2.2.



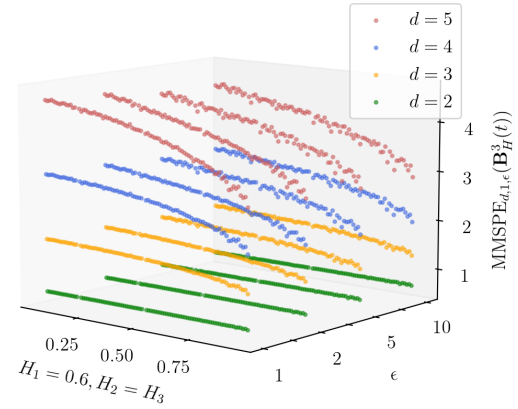
(a) $\tau \in \{1, 2, 25, 100\}$, $\epsilon = 2$, $m = 3$, $H_i = H_j$.



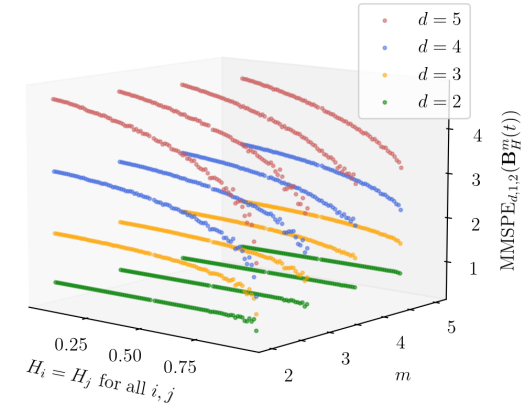
(b) $\tau \in \{1, 2, 25, 100\}$, $\epsilon = 2$, $m = 3$, $H_i \neq H_j$.



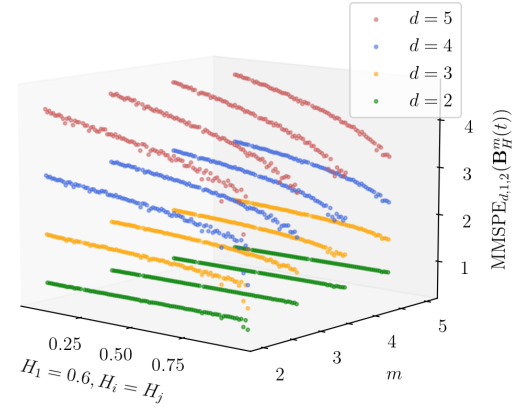
(c) $\tau = 1$, $\epsilon \in \{1, 2, 5, 10\}$, $m = 3$, $H_i \neq H_j$.



(d) $\tau = 1$, $\epsilon \in \{1, 2, 5, 10\}$, $m = 3$, $H_i = H_j$.



(e) $\tau = 1$, $\epsilon = 2$, $m \in \{2, 3, 4, 5\}$, $H_i \neq H_j$.



(f) $\tau = 1$, $\epsilon = 2$, $m \in \{2, 3, 4, 5\}$, $H_i = H_j$.

Figure 7.1: Experimental computations of MMSPE of orders $d = 2$ (green), $d = 3$ (orange), $d = 4$ (blue), and $d = 5$ (red) on mfBm in variation of its Hurst parameter H.

All in all, the experiments underpin our theoretical results from the previous Section 7.2. Since MMSPE is based on PPE and scaling has no influence on mfBm, the behaviour of MMSPE on mfBm in Fig. 7.1 is to be expected the same as that of PPE in Fig. 6.2. A direct comparison between PPE and MMSPE, as well as other representations and their possible applications, is discussed in Chapter 11.

7.4 Interim Conclusion: MMSPE

In this chapter, we have investigated the behaviour of MMSPE of different orders d and delays τ on mfBm of different numbers of variables m in variation of its Hurst parameter $H \in (0,1)^m$. For order $d = 2$, MMSPE is constant, namely $-\ln(1/2)$, independent of the number of variables, the Hurst parameter or the delay. In the case of order $d > 3$, the behaviour of MMSPE on mfBm is monotonically dependent on the Hurst parameter $H \in (0,1)^m$, i.e., the entropy decreases with increasing H – as does PPE.

In general, scaling the elements of mfBm by a coarse gained procedure does not reveal relevant structures on mfBm with infinite length, although the long and short memory correlations of mfBm generate a very complex behaviour. The reason for this is that the process is scale invariant. This is particularly related to or reflects the fractal property, because when a fractal is zoomed in or scaled, it resembles the original shape. Since scaling does not change the structure of mfBm, MMSPE of any scale ϵ is equal to PPE and analysis with MMSPE of higher orders $d > 2$ does not provide any additional insight compared to PPE. Further representations are necessary to inspect the qualitative behaviour of mfBm, as follows in the next chapters.

Chapter 8

Multivariate Weighted Permutation Entropy Applied to Multivariate Fractional Brownian Motion

After examining the existing approaches, PPE and MMSPE, this chapter explores the behaviour of the newly introduced multivariate weighted permutation entropy (MWPE) on mfBm in both a theoretical and experimental framework. We first presented this idea in the following conference paper.

> Marisa Mohr, Florian Wilhelm, and Ralf Möller. On the Behaviour of Weighted Permutation Entropy on Fractional Brownian Motion in the Univariate and Multivariate Setting. *The International FLAIRS Conference Proceedings*, 34, 2021

The univariate case, i.e., the behaviour of WPE on fBm, has not yet been studied in research, which is why we start with it. The remainder of this chapter contains contribution 2c) of this work. A theoretical analysis and an experimental evaluation of the behaviour of MWPE on mfBm follows. Again, we end with an interim conclusion.

8.1 The Univariate Case: WPE on fBm

We first consider the behaviour of WPE of order $d = 2$ on fBm.

Theorem 8.1.1. *For order $d = 2$ and $H \in (0, 1)$, it holds that*

$$\mathrm{WPE}_{2,\tau}(B_H(t)) = -\ln(1/2) \tag{8.1}$$

for all delays $\tau \in \mathbb{N}$

Proof. WPE differs from PE in that the ordinal patterns are weighted depending on their position t according to Eq. (3.20). For a weight w_t of order $d = 2$, i.e., of two time steps x_{t-1} and x_t, we have

$$w_t = \frac{1}{2}\sum_{k=1}^{2}(x_{t-(k-1)\tau} - \overline{x}_t^{2,\tau})^2 \tag{8.2}$$

$$= \frac{1}{2}(x_t - x_{t-\tau})^2. \tag{8.3}$$

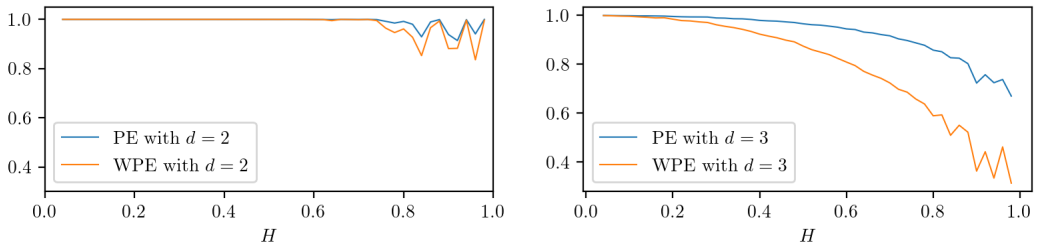


Figure 8.1: Comparison of PE and WPE.

Since $x_t \sim B_H(t)$, from Definition 3.2.4, property (3), i.e., fBm has stationary increments, we conclude that

$$\frac{1}{2}(B_H(t) - B_H(t-\tau))^2 \sim \frac{1}{2}(B_H(\tau))^2 \tag{8.4}$$

with $\mathrm{Var}(B_H(\tau)) = \sigma^2\tau^{2H}$ and $\sigma^2 = \mathrm{Var}(B_H(1)) = s^{2H}$ as in Definition 3.2.2 (iii). Consequently, the weights w_t are independently distributed from t, i.e.,

$$w_t \sim \mathcal{N}(0, \frac{1}{2}(s\tau)^{2H}). \tag{8.5}$$

Considering the distribution of all possible realisations of fBm, we see from the use of the weights w_t from Eq. (3.20) in the calculation of WPE that it cancels out for a constant delay $\tau \in \mathbb{N}$. With Eq. (6.1) follows the assertion, i.e.,

$$\mathrm{WPE}(B_H(t)) = -\ln(1/2). \tag{8.6}$$

. □

Figure 8.1 (left) confirms the match of PE and WPE of order $d = 2$ applied to fBm, i.e., weighting has no influence in this case. All experimental calculations are based on a simulation of fBm using the fBm package available on PyPI. Note that in the case of $d \geq 3$, however, the weighting has influence due to the lack of symmetry. The weight of order $d = 3$ is given by

$$w_t = \frac{2}{9}(x_t^2 - x_t(x_{t-\tau} - x_{t-2\tau}) + x_{t-\tau}^2 + x_{t-2\tau}^2 - x_{t-\tau}x_{t-2\tau})). \tag{8.7}$$

In particular, the above arguments cannot be applied. To confirm this, Fig. 8.1 (right) shows that WPE of order $d = 3$ applied to fBm decreases more sharply with increasing Hurst parameter H than PE. Thus, for order $d = 3$, the weighting must affect the distribution of ordinal patterns.

Table 8.1 shows exemplary the frequencies of (weighted) ordinal patterns (wp_j) p_j of order $d = 3$ and delay $\tau = 1$ applied to a simulated fBm of length $T = 30,000$ with Hurst parameter $H = 0.8$. While for PE it is $w_t = 1$ for every pattern j, we see that the weights w_t of ordinal patterns $(0, 1, 2)$ and $(2, 1, 0)$ are on average larger

j	counts	w_t	wp_j	p_j
$(0,1,2)$	10539	0.76	0.48	0.35
$(0,2,1)$	2470	0.19	0.03	0.08
$(1,0,2)$	2482	0.19	0.03	0.08
$(1,2,0)$	2457	0.19	0.03	0.08
$(2,0,1)$	2462	0.19	0.03	0.08
$(2,1,0)$	9588	0.72	0.40	0.33

Table 8.1: Frequencies of (weighted) ordinal patterns (wp_j) p_j of order $d = 3$.

than those of the other four ordinal patterns. Consequently, the frequencies of these ordinal patterns are also higher than in the unweighted case of PE. With Eq. (6.3), WPE decreases more sharply than PE. For $H > 0.5$, fBm is positively correlated, i.e., after an upward jump a further upward jump is more likely to follow and vice versa, which increases the effect. For ordinal patterns with order $d > 3$, a similar behaviour is to be expected. For example, the weight of order $d = 4$ is given by

$$\begin{aligned} w_t &= \frac{1}{16}(3x_t^2 - 2x_t(x_{t-\tau} + x_{t-2\tau} + x_{t-3\tau}) + 3x_{t-\tau}^2 \\ &\quad -2x_{t-\tau}(x_{t-2\tau} + x_{t-3\tau}) + 3x_{t-2\tau}^2 - 2x_{t-2\tau}x_{t-3\tau} + 3x_{t-3\tau}^2). \end{aligned} \tag{8.8}$$

In particular, WPE of orders $d > 2$ are dependent on the delay τ.

8.2 The Multivariate Case: Theoretical Analysis

In this section, we transfer the idea from the previous section to the multivariate case. We first consider the behaviour of MWPE of order $d = 2$ applied to mfBm. Again, MWPE of order $d = 2$ is constant for all delays τ, number of variables m and corresponding Hurst parameters H_i with $i = 1, \ldots, m$ of mfBm.

Theorem 8.2.1. *For order $d = 2$, it holds*

$$\mathrm{MWPE}_{2,\tau}(\mathbf{B}_H^m(t)) = -\ln(1/2) \tag{8.9}$$

for all delays $\tau \in \mathbb{N}$, Hurst parameters $H \in (0,1)^m$, and number of variables $m \in \mathbb{N}$ of mfBm.

Proof. First, the distribution of ordinal patterns of order $d = 2$ in fBm for calculating MWPE remains the same as in Equation (6.1), e.g., it is equally distributed for all Hurst parameters $H \in (0,1)$ and delays $\tau \in \mathbb{N}$. Second, the weighting of ordinal patterns of order $d = 2$ is the same for $(0,1)$ as for $(1,0)$, because the distribution and variance at time $t + 1$ is the same (for more details see Thm. 8.1.1).

Using these facts, the frequencies of the ordinal patterns in weighted the pooling matrix $P_w^{\tau\, d} \in \mathbb{R}^{m \times d!}$ are again equally distributed, i.e.,

$$P^{\tau,2}_{w_{i,(1,0)}} = P^{\tau,2}_{w_{i,(0,1)}} = \frac{1}{2m} \tag{8.10}$$

for all $i = 1, .., m$. When calculating the marginal weighted frequencies, the number of variables m is reduced again, that is

$$P^{\tau,2}_{w_{\cdot j}} = \frac{1}{2} \tag{8.11}$$

for all $j = 1, .., d!$. Overall, MWPE of order $d = 2$ applied to mfBm is neither dependent on delay τ, Hurst parameter H or on the number of variables m nor weighting has an effect on the calculation. □

Theorem 8.2.2. *For orders $d = 3, 4$, $\mathrm{MWPE}_{d,\tau}(\mathbf{B}_H^m(t))$ monotonically dependent on the delay $\tau \in \mathbb{N}$, number of variables $m \in \mathbb{N}$ and on the Hurst parameter $H \in (0,1)^m$ of mfBm.*

Corollary 8.2.1. *If $H_i = H_j$ for all i, j, then $\mathrm{MWPE}_{d,\tau}(\mathbf{B}_H^m(t))$ is independent of the delay $\tau \in \mathbb{N}$ and number of variables $m \in \mathbb{N}$ of mfBm.*

The proofs of Thm. 8.2.2 and Cor. 8.2.1 follow from Thm. 6.2.2 and Cor. 6.2.1 by adding factors from Section 8.1.

8.3 Experimental Evaluation

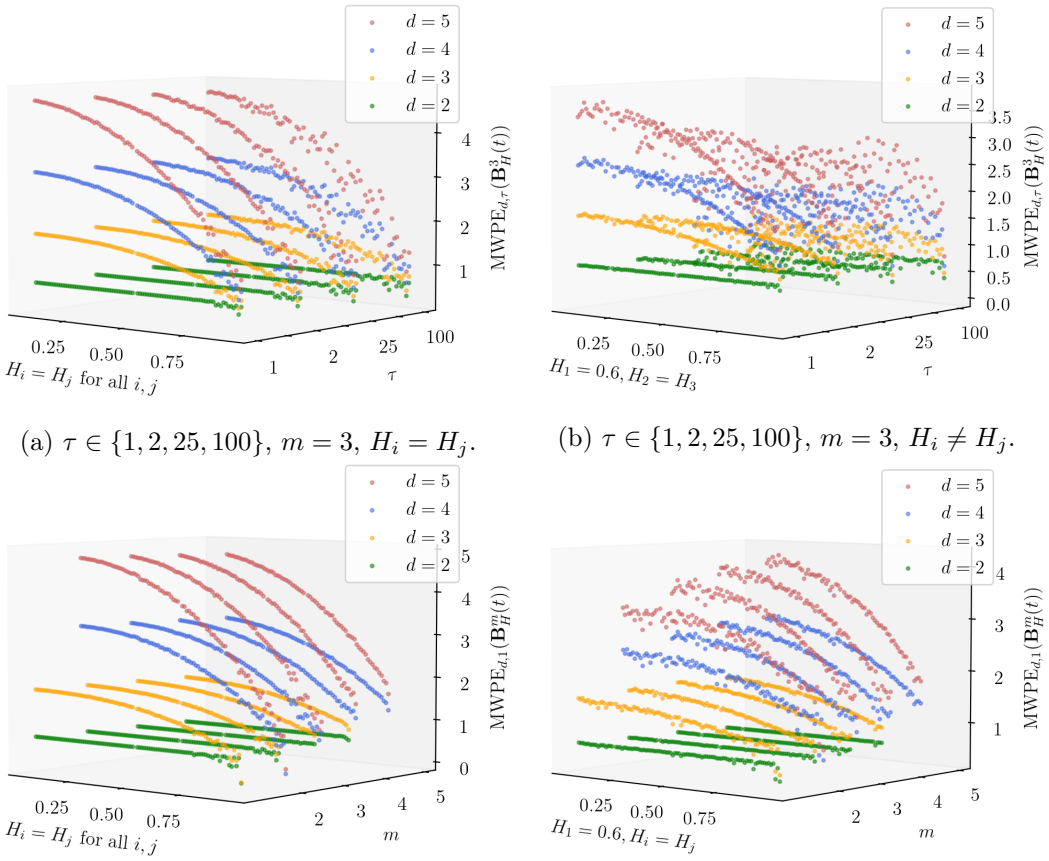
In this experiment we compute MWPE of different orders d and different delays τ for simulations of mfBm with different numbers of variables m and different Hurst parameters H_i for $i = 1, \ldots, m$. The results of the computations of MWPE of orders $d = 2$ (green), $d = 3$ (orange), $d = 4$ (blue), and $d = 5$ (red) are visualised in Fig. 8.2 and underpin the theorems derived in Sections 8.1 and 8.2. The experimental computations are based on same simulations of mfBms as described in Section 6.3. In particular, in the experiments we underpin the following two aspects

i) (in)dependence from the delay τ, and

ii) (in)dependence from the number of variables m.

Overall, all sub-figures of Fig. 8.2 show a constant MWPE for order $d = 2$ with value $-\ln(1/2)$ confirming Thm. 8.2.1. For orders $d > 2$ there is a monotonic dependence on the Hurst parameter H, i.e., for increasing Hurst parameter H the entropy decreases, which underpins a part of Thm. 8.2.2. The deviations with increasing Hurst parameter H occur for the same reason as in the last experiments, namely because of experimental length constraints.

Figures 8.2a and 8.2b illustrate the (in)dependence of MWPE on mfBm from the from the delay τ in the special case $H_i = H_j$ for all variables i, j (left) and the



(a) $\tau \in \{1, 2, 25, 100\}$, $m = 3$, $H_i = H_j$.

(b) $\tau \in \{1, 2, 25, 100\}$, $m = 3$, $H_i \neq H_j$.

(c) $\tau = 1$, $m \in \{2, 3, 4, 5\}$, $H_i = H_j$.

(d) $\tau = 1$, $m \in \{2, 3, 4, 5\}$, $H_i \neq H_j$.

Figure 8.2: Experimental computations of MWPE of orders $d = 2$ (green), $d = 3$ (orange), $d = 4$ (blue), and $d = 5$ (red) on mfBm in variation of its Hurst parameter H.

dependence in the general case (right). For this, the number of variables is fixed to $m = 3$. In Fig. 8.2b, all lines for $d > 2$ are the same for any delay τ confirming the independence in Cor. 8.2.1. In Fig. 8.2a, all lines for $d > 2$ decrease for increasing delay τ confirming the dependence in Thm. 8.2.2.

Figures 8.2c and 8.2d illustrate the (in)dependence of MWPE on mfBm from the number of variables m. For this, the delay is fixed to $\tau = 1$. All lines in Fig. 8.2c for $d > 2$, are the same for any m but monotonically dependent on the Hurst parameter H, as stated in Cor. 8.2.1. On the other hand, in Fig. 8.2d, the lines for $d > 2$ increase with increasing number of variables m, again confirming the dependence on the number of variables m in Thm. 8.2.2.

In summary, the experiments support our theoretical results from Sections 8.1 and 8.2.

8.4 Interim Conclusion: MWPE

In Section 5.1, we have proposed a well-defined definition of MWPE, that is consistent to other canonical multivariate extensions of PE. In this chapter, we have discussed the influence of weighting ordinal patterns of different orders $d = 2, \ldots, 5$ on fBm or mfBm. In case of order $d = 2$, we show that WPE and MWPE match with PE and PPE, respectively. There is no effect of weighting. For orders $d > 2$ we derive that certain ordinal patterns (strictly ascending and strictly descending) must have higher weights than others when calculating WPE and MWPE. Consequently, WPE and MWPE of order $d = 3$ decrease more strongly with increasing Hurst parameter H than PE and PPE. A direct comparison between PPE and MWPE, as well as other representations and their possible applications, is discussed in Chapter 11.

Unfortunately, the canonical definition of MWPE, as well as MWPE and PPE, does not take into account the interaction of different Hurst parameters H_i in the form of cross-correlations from Eq. (3.8), i.e., variables that influence each other at a fixed point in time and resulting possible simultaneous movements. Therefore, in the following we examine further multivariate extensions such as MPE-PCA or MOPPE, which are designed to consider cross-correlations.

Chapter 9

Multivariate Permutation Entropy Based on Principal Component Analysis Applied to Multivariate Fractional Brownian Motion

In the following, we investigate the behaviour of multivariate permutation entropy based on principle component analysis (MPE-PCA) on mfBm. We first presented this idea in the following conference paper.

> Marisa Mohr and Ralf Möller. Ordering Principal Components of Multivariate Fractional Brownian Motion for Solving Inverse Problems. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2021 (APSIPA-ASC)*, 2021

As shown in the previous chapters, the canonical multivariate extensions of PE, specifically PPE, MMSPE and MWPE of orders $d > 2$, are suitable for studying self-similarity, i.e., the Hurst parameter H. However, with this representations, as we discuss in this chapter, it is not possible to distinguish mfBms with the same Hurst parameters but different cross-correlations between variables. Figure 9.1 shows six realisations of mfBm of length $T = 2000$, different numbers of variables and different Hurst parameters with low correlation coefficients $\rho_{ij} = 0.1$ (left) such as high correlation coefficients $\rho_{ij} = 0.65$ (right). In the case of high cross-correlation between two variables i and j, the variables adjust their behaviour to each other. For example, the 5-th variable in Fig. 9.1(e) follows a steep downward trend caused by its high Hurst parameter $H = 0.75$. As the cross-correlation increases from 0.0 (left) to 0.65 (visualised in Fig. 3.2(f)), the steep downward trend adapts to the behaviour of the other variables with lower Hurst parameters $H_i \in [0.35, 0.5]$ for $i = 1, \ldots, 4$. A slight upward trend of the 5-th variable is noticeable. Moreover, the total range of values of all mfBms in all sub-figures of Fig. 3.2 decreases from the left to the right side, which underpins the adjustment of variables with higher cross-correlations.

Note that the cross-correlations of mfBm given in Equations (3.8) and (3.9) depend on both the correlation parameter $\rho_{ij} \in (-1, 1)$ and the time-reversibility $\eta_{ij} \in \mathbb{R}$ of mfBm. For simplicity, we focus on the correlation parameter ρ_{ij} and leave the time-

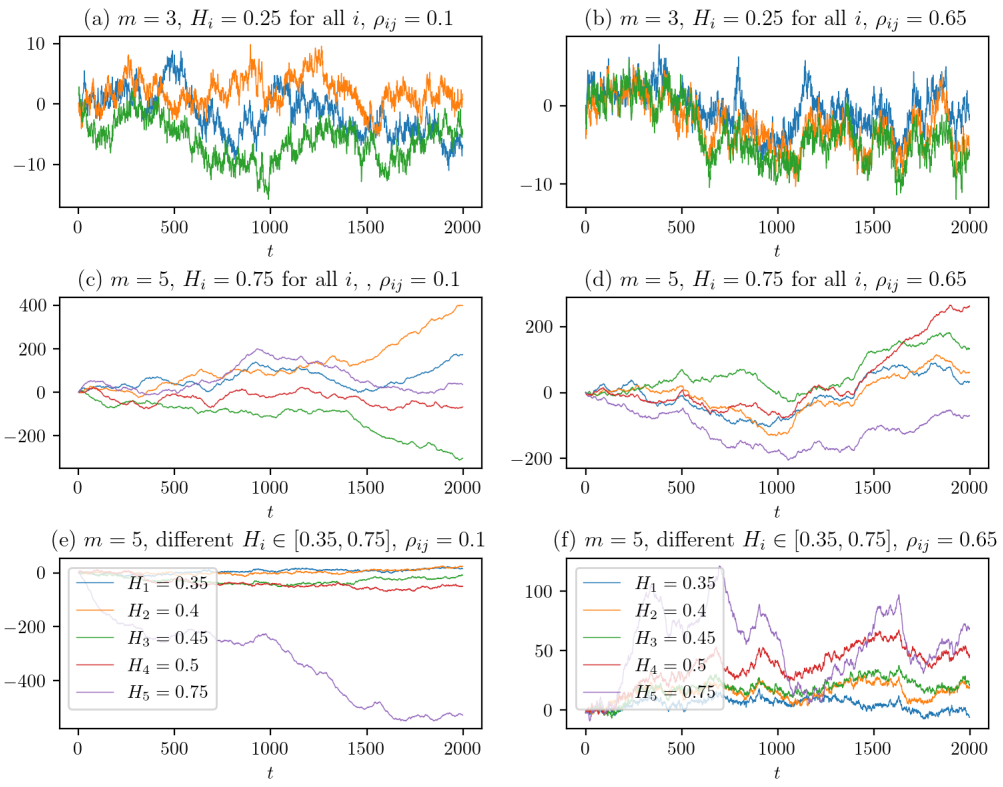


Figure 9.1: Six realisations of mfBm with different correlation coefficients $\rho_{ij} = 0.1$ (left) and $\rho_{ij} = 0.65$ (right).

reversibility parameter fixed at $\eta_{ij} = 0.1/(1 - H_i - H_j)$ as in all other experiments. If the correlation parameter increases, the cross-correlation also increases.

To distinguish between mfBm with the same Hurst parameters but different cross-correlations, MPE-PCA is an interesting approach as it is designed to handle correlations. We start with an application of PCA to mfBm. The remainder of this chapter contains contribution 2d) of this thesis, i.e., a theoretical analysis and an experimental evaluation of the behaviour of MPE-PCA when applied to mfBm. We end with an interim conclusion.

9.1 PCA Applied to mfBm

Given a sample of mfBm with number of variables m, i.e., $X = ((x_t^i)_{i=1}^m)_{t=1}^T \in \mathbb{R}^{m \times T}$, where each row $i = 1, \ldots, m$ marginally is an fBm by Lem. 3.2.1. As discussed in Section 5.2, PCA converts a set of observations of possibly correlated variables into a linear combination of uncorrelated variables. That is, a decomposition is found such

that the matrix-vector multiplication $\mathbf{X} \to \mathbf{Z} = \mathbf{V}^T\mathbf{X}$ gives the equations

$$\begin{aligned} Z^1 &= v_{11}X^1 + \cdots + v_{m1}X^m \\ Z^2 &= v_{12}X^1 + \cdots + v_{m2}X^m \\ \vdots &= \qquad\qquad \vdots \\ Z^m &= v_{1m}X^1 + \cdots + v_{mm}X^m \end{aligned}, \tag{9.1}$$

where X^i is a single fBm of length T. The variance of Z^1 is $\mathrm{Var}(Z^1) = \lambda_1$.

van Zanten (2007) shows that the local almost sure behaviour of a linear combination of independent fBms is equivalent to a multiple of a single fBm.

Corollary 9.1.1 (van Zanten (2007))**.** *Let $X = \sum_i^m a_k X^k$, where $X^1, \ldots, X^m$ are independent fBms with Hurst parameters $H_1 < \cdots < H_m$ and $a_1, \ldots, a_m \in \mathbb{R}\backslash\{0\}$.*

(i) If X is equivalent to a multiple of an fBm on $[0, T]$ for some $T > 0$, then X is equivalent to a_1X^1 and $H_2 - H_1 > 1/4$.

(ii) If $H_2 - H_1 > 1/4$ then X and a_1X^1 are locally equivalent.

Note that if $H_2 - H_1 < 1/4$, the local almost sure behaviour X of is not the same as that of any fBm. Thus, if all marginal fBms X^i of mfBm are independent, and for the Hurst parameters $H_2 - H_1 > 1/4$ holds, then Z^i and $v_{1i}X^1$ from Eq. (9.1) are locally equivalent. If there are cross-correlations between the variables of mfBm, PCA identifies eigenvectors that decorrelate the data and decorate the behaviour of fBm with a multiple. This results in different representations or principal components for mfBm with the same Hurst parameter H and different cross-correlations ρ_{ij}.

9.2 Ordinal Pattern Distributions of Principal Components

In general, (permutation) entropy increases with the degree of disorder and is maximum for absolutely random states. A time series with Hurst parameter $H = 1/2$ is called a random walk, while for Hurst parameter $H < 1/2$ or $H > 1/2$ the increments of (m)fBm are negatively or positively correlated, respectively. Thus, if the Hurst parameter varies, the entropy of (m)fBm also varies, i.e., if the Hurst parameter $H < 1$ increases, the information content of (m)fBm increases due to strong positive correlations. In case of high cross-correlation between the variables, the variables adjust their behaviour to each other as can be seen in Figure 3.2 (low cross-correlation at the top, high cross-correlation at the bottom). For example, as the cross-correlation increases, the steep downward trend of the 5-th variable in Figure 3.2(c), dominated by the high self-similarity or Hurst parameter $H = 0.75$, changes to a slightly upward trend in Figure 3.2(f), adjusted by the remaining variables with Hurst parameters $H_i \in [0.35, 0.5]$ for $i = 1, \ldots, 4$.

Hereafter we restrict ourselves to the case $H_2 - H_1 > 1/4$. Since in this paper we study the detection of cross-correlations using MPE-PCA, we consider two cases:

- $\rho_{ij} = 0$: Since performing PCA or decorrelation has no effect, the distributions of ordinal patterns and the behaviour of MPE-PCA and PPE-PCA on the principal components of mfBm are the same as in Section 6.1.

- $\rho_{ij} \neq 0$: Since the i-th principal component Z^i and $v_{1i}X^1$ are locally equivalent and each element $v_{ij} \in V$ represents a loading, namely the correlation between the original variable and the principal component, Z^i behaves like an fBm decorated by its loading. As the behaviour of fBm is directly related to the Hurst parameter H, but the distribution of ordinal patterns of order $d = 2$ does not depend on the Hurst parameter H of an fBm, neither do MPE-PCA$_{2,\tau,i}$ for all τ, i and PPE-PCA$_{2,\tau}$. It is MPE-PCA$_{2,\tau,i} = -\ln(1/2) =$ PPE-PCA$_{2,\tau}$ (see Section 6.1). In contrast, PE and PPE of orders $d = 3$ and $d = 4$ are monotonically dependent on the Hurst parameter H and Thm. 6.2.2), i.e., the decorrelation and the loadings, respectively, influence MPE-PCA and PPE-PCA. Although there are no closed formulas for $d > 4$, analogous behaviour is to be expected as in the chapters.

In contrast to MPE-PCA$_i$ and PPE-PCA, the computations of PE$_i$ on the i-th variable and PPE are independent of the Hurst parameter H as well as on cross-correlations, such that these measures are not able to detect cross-correlations.

9.3 Experimental Evaluation

In this experiment we compute MPE-PCA and PPE-PCA of different orders d and different delays τ for simulations of mfBm with different numbers of variables m and different Hurst parameters H_i for $i = 1, \ldots, m$. The experimental computations are based on same simulations of mfBms as described in Section 6.3 but with different correlation coefficients $\rho_{ij} = \{0.0, 0.8\}$ for all i, j. In order for Corollary 9.1.1 to be satisfied, the relationship $H_2 = H_1 - 0.26$ or $H_{2,3} = H_1 - 0.26$ is chosen for $m = 2$ or $m = 3$ variables, respectively. PE$_i$ and MPE-PCA$_i$ are calculated on the single-dimensional i-th variable and i-th principal component of mfBm with $i = 1, \ldots, m$, respectively. The visualised values correspond to the mean of 20 simulated experiments. The deviations with increasing Hurst parameter H occur for the same reason as in the last experiment in Section 6.3, namely because of experimental length constraints. In particular, with the experiments we support the following two aspects:

i) under certain conditions PCs of fBm behave like their origin , i.e., as H increases, MPE-PCA decreases, but

ii) unlike PPE, MPE-PCA detects when large cross-correlations at high Hurst parameter value H cause the behaviour of all variables to converge.

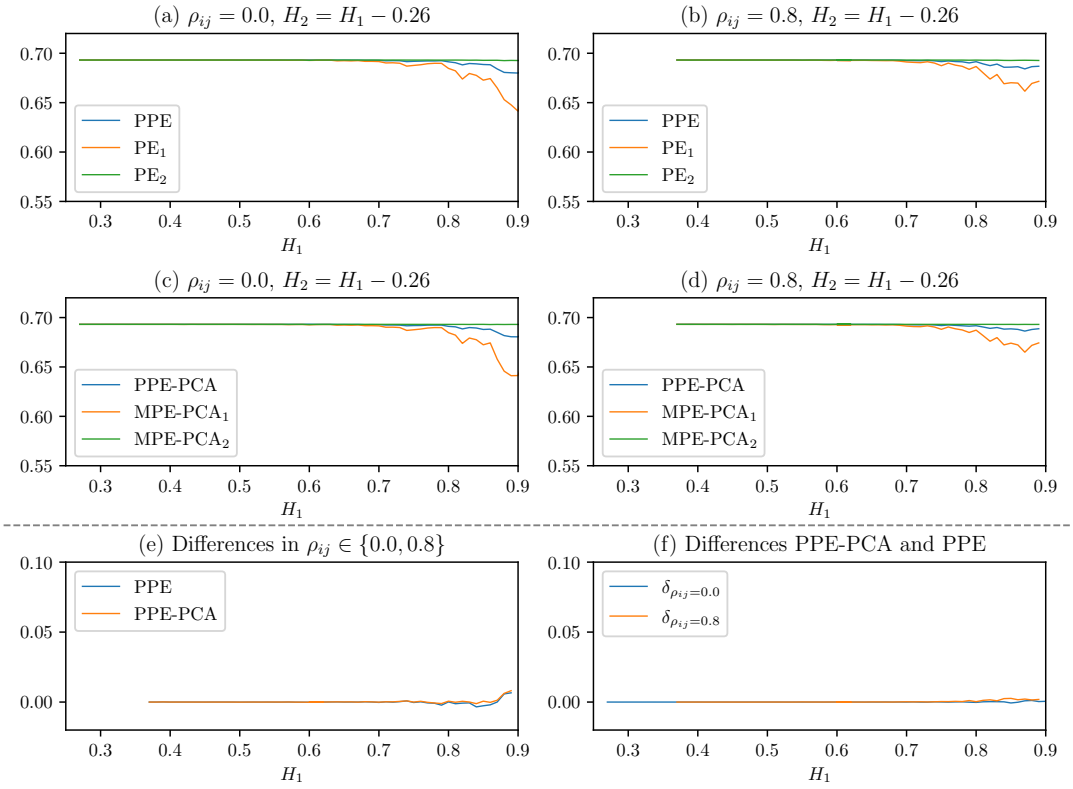


Figure 9.2: Comparison of PPE and MPE-PCA of order $d = 2$ on mfBm.

In Figure 9.2 we compare PPE (top) and MPE-PCA (middle) of order $d = 2$ and delay $\tau = 1$ on mfBms with $m = 2$ variables and different cross-correlations $\rho_{ij} = 0.0$ (left) and $\rho_{ij} = 0.8$ (right). Figure 9.2(a) shows a constant entropy of $-\ln(1/2)$ for all variables i, the pooled version PPE, and Hurst parameters H, again confirming Theorem 6.2.1. As PE and PPE of order $d = 2$ are independent of the Hurst parameter H as well as the cross-correlation, Figure 9.2(b), (c) and (d) are equal to Figure 9.2(a). Figure 9.2(e) and (f) confirm that neither PPE nor MPE-PCA of order $d = 2$ are able to detect cross-correlations of the variables.

In Fig. 9.3 we compare of PPE (top) and MPE-PCA (middle) of order $d = 3$ and delay $\tau = 1$ on mfBms with $m = 2$ variables and different cross-correlations $\rho_{ij} = \{0.0, 0.8\}$. Figure 9.3(a)–(d) confirm that PE, PPE, MPE-PCA and PPE-PCA depend monotonically on the Hurst parameter H, i.e., entropy decreases for increasing H. With Cor. 9.1.1, the principal components are locally equivalent to an fBm decorated by its loadings. Figure 9.2(e) and (f) confirm that unlike PPE and PPE-PCA of order $d = 2$, PPE-PCA of orders $d = 3$ and $d = 4$ are able to detect cross-correlation of variables, since decorrelation of variables using PCA decorates the behaviour of mfBm. In App. B.1 the same results can be found for order $d = 4$.

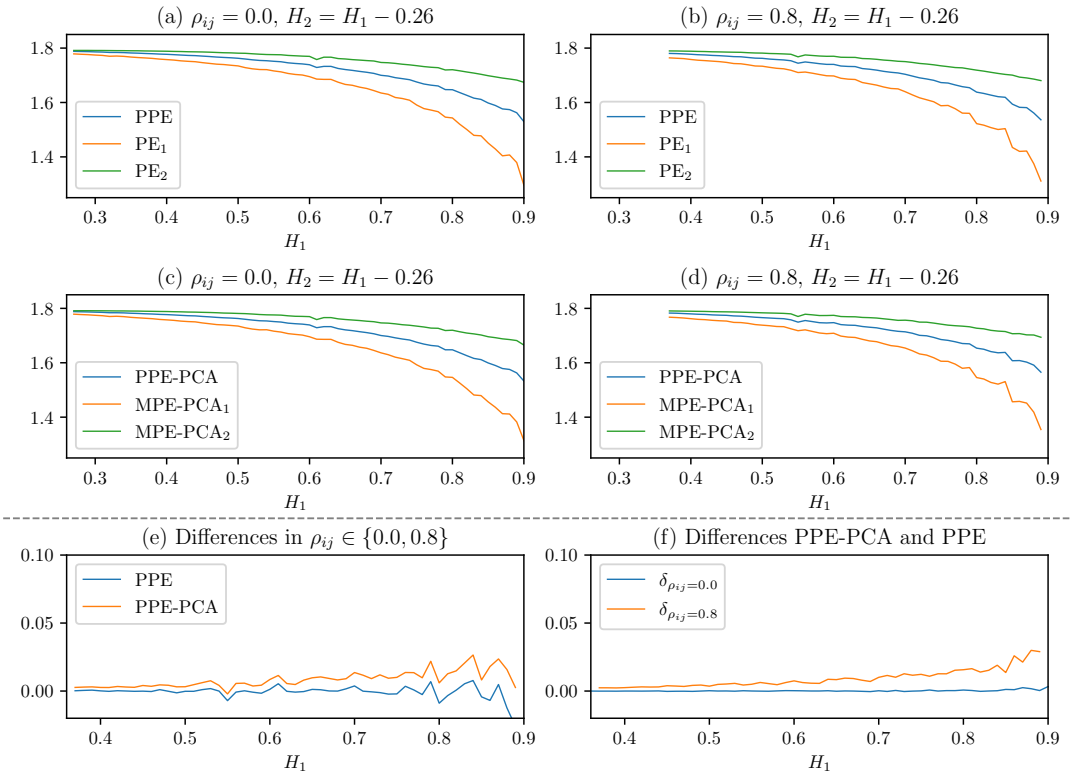


Figure 9.3: Comparison of PPE and MPE-PCA of order $d = 3$ applied to mfBm.

9.4 Interim Conclusion: MPE-PCA

In this chapter, we have investigated the behaviour of MPE-PCA and PPE-PCA of different orders d applied to mfBm in variation of its Hurst parameter $H \in (0, 1)^m$ with different correlations parameters $\rho_{ij} = \{0.0, 0.8\}$. For order $d = 2$, MPE-PCA and PPE-PCA are constant with $-\ln(1/2)$. We show that the entropies of the principal components of orders $d > 2$ are monotonically dependent on the Hurst parameter H, i.e., entropy decreases as H increases. Thus, MPE-PCA and PPE-PCA are appropriate for solving inverse problems, i.e., given an observed realisation of mfBm the calculation of MPE-PCA or PPE-PCA provide information about the level of H and thus parameters of the generating mfBm. Moreover, we show that unlike PPE, MPE-PCA or PPE-PCA of order $d > 2$ can uncover large cross-correlations at large Hurst parameter H. Since information about the Hurst parameter H as well as about correlations of the variables can be derived single-handed from MPE-PCA or PPE-PCA, this approach offers interesting advantages. Indeed, this chapter does not fully solve the inverse problem, but focuses on the theoretical relationships that motivate the solution of inverse problems. In Chapter 11 we discuss MPE-PCA or PPE-PCA in in the context of all other MPEs.

The main limitation of PCA is that it projects the data in a linear way, with many real-world challenges containing complex, non-linear relationships between variables. Since this work examines counts or sums of ordinal patterns, a linear relationship may be reasonable. Nevertheless, it remains to be investigated whether, for example, kernel PCA, independent component analysis or functional PCA analysis can improve the results.

Chapter 10

Multivariate Ordinal Pattern Permutation Entropy Applied to Multivariate Fractional Brownian Motion

Finally, we investigate the behaviour of multivariate ordinal pattern permutation entropy (MOPPE) applied to mfBm. We first presented this idea in the following journal article.

> Marisa Mohr and Ralf Möller. A Summary of Canonical Multivariate Permutation Entropies on Multivariate Fractional Brownian Motion. *Advances in Science, Technology and Engineering Systems Journal*, 6(5):107–124, 2021

Like MPE-PCA, MOPPE is designed to include interdependencies between the spatial variables of a multivariate time series in its real-valued representation. Unlike MPE-PCA, however, MOPPE does not involve correlations in the mathematical sense, but encodes the simultaneous movements of two or more spatial variables at a fixed point in time in its representation. This chapter provides contribution 2e) of this work. For this, we provide a theoretical analysis of the behaviour of MOPPE on mfBm in variation of its Hurst parameter H, before underpinning the results in an experimental setting. We end with an interim conclusion.

10.1 Theoretical Analysis

Since multivariate ordinal patterns are defined as univariate ordinal patterns combined in a matrix, joint probabilities are considered to investigate the behaviour of MOPPE on mfBm. Let p_j^τ be a probability for a univariate ordinal pattern j of arbitrary order d in fBm given in Section 6.1. Since any univariate ordinal pattern of any order d is independent of the delay τ, we write p_j in the following. Since there are no results on conditional probabilities of ordinal patterns in the literature, we focus on independent variables in the theoretical part.

In contrast to the previous MPE representations of order $d = 2$, the behaviour of MOPPE on mfBm is dependent on the number of variables m but remains independent of the delays τ.

Lemma 10.1.1. *Let $B^k_{H_k}(s)$ and $B^l_{H_l}(t)$ for every $k, l = 1, \ldots, m$ conditionally independent, then it holds*

$$\mathrm{MOPPE}_{2,\tau}(\mathbf{B}^m_H(t)) = -\ln\left(\frac{1}{2^m}\right) \tag{10.1}$$

for all τ.

Proof. The independence from the delay τ follows directly from Definition 5.3.1 and the distribution of the univariate ordinal patterns of order $d = 2$ in Eq. (6.1). Let $j \in \{(0,1),(1,0)\}$ be a univariate ordinal patterns of order $d = 2$. Let $X_k = B^k_{H_k}(t)$ and $X_l = B^l_{H_l}(t)$ for every $k, l = 1, \ldots, m$ conditionally independent, then the joint probability function satisfies

$$\begin{aligned} P(X_1 = j, \ldots, X_m = j) &= P(X_1 = j) \cdot \cdots \cdot P(X_m = j) && (10.2)\\ &= {p_j}^m && (10.3) \end{aligned}$$

With Eq. (6.1) it is $p_j = 1/2$ for every j, so that the joint distribution of every m-fold combination of ordinal patterns $j \in \{(0,1),(1,0)\}$ stays ${p_j}^m$. For number of variables m, there exist 2^m multivariate ordinal patterns as combinations from univariate ordinal patterns, so that

$$-\sum_{j=1}^{2^m} \frac{1}{2}^m \cdot \ln\left(\frac{1}{2}^m\right). \tag{10.4}$$

□

For orders $d = 3$ and $d = 4$, the following theorem can be derived by considering joint probabilities of univariate ordinal pattern distributions introduced in Section 6.1.

Lemma 10.1.2. *Let $X_k = B^k_{H_k}(t)$ and $X_l = B^l_{H_l}(t)$ for every $k, l = 1, \ldots, m$ conditionally independent, then for orders $d = 3$ and $d = 4$, $\mathrm{MOPPE}_{d,m}(\mathbf{B}^m_H(t))$ is independent of all delays τ, but dependent on number of variables m, and monotonically dependent on the Hurst parameter H.*

Proof. The independence from the delay τ follows directly from Definition 5.3.1 and the distribution of the univariate ordinal patterns or orders $d = 3$ or $d = 4$ in Eq. (6.2) or Eq. (6.4), respectively. As introduced in Section 6.1, the distribution of ordinal patterns of order $d = 3$ is twofold (see Eq. (6.3)). Therefore, let $A = \{(0,1,2),(2,1,0)\}$ and $B = \{(0,2,1),(1,2,0),(1,0,2),(2,0,1)\}$. Since multivariate ordinal patterns are combinations of univariate ordinal patterns, in the following we consider three cases where the ordinal patterns of all m variables are either from A, from B or from A and B. Again, let $B^k_{H_k}(s)$ and $B^l_{H_l}(t)$ be conditionally independent for each $k, l = 1, \ldots, m$.

1. Let the univariate pattern a_i of the i-th variable be $a_i \in A$ for all $i = 1, \dots, m$. Using Eq. (6.2) and Section 10.1 the joint distributions of all 2^m combinations of $a_1, \dots, a_m \in A$ are given by

$$\prod_{i=1}^{m} \frac{1}{\pi^m} \arcsin 2^{H_i - 1} =: \prod_{i=1}^{m} \frac{1}{\pi^m} u_i. \tag{10.5}$$

2. Let the univariate pattern b_i of the i-th variable be $b_i \in B$ for all $i = 1, \dots, m$. Using Eq. (6.3) and Section 10.1 the joint distributions of all 4^m combinations of $b_1, \dots, b_m \in B$ are given by

$$\prod_{i=1}^{m} \frac{1}{4}(1 - \frac{1}{\pi^m} u_i). \tag{10.6}$$

3. Let the univariate pattern c_i of the i-th variable be, where $c_1, \dots, c_k \in A$ and $c_{k+1}, \dots, c_m \in B$. Then using Eqs. (6.2) and (6.3) and Section 10.1 the joint distributions of all $6^m - 4^m - 2^m$ remaining combinations $c_1, \dots, c_m$ are given by

$$\prod_{i=1}^{k} \frac{1}{\pi^k} u_i \cdot \prod_{i=k+1}^{m} \frac{1}{4}(1 - \frac{1}{\pi^{m-k}} u_i). \tag{10.7}$$

In particular, the joint distributions or the distributions of the multivariate ordinal patterns remain monotonically dependent on the Hurst parameter H after the monotonic transformations. For order $d = 4$, formulas for joint distributions corresponding to multivariate ordinal patterns are derivable with Eq. (6.4) and Section 10.1 in the same way as for order $d = 3$. □

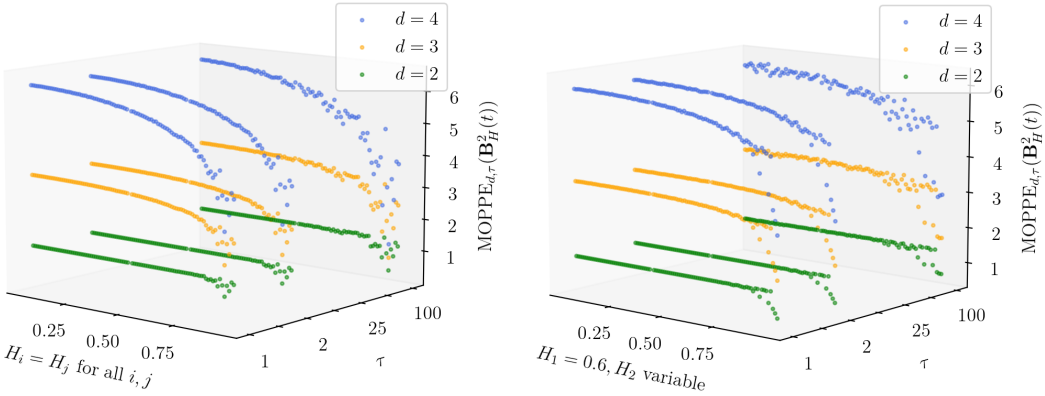
For orders $d > 4$, no closed formulas for the distribution of ordinal patterns in fBm exist.

10.2 Experimental Evaluation

In this experiment, we compute MOPPE of different orders d and different delays τ for simulations of mfBm in variation of the Hurst parameter H. Since the number of MOPs depends on the number of variables m and thus also on MOPPE, we refrain from an experimental evaluation of different m and assume $m = 2$ small for computational reasons. The experimental computations are based on the same simulations of mfBms as in Section 6.3 with the difference that in this experiment we include more than one correlation coefficient as MOPPE takes into account the interdependence of spatial variables. With this experiment we underpin two aspects:

(i) independence of MOPPE from the delay τ, and

(ii) dependence of MOPPE from the correlation coefficient ρ_{ij}.



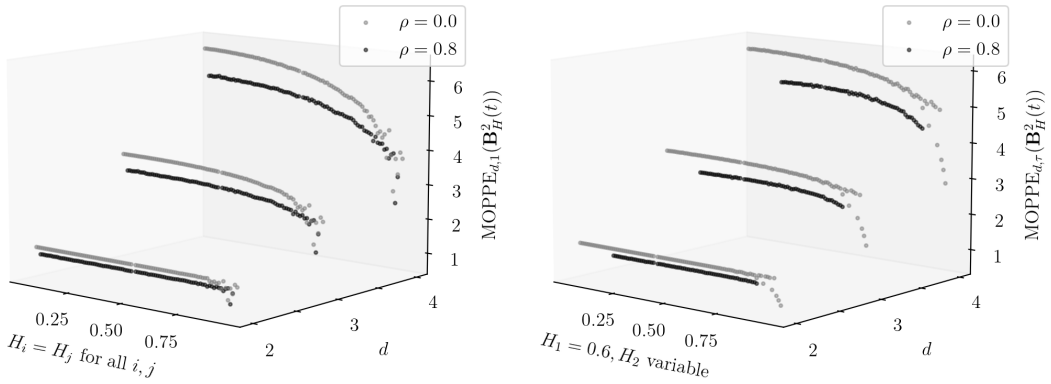
(a) $\tau \in \{1, 2, 5, 100\}$, $H_1 = H_2$, $\rho_{ij} = 0.0$. (b) $\tau \in \{1, 2, 5, 100\}$, $H_1 \neq H_2$, $\rho_{ij} = 0.0$.

Figure 10.1: Experimental computations of MOPPE for different delays τ applied mfBm with $m = 2$ spatial variables in variation of its Hurst parameters H.

The results of the computations of MOPPE of orders $d = 2$ (green), $d = 3$ (orange), and $d = 4$ (blue) for different delays τ applied to mfBm with correlation coefficient $\rho_{ij} = 0.0$ are visualised in Fig. 10.1 and underpin the lemmas derived in Section 10.1. The results of the computations of MOPPE for different correlation coefficients $\rho_{ij} = 0.0$ (gray) and $\rho_{ij} = 0.8$ (black) of mfBm are visualised in Fig. 10.2.

In Fig. 10.1 we show the independence of MOPPE on mfBm from delay τ. Again, the left figure (Fig. 10.1a) considers the special case where all components of the Hurst parameter H_i are equal. The right figure (Fig. 10.1b) shows the general case where the components of the Hurst parameter H_i are not equal. Specifically, H_2 is variable, while $H_1 = 0.6$ is fixed. For both, the number of variables is fixed to $m = 2$. All green lines, i.e., for $d = 2$, are constant with value $-\ln(1/4)$ for each τ, confirming the independence of delay τ in Equation (10.1). All other lines, i.e., for $d > 2$, are also the same for any τ, but monotonically dependent on the Hurst parameter H, i.e., for increasing Hurst parameter H the value of MOPPE decreases. The reason for the deviations of the values of MOPPE in Fig. 10.1 with increasing Hurst parameter H is the same as in the previous Section 6.3, namely from length restriction $T < \infty$ of mfBm. All in all, the experiment underpins our theoretical results from Lems. 10.1.1 and 10.1.2.

Since the definition of multivariate ordinal patterns involves dependence on several spatial variables at a fixed time, we examine dependence on cross-correlations of spatial variables in Fig. 10.2. A high cross-covariance or correlation of the variables causes the behaviour of the individual variables (for different Hurst parameters) to converge. The behaviour of one variable is influenced by another and so is the occurrence of the higher order ordinal patterns that are dependent on the Hurst parameter. For the investigation of correlations using MOPPE the delay or the number of variables are fixed to $\tau = 1$ or $m = 2$, respectively. In Fig. 10.2, the black lines corresponding to a correlation coefficient of 0.8 are constantly below the gray lines corresponding to a



(a) $\tau = 1$, $H_1 = H_2$, $\rho_{ij} \in \{0.0, 0.8\}$. (b) $\tau = 1$, $H_1 \neq H_2$, $\rho_{ij} \in \{0.0, 0.8\}$.

Figure 10.2: Experimental computations of MOPPE applied mfBm with $m = 2$ spatial variables and different correlation coefficients ρ_{ij} in variation of different Hurst parameters H.

correlation coefficient of 0.0. This reflects the fact that due to a higher correlation of the variables, there is a greater fit of the individual univariate ordinal pattern order in the multivariate combination, resulting in a smaller MOPPE value.

10.3 Interim Conclusion: MOPPE

In this chapter, we have investigated the behaviour of MOPPE of different orders d and delays τ on mfBm for different numbers of variables m in variation of its Hurst parameter $H \in (0, 1)^m$. For order $d = 2$, MOPPE is constant with value $-\ln(1/2^m)$, independent of the Hurst parameter H or the delay τ but dependent on the number of variables m. The distribution of ordinal patterns of orders $d = 2, 3, 4$, and thus also MOPPE, are directly related to the Hurst parameter H and can be expressed in formulae. Since MOP encodes the behaviour of several spatial variables at the same time, MOPPE, unlike canonical MPEs, is able to detect cross-correlations between spatial variables. In the following chapter, we compare MOPPE with all the representations discussed so far by summarising the individual advantages and disadvantages.

Chapter 11

Part II: Interim Conclusion

Up to this point, we have studied the behaviour of numerous multivariate representations of the class of multivariate permutation entropy (MPE), namely pooled permutation entropy (PPE), multivariate multi-scale permutation entropy (MMSPE), multivariate weighted permutation entropy (MWPE), multivariate permutation entropy based on principle component analysis (MPE-PCA) and multivariate ordinal pattern permutation entropy (MOPPE), applied individually to mfBm. In this chapter we summarise all the results and discuss both the differences and the similarities between the different representations. We first presented this comparison in the following journal paper.

> Marisa Mohr and Ralf Möller. A Summary of Canonical Multivariate Permutation Entropies on Multivariate Fractional Brownian Motion. *Advances in Science, Technology and Engineering Systems Journal*, 6(5):107–124, 2021

The differences between the representations result in different possible applications as well as different recommendations, with which this part of the work is concluded.

11.1 Comparison of MPE-Variants

The MPE-variants discussed in Part II have similarities as well as differences, especially in their behaviour when applied to mfBm. PPE can be understood as a canonical extension of PE that pools the univariate ordinal patterns of the individual spatial variables into a multivariate variant. MMSPE and MWPE belong to the same family as PPE as they are all based on a pooled matrix, while addressing different aspects within the time series and ordinal patterns respectively, namely scaling and amplitudes. One advantage of these approaches is that, in contrast to MPE-PCA and MOPPE, they also allow univariate analyses on the individual spatial variables within the algorithms (by a simple extension). MPE-PCA or MOPPE are not designed for univariate analysis, but on the other hand they are able to take into account interdependencies in the form of mathematical correlations or similar movements, respectively, of the spatial variables in a real-valued representation. Table 11.1 gives an overview of the dependencies of the different MPE-variants on the parameters belonging to MPE or mfBm.

	MPE		mfBm		
$d = 2$	$\tau \in \mathbb{N}$	$\epsilon \in \mathbb{N}$	$m \in \mathbb{N}$	$H \in (0,1)^m$	$\rho_{ij} \in (-1,1)$
PPE (Thm. 6.2.1)	✗	n\a	✗	✗	✗
MMSPE (Thm. 7.2.1)	✗	✗	✗	✗	✗
MWPE (Thm. 8.2.1)	✗	n\a	✗	✗	✗
MPE-PCA (Section 9.2)	✗	n\a	✗	✗	✗
MOPPE (Lem. 10.1.1)	✗	n\a	✓	✗	✗
$d = 3, 4, 5$	$\tau \in \mathbb{N}$	$\epsilon \in \mathbb{N}$	$m \in \mathbb{N}$	$H \in (0,1)^m$	$\rho_{ij} \in (-1,1)$
PPE (Thm. 6.2.2)	✗	n\a	✓	✓	✗
MMSPE (Thm. 7.2.2)	✗	✗	✓	✓	✗
MWPE (Thm. 8.2.2)	✓	n\a	✓	✓	✗
MPE-PCA (Section 9.2)	✗	n\a	✓	✓	✓
MOPPE (Lem. 10.1.2)	✗	n\a	✓	✓	✓

Table 11.1: Dependencies of the different MPE-variants on all parameters.

For order $d = 2$, all five MPE-variants are constant with value $-\ln(1/2)$ regardless of the delay $\tau \in \mathbb{N}$, scale factor $\epsilon \in \mathbb{N}$, number of variables $m \in \mathbb{N}$, Hurst parameter $H \in (0,1)^m$, or correlation coefficients $\rho_{ij} \in (-1,1)$ (or $-\ln(1/2^m)$ in case of MOPPE). Because all MPE representations behave equally when applied to mfBm, no characteristics or structures of mfBm can be derived via these representations. Therefore the usage of order $d = 2$ is not reasonable in applications. Moreover, since scaling does not change the structure of mfBm, MMSPE of any scale factor ϵ behaves equally to PPE and analysis with MMSPE of higher orders $d > 2$ does not provide any additional insight than PPE. PPE and MMSPE can be used interchangeably. This result is consistent with the fractal property or scale invariance of mfBm.

However, for orders $d > 2$, the use of PPE, MWPE, MPE-PCA and MOPPE provide interesting insights and possible applications to be discussed in the next section. The distribution of ordinal patterns, and thus the MPE-variants, is directly related to the number of spatial variables m and its individual Hurst parameter $H \in (0,1)^m$, which can be expressed in formulas. As given in Section 8.1, the weighting in MWPE is dependent on the delay τ. This pays off in that strictly rising as well as falling ordinal patterns are weighted more so that MWPE falls more sharply than PPE for increasing Hurst parameter H. Since MPE-PCA and MOPPE are designed to account for interdependencies of spatial variables, they are dependent on the correlation coefficients of mfBm.

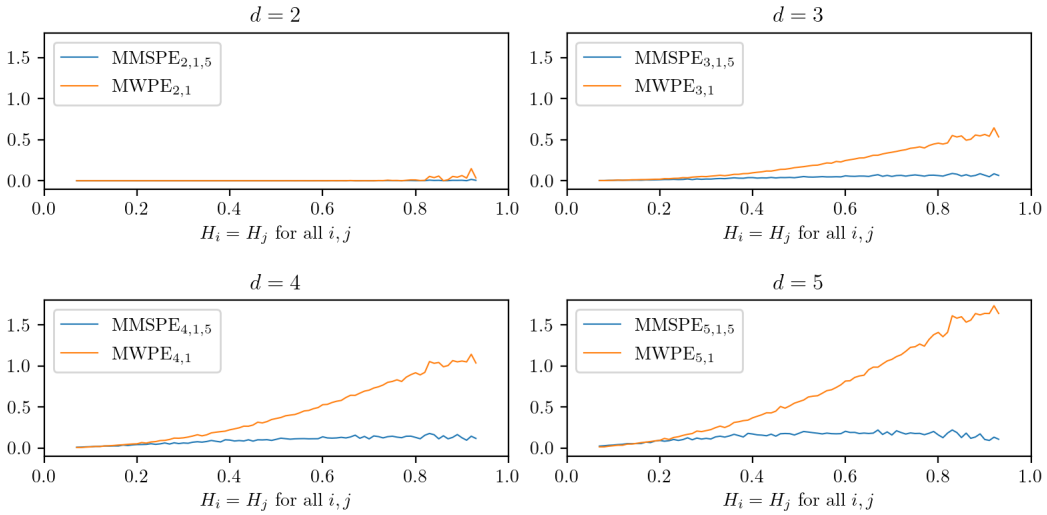
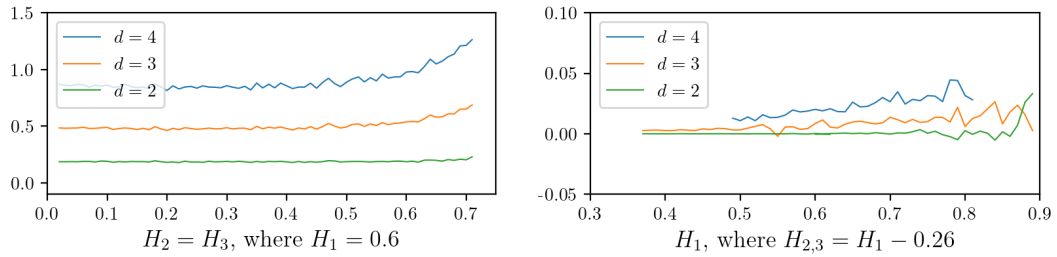


Figure 11.1: Differences between PPE and MMSPE or MWPE of different orders applied to simulations of mfBm in a variation of its Hurst parameter.

Figure 11.1 compares PPE with its adaptations MMSPE and MWPE with different orders $d = 2, \ldots, 5$ from top left to bottom right. We set $\tau = 1$, $\epsilon = 5$ and $m = 3$. The sub-figure top left shows the equality of PPE, MMSPE and MWPE for order $d = 2$, which again underpin Thms. 6.2.1, 7.2.1 and 8.2.1. Thus, scaling or weighting has no effect in this case. The other three right sub-figures show the equality of PPE and MMSPE (blue lines) for orders $d = 3, 4, 5$. Similarly, they show a difference between PPE and MWPE as the orange lines increase in variation of increasing H, indicating a faster decrease of MWPE.

MOPPE can be understood as a canonical extension of univariate ordinal patterns to multivariate ordinal patterns by conceiving univariate patterns as multidimensional patterns in a matrix. The advantage of this approach lies in consideration of the interdependencies of several variables at one point in time. A disadvantage in the application is the exponentially increasing number of possible ordinal patterns $(d!)^m$. When many patterns exist relative to the length of the time series, each pattern occurs in vanishingly small numbers, resulting in a uniform distribution and thus maximum entropy for $T \ll \infty$. In this case, MPE-PCA offers a reasonable alternative since here the multivariate problem is transformed into a univariate problem by means of PCA before univariate patterns of each order are determined on it.

Figure 11.2 visualises the differences of MOPPE or MPE-PCA values when applied to mfBm with fixed variable dimension $m = 3$ but different correlation coefficients $\rho_{ij} = \{0.0, 0.8\}$ for different orders $d = 2$ (green), $d = 3$ (orange) and $d = 4$ (blue), fixed delay $\tau = 1$ and in variation of its Hurst parameter. In both sub-figures, the differences are almost constantly zero for order $d = 2$, while they increase slightly for orders $d \in \{3, 4\}$ as Hurst parameter H increases. In particular, differences in



(a) Differences using MOPPE.

(b) Differences using MPE-PCA.

Figure 11.2: Differences of MOPPE (left) or MPE-PCA (right) applied to mfBm with different correlation coefficients $\rho_{ij} = \{0.0, 0.8\}$.

MOPPE or MPE-PCA with different correlation coefficients are visible. The difference increases with increasing Hurst parameter H, as the behaviour of the correlating spatial variables converges for large correlation coefficients, reducing overall complexity. Furthermore, the increasing differences for increasing orders d result from increasing maximum entropy values per order d.

11.2 Applications and Future Work

As motivated in Chapter 2, there are several possible applications for ordinal pattern representations. As already mentioned in the previous section, the application of all MPE-variants of order $d = 2$ leads to a constant value. A good distinguishability or separability of different mfBm by using one or more MPE-variants is therefore not given. However, for orders $d > 2$ the application of PPE, MWPE, MPE-PCA or MOPPE to realisations of mfBm provide interesting possible applications.

For example, as discussed in Section 2.1, PPE, MWPE, MPE-PCA and MOPPE are interesting candidates for solving the inverse problem, since they all depend monotonically on H, i.e. the entropy decreases as H increases. Given one or a few realisations of mfBm, the calculation of MPE can provide conclusions about the value of the Hurst parameter H. Since multivariate ordinal patterns involve the dependence of multiple spatial variables on mfBm, MOPPE is promising for estimating the Hurst parameter H and cross-correlations ρ_{ij} from a single source, which the other measures cannot do. However, MOPPE is only practical for a small number of spatial variables m. In this case, it may be helpful to reduce the spatial dimensions beforehand, while encoding the information about the correlation in the lower-dimensional representation, and then use ordinal pattern representations. In this case, MPE-PCA is an interesting candidate. In fact, Part II does not fully solve the inverse problem but focuses on the theoretical relationships that motivate the solution of inverse problems. Experiments need to be evaluated in a detailed study to assess performance, especially compared to other point estimators mentioned in Section 2.1. Similarly, for simplicity, in Part II we have neglected the influence of the anti-symmetry parameter

$\eta_{ij} \in \mathbb{R}$ of mfBm. It is possible that one or more of all the representations presented here is capable of uncovering corresponding information.

In addition, the theoretical results of the representations investigated in this part also motivate the use of MPE as features in a learning task from the field of ML. In classification tasks, for example, it is necessary to extract features that allow good separation of the data. In contrast to PPE, MPE-PCA and MOPPE, MWPE is substantially influenced by strictly ascending and descending ordinal patterns. For this reason, MWPE decreases more sharply than PPE as H increases, providing more expressive representations that may promise better discriminability. Moreover, if one knows that the underlying data are long-range dependent and self-similar, the relationship in the form of MPE does not have to be learned again each time, for example by complex deep learning methods. There are other approaches for the determination of multivariate permutation entropy, specifically MvPE, MPE-EUCL, MPE-MANH or MPE-NORM introduced in Chapter 4. Since their application requires a prior transformation of mfBm and thus is not a canonical extension of the univariate definition, the study of it is not part of this work. Nevertheless, both have the potential to uncover (additional) structures in mfBm, but are left for future work. Similarly, the application of these representations is of great interest not only in the context of mfBm, but for any real-world data sets, which is the focus of Part III of this work.

Part III

Further Applications

In the previous part, we investigated the behaviour of several MPE-variants in the context of the theoretical construct of multivariate fractional Brownian motion. From this, interesting applications have been derived, such as the solution of an inverse problem given, i.e., the derivation (of parameters) of a generating system from one or a few observations, as motivated in Section 2.1. Multivariate ordinal pattern representations can be profitably used not only in the context of model construction or model analysis, but also in other real-world applications involving query answering or prediction in general.

In this last part, we no longer focus only on processes subject to the properties of long-range dependence and self-similarity, but any real time series data from different applications. We expand the ideas from Sections 2.2 and 2.3, and enrich them with the multivariate ordinal pattern representations derived in this dissertation, MWPE, MPE-PCA and MOPPE. In Chapter 12 we consider an example from the research area of lifted probabilistic inference. We use multivariate ordinal patterns to approximate groups of redundant or symmetric objects, a selected representative of which can be used in query answering to reduce computational complexity. In Chapter 13 we consider two examples from the field of supervised learning, i.e., regression and classification.

Chapter 12

Multivariate Ordinal Patterns for Symmetry Approximation

At the heart of many applications are large models that represent our world characterised by uncertainty and complex relational structures with additional temporal information. The more objects a model comprises, the more complex it becomes. However, as the domain of objects increases, so does the likelihood of redundant information being included in the model. As motivated in Section 2.2, exploiting *symmetries* is key to a variety of different challenges in AI in general, e.g. discussed by Dieleman *et al.* (2016); Satorras *et al.* (2021), and in dynamic probabilistic relational models (DPRMs) in particular. Informally speaking, symmetries are groups of objects of redundant information. In the research area of lifted probabilistic inference, a representative of a symmetry group is used to answer queries in order to reduce computational complexity. In this chapter, we use multivariate ordinal patterns to build symmetries. The idea is first presented in the following conference papers, with the first paper focusing on the univariate case and the second paper extending to the multivariate case. Note that Marisa Mohr co-authored the first paper, while the authorship of Nils Finke and Marisa Mohr is equal for the second paper.

> Nils Finke and Marisa Mohr. A Priori Approximation of Symmetries in Dynamic Probabilistic Relational Models. In Stefan Edelkamp, Ralf Möller, and Elmar Rueckert (Eds.), *KI 2021: Advances in Artificial Intelligence*, pages 309–323. Springer, 2021

> Nils Finke, Ralf Möller, and Marisa Mohr. Multivariate Ordinal Patterns for Symmetry Approximation in Dynamic Probabilistic Relational Models. In Guodong Long, Xinghuo Yu, and Sen Wang (Eds.), *AI 2021: Advances in Artificial Intelligence*, pages 543–555. Springer International Publishing, 2022

Although the approach does not depend on any particular model class, we use DPRMs as a reference formalism. Having introduced DPRMs in as much detail as necessary, we extend the motivation of symmetries introduced in Section 2.2 and discuss the challenges that arise in maintaining symmetries in the context of DPRMs. The remainder of this chapter provides **contribution 3a** of this work by introducing MOP_4SA, an approach to approximate symmetries based on MOP encodings and spectral clustering. Understanding symmetrical behaviour by MOP_4SA has several

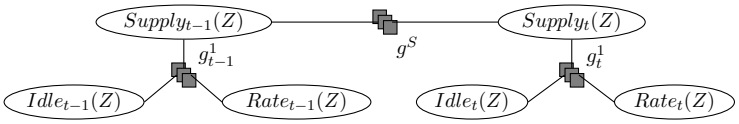


Figure 12.1: Two-slice parameterised probabilistic model.

benefits that we evaluate in an application from logistics. We end with an interim conclusion and future work.

12.1 Reference Formalism: Dynamic Probabilistic Relational Models

In this section, we introduce dynamic probabilistic relational models (DPRMs) as a reference formalism suitable to prove the effectiveness of the following approach. Nevertheless, we refrain from formal definitions of DPRMs as far as possible and only give the intuition with an example from seaborne transportation. For a complete specification, we recommend reading the work of Braun and Möller (2016) for the static case, Gehrke *et al.* (2018) for the dynamic case, and Finke *et al.* (2020) for a discussion of DPRMs in the context of dry-bulk shipping.

In a DPRM, relational logic is combined with a probabilistic representation, where logical variables (logvars) $L_1, \ldots, L_n \in \mathbf{L}, n \geq 0$ are used as parameters for random variables (randvars) R also called parameterized random variables (PRVs) for short and denoted as $R(L_1, \ldots, L_n)$ (Poole, 2003; Gehrke *et al.*, 2018). That is, PRVs compactly represents multiple entities that are considered indistinguishable as a group without further evidence, which is called a lifted representation. To represent independent relations, PRVs are linked by parametric factors (parfactors) to compactly encode the full joint distribution of the DPRM. Like most dynamic model formalisms, DPRMs use two static parameterised models to describe how a model changes from one time step to the next. A DPRM encodes a sequential dimension by a pair of parameterised models, one representing an initial time step and the other representing how the model transitions from one time step to the next. They follow the same idea as dynamic Bayesian networks with an initial model and a copy pattern for further time steps (Murphy, 2002). DPRM are based on the first-order Markov assumption, i.e., randvars from each time slice t depend only on randvars from the preceding time slice $t-1$. DPRMs model a stationary process, i.e., changes from one time step to the next follow the same distribution. The Semantics of DPRMs are given by instantiating a DPRM for a given number of time steps, followed by grounding and building a full joint distribution (Sato, 1995).

Figure 12.1 shows a DPRM illustrating certain aspects of seaborne transportation. Variable nodes (ellipses) correspond to PRVs, factor nodes (boxes) to parfactors. Edges between factor and variable nodes denote relations between PRVs, encoded in parfactors. The parfactor g^S denotes a so-called inter-slice parfactor that separates

the past from the present. The submodel on the left and the one on the right of this inter-slice parfactor are duplicates of each other, with the one on the left referring to time step $t-1$ and the one on the right referring to time step t. Parfactors refers to time-indexed PRVs, namely $Idle_t(Z)$, $Rate_t(Z)$ and $Supply_t(Z)$ with range values such as $\{high, medium, low\}$, built from randvar names $\mathbf{R} = \{\text{Idle, Rate, Supply}\}$ and the logvar name $\mathbf{L} = \{Z\}$. Note that idle is of non-boolean type, but a shortened notation for idle time. Thus, intuitively a PRV represents multiple entities of the same type represented through the logvar, e.g., supply within zones $Z \in \{z_1, z_2, \ldots, z_n\}$. Seaborne transportation is subject to supply chain, i.e., vessels head towards zones Z with supply to load cargo. Certain zones around the globe are of greater interest due to higher freight rates $Rate_t(Z)$, a fee per ton paid to transport cargoes. If zones are overcrowded, idle time $Idle_t(Z)$ can occur, affecting the overall vessel schedule. In the context of a shipping application, an example query for time step $t = 10$, such as $P(Rate_{10}(z_1) \mid Supply_{10}(z_2) = high, Supply_{10}(z_3) = high)$, contains a set of observations $Supply_{10}(z_2) = high$ and $Supply_{10}(z_3) = high$ as evidence.

12.2 Maintaining Lifted Representations

Primarily, in a DPRM, entities having similar or approximately similar behaviour are treated together as a group and represented by a single entity. Lifting a model thus enables a sparse representation of the model, reducing complexity and achieving good performance. However, evidence leads to splits within the symmetrical structures of the models, i.e., asymmetric evidence slowly grounds a lifted model to its propositional counterpart over time. Suppose we ask for the probability distribution of supply at a time step $t = 2$ in a certain zone z_1, given that in the previous time step $t = 1$ the supply was high, i.e., $P(Supply_2(z_1) \mid Supply_1(z_1) = high)$. Then, evidence is encoded in parfactors g_1^1 by duplicating the parfactor and using one to encode evidence and one to represent all sets of entities that are still considered indistinguishable. Each parfactor represents a different set of entities bounded by the use of constraints, e.g., limiting the domain for the evidence parfactor to z_1. The parfactor that encodes evidence is adjusted such that all range value combinations in the parfactors distribution ϕ are dropped for $Supply_1(z_1) \neq high$. During message passing, the splits carry over. Thus, the parfactors g^S and g_1^2 also split into one part for z_1 and another for all other instances.

Specifically, we denote a parfactor g by $\phi(\mathcal{A})_{|C}$ with $\mathcal{A} = (A^1, \ldots, A^n)$ a sequence of PRVs, $\phi : \times_{i=1}^{n} \mathcal{R}(A^i) \mapsto \mathbb{R}^+$ a function with name $\phi \in \Phi$, and C a constraint on the logvars of $\mathcal{A}$. All PRVs are interdependent and are therefore combined by a parfactor, i.e., in our example a parfactor $g^1 = \phi^1(Idle(Z), Rate(Z), Supply(Z))$, indicates the joint probability distribution. For simplicity, we omit the concrete mappings of potentials to range values of ϕ^1. Then, under evidence a model $G_t = \{g_t^i\}_{i=1}^n$ at time step t, is split with respect to its parfactors such that its structure

remains

$$G_t = \{g_t^{i,1}, \ldots, g_t^{i,k}\}_{i=1}^{n} \tag{12.1}$$

with $k \in \mathbb{N}^+$ is the number of splits $g_t^{i,j} = \phi_t^{i,j}(\mathcal{A}^i)_{|C^{i,j}}$, where $1 \leq j \leq k$ and $\mathcal{A}^i$ is a sequence of the same PRVs but with different constraint $C^{i,j}$ and varying functions $\phi_t^{i,j}$ due to evidence.

In the worst case a model is fully grounded, i.e., a model as defined in Eq. (12.1) contains

$$k = \prod_{L \in lv(\mathcal{A})} |L| \tag{12.2}$$

splits for every parfactor g_t^i, where the term $lv(\mathcal{A}^i)$ refers to the logvars L in $\mathcal{A}^i$. Then each object $l \in L$ is in its own parfactor split, and the lifted model is grounded to its propositional model. In particular, performing inference is again extremely costly.

In order to maintain a lifted representation, the field of approximate inference, i.e., approximation of symmetries, has emerged, in which similar but distinguishable objects are treated as if they were identical. While only a small and limited error is introduced, efficient reasoning is obtained in return. Dealing with groundings to recover a lifted representation a posteriori has been studied extensively in the literature (Singla *et al.*, 2014; Venugopal and Gogate, 2014; den Broeck and Niepert, 2015; Gehrke *et al.*, 2020). Common to all these approaches is that in the first place groundings are allowed and then dealt with afterwards, i.e., by exploiting symmetries within message passing in inference tasks. While undoing splits a posteriori, also known as **taming** (Gehrke *et al.*, 2020), is reasonable, we propose **learning** symmetries a priori to **prevent** unnecessary splits due to inaccuracy or one-time events. The combination of both methods provides a powerful tool in the research field of lifted probabilistic inference.

12.3 MOP$_4$SA: An Approach for Symmetry Approximation

In previous research, it is assumed that lifted graphical models already contain symmetries that do not even need to be constructed, i.e., they have defaults. As far as we know, we are the first to propose a priori symmetry construction, i.e., to learn intrinsic defaults. Note that this idea opens up a whole new branch of research that we cannot deal with in detail in this work. Rather, we are interested in demonstrating the possibilities and value of multivariate ordinal patterns.

In the following, we propose an approach that uses **M**ultivariate **O**rdinal **P**attern for **S**ymmetry **A**pproximation, **MOP$_4$SA** for short. In the special framework of DPRMs, symmetries of multivariate objects are approximated, i.e., entities represented by PRVs and linked in a parfactor. For this purpose, first the behaviour of the multivariate time-dependent objects is encoded by MOPs. Subsequently, symmetries

of MOPs are approximated via spectral clustering. Based on these approximate symmetry clusters, we get a better understanding of potential symmetric behaviour and gain several advantages, which we evaluate in two use cases in the next section.

12.3.1 From DPRMs to Time Series

DPRMs are usually factorised with respect to a full joint probability distribution by exploiting (conditional) independencies between randvars that evolve over time. As such, DPRMs encode multivariate time series data through a set of dependent randvars. The notion of symmetry of (multivariate) time series has indeed been discussed extensively in the literature in the context of similarities (see Section 1.1). The direct search of similarities between multivariate time series is computationally intensive due to the high dimensionality of the time series itself, and additionally complex due to many entities represented in a lifted model. While the detection of exact symmetric behaviour, i.e., two or more entities show exactly the same behaviour, by comparing all values of a time series is computationally expensive and often not realistic, approximate symmetric behaviour with a bounded error is usually desired. That is, the behaviour of two or more entities is approximately the same over time. Symbol-based approaches such as MOP encode the multivariate time series observations as sequences of symbolic abstractions that match with the shape or structure of the time series.

12.3.2 Encoding Object Behaviour by MOPs

To find symmetries of multivariate objects, we use evidence to encode a models' entity behaviour w.r.t. a context, i.e., w.r.t. a parfactor. In particular, this means: Every PRV represents multiple entities, e.g., zones Z, of the same type. That is, for a PRV $Supply_t(Z)$, entities z are represented by a logvar Z with domain $\mathcal{D}(Z)$ and size $|\mathcal{D}(Z)|$. Note that a PRV can be parameterised with more than one logvar, but for the sake of simplicity we introduce our approach using PRVs with only one logvar. Symmetry detection for m-logvar PRVs works similarly to one-logvar PRVs, with the difference, that in symmetry detection, entity pairs, i.e., m-tuples, are used. As an example, for any 2-logvar PRV $P_t(A, B)$, an entity pair is a tuple (a_1, b_1) with $x_1 \in \mathcal{D}(A)$ and $y_1 \in \mathcal{D}(B)$.

A DPRM, as introduced in Section 12.1, encodes temporal data by unrolling a DPRM while observing evidence for the models PRVs, e.g., the PRV $Supply_t(Z)$ encodes supply at time t in various zones Z on the globe. In addition, a DPRM exploits (conditional) in-dependencies between randvars by encoding interdependencies in parfactors. As such, parfactors describe interdependent data through its linked PRVs, e.g., the correlation between supply $Supply_t(Z)$, freight rates $Rate_t(Z)$ and idle times $Idle_t(Z)$ within a common zone Z encoded by the parfactor g_t^1. For each entity $z_j \in \mathcal{D}(Z)$ from the PRVs $Supply_t(Z)$, $Rate_t(Z)$ and $Idle_t(Z)$ observations are made over time, i.e., a time series $((x_t^i)_{i=1}^3)_{t=1}^T$ is generated. In this work, the time series is to be assumed multivariate, containing interdependent variables, i.e.,

$m > 1$. Note that for the special case $m = 1$ univariate ordinal patterns from Definition 3.3.1 are used. Further details can be found in the conference paper by Finke and Mohr (2021). Having $|\mathcal{D}(Z)|$ entities in Z, we consider $|\mathcal{D}(Z)|$ samples of multivariate time series $X = (((x_t^i)_{i=1}^m)_{t=1}^T)_{j=1}^{|\mathcal{D}(Z)|} \in \mathbb{R}^{m \times T \times |\mathcal{D}(Z)|}$, e.g., for $m = 3$ with observations $(x_t^1, x_t^2, x_t^3) = (Supply_t(z_j), Idle_t(z_j), Rate_t(z_j))$ for every $z_j \in \mathcal{D}(Z)$ in time $t \in \{1, \ldots, T\}$. As such, a multivariate time series is defined for several PRVs linked in a parfactor. Identification of entity symmetries is performed on a set of observed multivariate time series that are related to the parfactor g^1 (see Fig. 12.1).

Now, for each time step $t = \tau(d-1)+1, \ldots, T$ of a multivariate time series X, MOP is determined as described in Section 5.3. However, ordinal patterns are well suited to characterise an overall behaviour of time series that is independent of the data range, but the dependence on the data range can be also relevant, i.e., time series can be similar in terms of their ordinals patterns, but differ considering their y-intercept. In other words, transforming a sequence $x = (x_t^i)_{a \leq t \leq b}$ as $y = x + c$, where $c \in \mathbb{R}$ is a constant, should change y's similarity to other sequences, although the shape is the same. To address the dependence on the data range, we use the arithmetic mean

$$\overline{x}_t^{d,\tau} = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{d} \sum_{k=1}^{d} x_{i,t-(k-1)\tau} \tag{12.3}$$

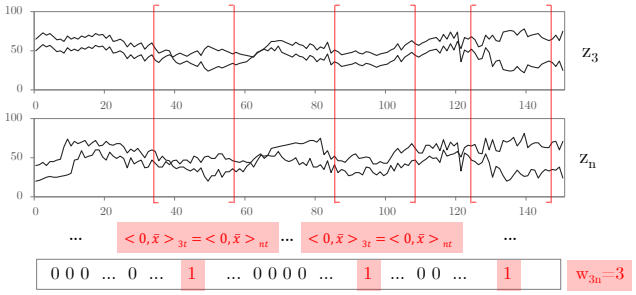
of the multivariate time series' values corresponding to the ordinal pattern, where $x_{i,t-(k-1)\tau}$ is min-max normalised, as an additional characteristic or feature of behaviour. If one of the variables changes its behaviour significantly along the intercept, the arithmetic mean uncovers this. There are still other features that can be relevant. For simplicity, we only determine ordinal patterns and their means for each parfactor g^1 with, e.g., PRVs ($Supply_t(Z)$, $Idle_t(Z)$, $Rate_t(Z)$), yielding a new symbolic representation

$$\mathcal{X} = \langle o, \overline{x} \rangle^{(T-(\tau(d-1)) \times |\mathcal{D}(Z)|} \tag{12.4}$$

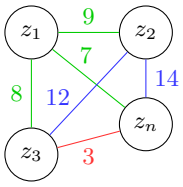
where $\langle o, \cdot \rangle_{tj}$ represents the MOP and $\langle \cdot, \overline{x} \rangle_{tj}$ represents the corresponding mean $\overline{x}_t^{d,\tau}$ for entity z_j at time step t. The order d and delay τ are passed in from the outside and might depend on, e.g., the frequency of the data, to capture long-term behaviour of each entity.

12.3.3 MOPs for Symmetry Approximation by Spectral Clustering

Based on the derived representation in Eq. (12.4), we perform symmetry approximation by clustering entities. Since lifted models are specifically designed for encoding large domains, we use spectral clustering, a popular setting for handling high-dimensional data (Bertozzi and Merkurjev, 2019).



(a) Similarity counting based on MOP and means.



(b) Similarity graph $\mathcal{W}$.

Figure 12.2: (a) Similarity count for w_{3n} (red) and (b) resulting similarity graph $\mathcal{W}$.

Similarity Graph

To perform spectral clustering on the symbolised representation $\mathcal{X}$ of the multivariate time series X, similarities have to be determined. The corresponding similarity graph of the symbolised representation $\mathcal{X}$ contains nodes for each entity represented by PRVs connected within a parfactor. Specifically, the similarity graph for a parfactor g_t^1 connecting the PRVs $Supply_t(Z)$, $Idle_t(Z)$ and $Rate_t(Z)$ contains one node for each entity $z \in \mathcal{D}(Z)$ observed in form of multivariate time series. The edges of the similarity graph represent the similarity between two nodes, or more precisely, how closely related two entities of the model are. To measure similarity, we use the symbolic representation $\mathcal{X}$, which contains tuples of multivariate ordinal numbers and mean values that describe the behaviour of an entity. The similarity of two entities z_i and z_j is given by counts w_{ij} of equal behaviours, i.e.,

$$w_{ij} = \sum_{t \leq T} \Big[\langle o, \cdot \rangle_{it} = \langle o, \cdot \rangle_{jt} \ \wedge \ |\langle \cdot, \overline{x} \rangle_{it} - \langle \cdot, \overline{x} \rangle_{jt}| < \delta \Big], \tag{12.5}$$

where $[x] = 1$ if x is true, and 0 otherwise.

Simply put, as visualised in Figure 12.2a, one counts the time steps t at which both multivariate time series of z_i and z_j have the same MOP and the absolute difference of the mean values of the corresponding MOPs is smaller than $\delta > 0$. Finally, as shown in Figure 12.2b the counts w_{ij} correspond to the weights of the edges in the similarity graph $\mathcal{W}$, where zero indicates no similarity between two entities, while the larger the count, the more similar two entities are.

Spectral Clustering

In the worst case, a similarity graph, representing the similarity of entities $z \in \mathcal{D}(Z)$, contains $\binom{|\mathcal{D}(Z)|}{2}$ fully-connected nodes. If the dimension of the similarity graph, due to the potentially large domain of a lifted model, becomes too large, classical clustering methods do not achieve good results due to the curse of dimensionality (Bellman, 2015). Spectral clustering involves dimensionality reduction in advance

before using standard clustering methods such as k-means or similar methods. For dimensionality reduction, the similarity graph $\mathcal{W}$ is transformed into the so-called graph Laplacian matrix $\mathcal{L}$, which describes the relations of the nodes and edges of a graph, where the entries are defined by

$$\mathcal{L}_{ij} := \begin{cases} \deg(z_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } w_{ij} > 0\,, \\ 0 & \text{else} \end{cases} \tag{12.6}$$

where $\deg(z_i) = \sum_{j=1}^{|\mathcal{D}(Z)|} w_{ij}$. For decorrelation, the data in the graph Laplacian matrix $\mathcal{L}$ are decomposed into its sequence of eigenvalues (spectrum) and the corresponding eigenvectors. The eigenvectors form a new uncorrelated orthonormal basis and are thus suitable for standard clustering methods. The observations of the reduced data matrix whose columns contain the smallest k eigenvectors can now be clustered using k-means. Considering the reduced data matrix whose columns contain the smallest k eigenvectors, the rows are used for clustering by, e.g., k-means. An observation assigned to cluster $\mathcal{C}_i, i = 1, ..., k$ can then be traced back to its entity $z \in \mathcal{D}(Z)$ by indices. That is, entity symmetry clusters $\mathcal{C}(g_t^1) = \bigcup_{i=1}^{k} \mathcal{C}_i$ are built for parfactor g_t^i with each $\mathcal{C}_i$ containing a subset of entities $z \in \mathcal{D}(Z)$. As symmetry clustering is done individually for each parfactor, $\mathcal{C}$ denotes the set of all entity symmetry clusters for all parfactors. Algorithm 6 presents the corresponding pseudocode. After learning the clusters, we discuss how the clusters can be used, for example, to prevent a DPRM from grounding, in the next section.

12.4 Practical Considerations of MOP$_4$SA

As already mentioned at the beginning, the introduction of MOP$_4$SA opens a new branch of research in the field of lifted probabilistic inference, which we cannot fully discuss within the scope of this work. Nevertheless, motivated by examples from a logistics application, we show that approximating multivariate symmetries is key for a variety of different challenges in DPRMs.

12.4.1 Preventing Groundings

Reasoning in lifted representations has a polynomial complexity in the domain size of the model (Niepert and Van den Broeck, 2014). This means that finding objects in the model domain that behave (almost) identically is the key to avoiding duplicate computations in inference, as computations only need to be performed for one representative of similarly behaving entities. For this reason, we evaluate the use of MOP$_4$SA to obtain a lifted solution of the model and to prevent a DPRM from being grounded. Assuming a symmetry cluster contains entities z_1, z_2 and z_3, groundings occur whenever observations differ across entities in a symmetry cluster, e.g., when

Algorithm 6: MOP$_4$SA

Input: Parfactor g, Evidence $\mathbf{E}_{0:T}$, Order d, Delay τ, Delta $\delta_{\bar{x}}$, Number of Clusters k

Function `MOP4SA`$(g, E_{0:T}, d, \tau, \delta_{\bar{x}}, k)$**:**

- $X^{|A|\times T} \leftarrow$ get all evidence from $\mathbf{E}_{0:T}$ concerning PRVs in g
- $\mathcal{X}^{|A|\times(T-\tau(d-1))} \leftarrow$ symbolic representation matrix initialised with zeros
- **for** *every dimension* $i = 1, \ldots, |A|$ *of* $\mathcal{X}$ **do**
  - $\mathcal{X}_{i\cdot} \leftarrow$ create a time series of tuples with $\langle ordinal, mean \rangle$
- $\mathcal{W}^{|A|\times|A|} \leftarrow$ similarity-matrix initialised with zeros
- **for** *every time step* t *of* $\mathcal{X}_{\cdot t}$ **do**
  - $w_{ij} \leftarrow$ do similarity counting
- $L \leftarrow$ Calculate the graph Laplacian matrix $\mathcal{L}$
- $\mathcal{C} \leftarrow$ Perform spectral clustering based on the k eigenvectors of $\mathcal{L}$
- **return** $\mathcal{C}$

- the observation $(Supply_1(z_1) = high, Idle_1(z_1) = high)$ of entity z_1 differs from observations $(Supply_1(z_i) = low, Idle_1(z_i) = mid)$ of entities z_i for $i = 2, 3$, or
- observations are made for only a subset of the entities, i.e., for entities z_2 and z_3 observations are made but not for entity z_1.

In both cases, the entities z_2 and z_3 would split off from their initial symmetry group, and are henceforth treated individually in exact inference. That means that DPRMs dissolve into ground instances through asymmetric evidence. However, differences introduced through evidence might be minimal or overcome by model behaviour over time (Gehrke *et al.*, 2020). Entity symmetry clusters, as created in Section 12.3.3, contain entities that all behave approximately the same over time and therefore can be used to avoid model splits to some extent by applying evidence to other entities in their group. The idea is to align evidence across all entities in the cluster to prevent the model from grounding. Since the entities of the cluster have behaved similarly before, it is assumed that they continue to behave similarly. The learned symmetry clusters are used as intrinsic default, i.e.,

(a) if evidence is observed only for a subset of the entities, evidence is also applied to the other entities of the symmetry cluster, or

(a) if evidence differs between the entities of a cluster, the most frequently observed observation is also applied to the other entities of the symmetry cluster and all contrary evidence is discarded.

We compare runtime advantages obtained by using symmetry clusters and the accuracy in lifted inference with and without the use of symmetry clusters. To setup a DPRM as shown in Figure 12.1, we use historical vessel movements from 2020 based on automatic identification system (AIS) data provided by the Danish Maritime

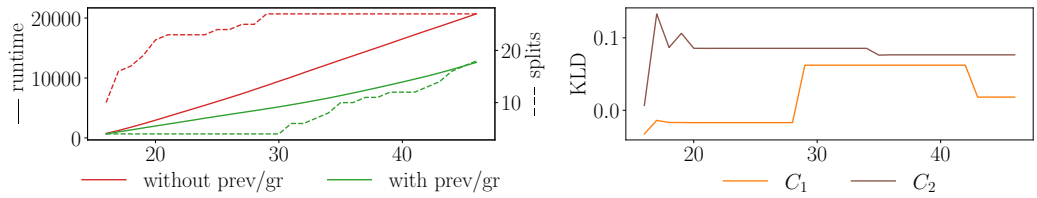


Figure 12.3: Runtime [ms] and splits [#] w/ and w/o preventing groundings and KLD.

Authority for the Baltic Sea (see Appendix A.2). Each AIS signal contains the current geo-position and the total cargo quantity of a vessel. Preprocessing for retrieving variables *Supply* and *Idle* for 367 defined *Zones* can be found on GitHub[1]. To construct symmetry clusters, we use Alg. 6 with MOP$_4$SA$(g, E_{0:15}, 2, 1, 0.1, 12)$.

Inference in DPRMs is performed by the lifted dynamic junction tree algorithm. Details are presented by Gehrke *et al.* (2018). More specifically, we perform inference by answering prediction queries $P(Supply_t(Z), Idle_t(Z))$ for each time step $t \in \{16, \ldots, 45\}$ and obtain a marginal distribution for each entity $z \in \mathcal{D}(Z)$. Figure 12.3 (left) shows runtime (solid) and number of splits (dotted) to answer queries for each time step. The two red lines, i.e., for answering queries without preventing groundings, are greater than the green ones with preventing groundings by means of MOP$_4$SA underpinning the runtime advantage. Figure 12.3 (right) shows Kullback Leibler divergence (KLD) as a measure of the accuracy of the predictions, exemplary for two constructed symmetry clusters $C_1 = \{z_5, z_{73}, z_{109}\}$ and $C_2 = \{z_{26}, z_{98}, z_{223}, z_{241}, z_{356}\}$. KLD compares the predicted probability distributions of the individual entities $z \in \mathcal{D}(Z)$ respectively with and without preventing groundings. More precisely, for discrete probability distributions P and Q defined on the same probability space $\mathcal{X}$, KLD is defined as the relative entropy of Q to P, i.e.,

$$D_{\mathrm{KL}}(P, Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right). \tag{12.7}$$

Here, the average KLD over all entities for each time step, denotes the overall accuracy for a time step. A KLD close to 0 is indicative of similar distributions, thus corresponds to a small error. Summing up, Figure 12.3 shows that the a priori introduction of symmetry clusters to prevent groundings by MOP$_4$SA speeds up inference while introducing only a small error in inference.

With MOP$_4$SA, we propose learning approximate model symmetries *a priori* to relieve the model from unnecessary splits due to inaccuracy or one time events. As already mentioned in Section 12.2, there exist some procedures to undo splits *a posteriori*. For example, Gehrke *et al.* (2020) propose **taming**, i.e., recreating a new lifted representation by merging groundings, which were introduced over time. As we argue, combining both kind of approaches brings together the best of both worlds:

[1] https://github.com/FinkeNils/Processed-AIS-Data-Baltic-Sea-2020

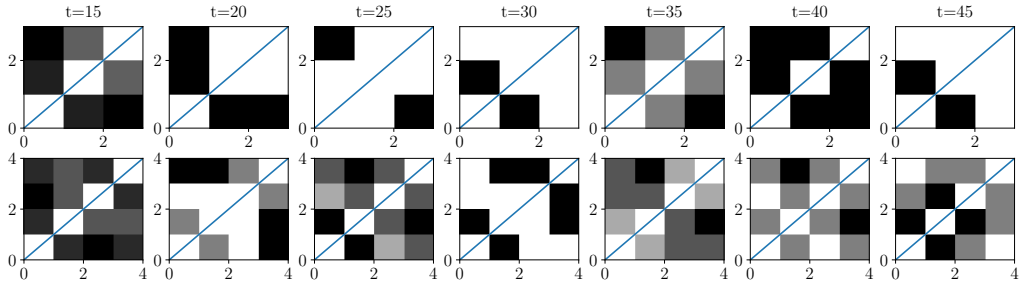


Figure 12.4: Heatmaps of the similarity graph of clusters C_1 (top) and C_2 (bottom) at different time steps t.

(a) While with determining approximate model symmetries a priori, we can use the full amount of historical training data to prevent groundings,

(b) with temporal approximate merging, we can merge non-preventable parfactor splits even after they occurred.

12.4.2 Detecting Structural Changes

Symmetries between entities can change over time. In the worst case, they can dissolve. Knowing changing symmetry structures is important when approximating symmetries in order not to compromise too much on accuracy in inference and can be detected as direct outcome of MOP$_4$SA. Figure 12.4 shows heatmaps of two symmetry clusters that relate the entities of a cluster based on their similarity count over several time steps. The heatmap at $t = 15$ shows the similarity at the time of learning clusters, while the heatmaps for all other time steps $t > 15$ show only the difference in similarity counts to the previous time step. That means that for $t = 20$ the heatmap is based on similarity counts between $t = 15$ and $t = 20$ only. Figure 12.4 (top) shows a symmetry cluster of which entities quickly drop out, but then at $t = 35$ entities behave similar again. Figure 12.4 (bottom) shows a symmetry cluster containing entities, which share similarities constantly over time with only a few exceptions for the relation between entities 1 and 3 and 2 and 3, which do not relate for $t = 20$ and $t = 30$. Both heatmaps indicate periodicities, while also giving insights about the validity of the cluster over time. In the context of MOP$_4$SA, the similarity graph builds up over time and also changes over time, i.e., it can be used to detect structural changes in symmetries. If the similarity graph changes over time in a constant and balanced way, symmetry clusters stay valid. If the similarity graph changes over time in an unbalanced manner, i.e., if similarity counts change significantly, there is a change in the structure of the symmetry clusters. Finke and Möller (2022) expand the idea of change detection in symmetry clusters and provide further discussions with suitable metrics and evaluations.

12.4.3 MOP$_4$SA for Further Model Classes and Applications

As the application of MOP$_4$SA is performed on observed time series of a DPRM, the approach is suitable for the symmetry approximation of arbitrary multivariate time series. Practicality and performance in context of other models need to be evaluated. Depending on the applications, there are parts of MOP$_4$SA that can be changed. Even if the choice of ordinal patterns is justified, limitations exist in the consideration of the data ranges. As a remedy, we have added the arithmetic mean $\overline{x}$ as a second feature. Whether further or other representations increase the accuracy is a question of feature selection and its underlying application. Similarly, even if the choice of spectral clustering including k-means is reasonable, there exist numerous other clustering algorithm that – in dependence of the application – may perform better.

12.5 Interim Conclusion: MOP$_4$SA

Exploiting symmetries is an important topic to obtain benefits in lifted inference in DPRMs. In this chapter, we have proposed MOP$_4$SA, an approach to approximate multivariate model symmetries in DPRMs. For the determination of symmetries, this novel approach uses spectral clustering based on similarity graphs of multivariate ordinal patterns representing the up and downs in an observed multivariate time series of an entity. By **learning** symmetry clusters with MOP$_4$SA as intrinsic defaults, we show that groundings in a DPRM can be **prevented** and runtime advantages in lifted inference can be obtained while accuracy remains good. Furthermore, we motivate various applications as well as extensions of MOP$_4$SA in the research area of lifted probabilistic inference, which we cannot fully discuss within the scope of this work. However, we show great potential for the use of multivariate ordinal patterns.

Chapter 13

Multivariate Permutation Entropy in Supervised Learning

As discussed in Section 2.3, a good prediction depends on both, the choice of features, as well as the choice of a model. In this chapter we show that ordinal pattern representations are appropriate features in supervised learning tasks. In a regression task from manufacturing, we compare the performance of entropies with respect to other prominent features classes. The entire case study is first presented in the following conference paper. Note that Nils Finke and Marisa Mohr contributed equally to this work.

> Nils Finke, Marisa Mohr, Alexander Lontke, Marwin Züfle, Samuel Kounev, and Ralf Möller. Recommendations for Data-Driven Degradation Estimation with Case Studies from Manufacturing and Dry-Bulk Shipping. In Samira Cherfi, Anna Perini, and Selmin Nurcan (Eds.), *Research Challenges in Information Science*, pages 189–204. Springer, 2021

In several classification tasks, we evaluate which of the MPE-variants presented in this work can be used flexibly and reliably and leads to good predictions on different real-world data. The experiments are first presented in the following conference paper.

> Marisa Mohr, Florian Wilhelm, Mattis Hartwig, Ralf Möller, and Karsten Keller. New Approaches in Ordinal Pattern Representations for Multivariate Time Series. In *Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference (FLAIRS-33)*, pages 124–129, 2020

The remainder of this chapter provides the last **contribution 3b** of this work. First, we start with a short introduction to regression before presenting a case study from the field of predictive maintenance that underpins the effectiveness of PE as a feature. Second, we present a small introduction to classification before solving several classification tasks from the multivariate time series classification (MTSC) archive using MPE, provided by the University of East Anglia (UEA), and determine the most efficient variants. Both parts of this chapter are followed by an interim conclusion involving future work.

13.1 Regression as a Learning Task

Regression models are used for simple output prediction. Basically, a regression algorithm learns a relationship between input (feature representations or predictors) and output (target or label) data and uses this relationship to make predictions on new, unseen observations. Usually, the relationship is expressed by output $Y_i \in \mathbb{R}$ of an observation i being a function of the input X_i, a parameter β and an additive noise ε_i, i.e.,

$$Y_i = f(X_i, \beta) + \varepsilon_i. \tag{13.1}$$

The goal is to estimate the function f that best fits the available data. Here, the prediction Y_i depends not only on the choice of the regression model f, but also on the features or representation of the data X_i used for training.

In the following, we briefly present the idea of four regression models. We have limited ourselves here to the most popular ones. Further background information can be found in popular books such as "The Elements of Statistical Learning" by Hastie *et al.* (2009).

Multiple Linear Regression MLR is a statistical technique that fits an observed dependent variable by several independent variables using the method of least squares. More precisely, the coefficients $w_1, \dots w_K$ of a linear function $y_t = x_{t1}w_1 + x_{t2}w_2 + \cdots + x_{tK}w_K + \varepsilon_t = \mathbf{x}_t^\top \boldsymbol{w} + \varepsilon_t, t = 1, 2, \dots, T$, are estimated, where y is the response variable, x_K are the predictors, and w the coefficients of the model. Basis functions are used to model the functional relationship, e.g., $y_t = \varphi(\mathbf{x}_t^\top)\boldsymbol{w} + \varepsilon_t$ with polynomial basis function $\varphi(x) \in \mathbb{R}^{d_\varphi}, [1, x] \xrightarrow{\varphi} [1, x, x^2, \dots, x^d]$.

Gaussian Process Regression In a traditional regression model, we infer a single function f, used in $Y = f(X)$. In Gaussian process regression (GPR), we place a Gaussian process over $f(X)$. A Gaussian process (GP) is a collection of random variables, of which any finite subset of random variables is Gaussian distributed. It is completely specified by its mean $\mu = m(x) = \mathrm{E}[f(x)]$ and its covariance or kernel function $k(x, x') = \mathrm{E}[(f(x) - m(x))(f(x') - m(x'))]$. As such, GP describes a distribution over possible Gaussian density functions. The chosen kernel k (e.g. periodic, linear, radial basis function) that describes the general shapes of the functions, defines a prior distribution of $f(X)$. This corresponds roughly to the selection of the degree of a polynomial function for the regression. Placing the Gaussian prior over $f(X)$ yields a posterior joint distribution being used to determine the future process.

Linear Network Regression An artificial neural network (ANN) can pretend to be any type of regression model. The output of an ANN is based on the activation function between input and output layer. As an ANN is mainly used for classification, the sigmoid function is used as a popular activation function, whereas when using an ANN to solve a linear regression problem, the activation function is chosen as linear equation $y = w_0 + w_1 x_1 + \cdots w_n x_n$.

Support Vector Regression Support vector regression (SVR) is based on similar principles as SVM for classification, identifying the optimal support vectors of a hyperplane that separates the data into their respective classes. Instead of separating classes, SVR fits a hyperplane describing the training data best. To solve the optimisation problem of finding the best hyperplane, the coefficient vector of the hyperplane is minimised – in contrast to ordinary least squares fitting where the squared error is minimised. Instead the squared error term is handled in the constraints allowing a certain error range ϵ, i.e., $\min \frac{1}{2}\|w\|^2$ s.t. $|y_i - w_i x_i| < \epsilon$.

13.2 PE for Degradation Estimation in Manufacturing

In this section, we present a general approach for estimating the degradation of a machine part and its remaining useful lifetime through regression. To challenge the effectiveness and the relationships between the extracted features such as entropies and the regression model, we conduct several experiments in a comprehensive case study from manufacturing.

Challenge. Data-driven products and machine learning methods offer large benefits for production engineering companies. Predictive maintenance can reduce the risk of unwanted production and operational downtime and help keep machines in optimal condition. One key challenge is to estimate the remaining useful lifetime (RUL), that is, predicting the time to failure. However, the development of appropriate machine learning methods requires a large initial investment in the model definition and training data acquisition. The latter is especially important as the prediction quality of a machine learning model is largely determined by the data used for training. In real-world challenges, as in case of large expensive machines, large amounts of labeled run-to-failure data are extremely rare. Of course, one could deliberately degrade machines to capture more failure patterns, but that is at least financially irresponsible. For this reason, we focus on *degradation models* to model the RUL without having labeled or complete failure data from similar machines. In Appendix B.2.1, two alternatives for modelling the RUL are presented. Degradation models estimate the RUL indirectly by relating the degradation of parts of the product itself to the failure mechanisms. In order to determine the current level of degradation of a machine part, data are usually used that are recorded by sensors of a machine over time, i.e., time series. Classical data-driven models, i.e., machine learning models such as multiple linear regression or artificial neural networks, cannot process time series directly as they require (time-)independent observations. Thus, it is necessary to extract scalar-valued representations (features) from time series before using these algorithms. In case the algorithms can process time series data directly, such as autoregressive integrated moving average (ARIMA) models or long-short-term-memory (LSTM) networks, time series representations or an adequate preprocessing result can be more useful than the raw time series data to detect the degradation. Consequently, feature extraction is a crucial part of the prediction process and is closely

related to the regression model.

Research provides numerous methods for modelling degradation and RUL. To decide on an appropriate approach, there are few or insufficient comparisons of existing methods. To help deciding on a solution for real-world challenges, one needs a mechanism to compare existing methods. To demonstrate feasibility, one is interested to setup a basic solution before improving the overall approach by, e.g., improving the accuracy of the prediction. In the following, we present a general data-driven approach for predicting RUL that considers comparability of existing approaches in the best possible sense.

Degradation Estimation Process. In general, the degradation estimation process consists of six technical steps, i.e., time series data acquisition, health stage classification, frequency analysis, feature or indicator extraction, degradation estimation by regression, and deployment and integration as presented in Fig. 13.1. Data acquisition (**step 1**) such as an appropriate deployment and integration of the predictive model (**step 6**) depend on both the domain and the user's system infrastructure, so we do not go into detail here. Health stage classification (**step 2**) and frequency analysis (**step 3**) are optional and do not directly address the problem of this work. For background on these two optional steps, see Appendices B.2.3 and B.2.4.

Instead, in the context of this work and by means of the approach, we focus on the following research questions that arise in the search for a suitable degradation estimation process:

(**RQ3**) Which feature sets are appropriate for the estimation of degradation?

As the choice of features affects the accuracy of the regression model, the following question, i.e.,

(**RQ4**) Which data-driven regression method yields the highest accuracy?

is directly related to **RQ3** and is considered in this section as well. Further research questions (RQ1 and RQ2) related to the problem and established in the original conference paper can be found in Appendix B.2.2. Since, in this work, we limit ourselves to two specific research questions, we only address feature extraction (**Step 4**) and degradation regression (**Step 5**) as parts of the degradation process in the following.

The extraction of features in **Step 4** visualised in Figure 13.1 refers to the creation of new information from time series, which is time-independent and was not previously available. Techniques for feature extraction can be classified into two groups, namely feature engineering and feature learning. Feature engineering is the older discipline of the two. New features are created by processing domain-specific knowledge or by transforming data. Techniques for feature engineering origin from at least two research areas. The first way to extract features is by means of statistical analysis. Classic statistical measures used in related work include root mean square and

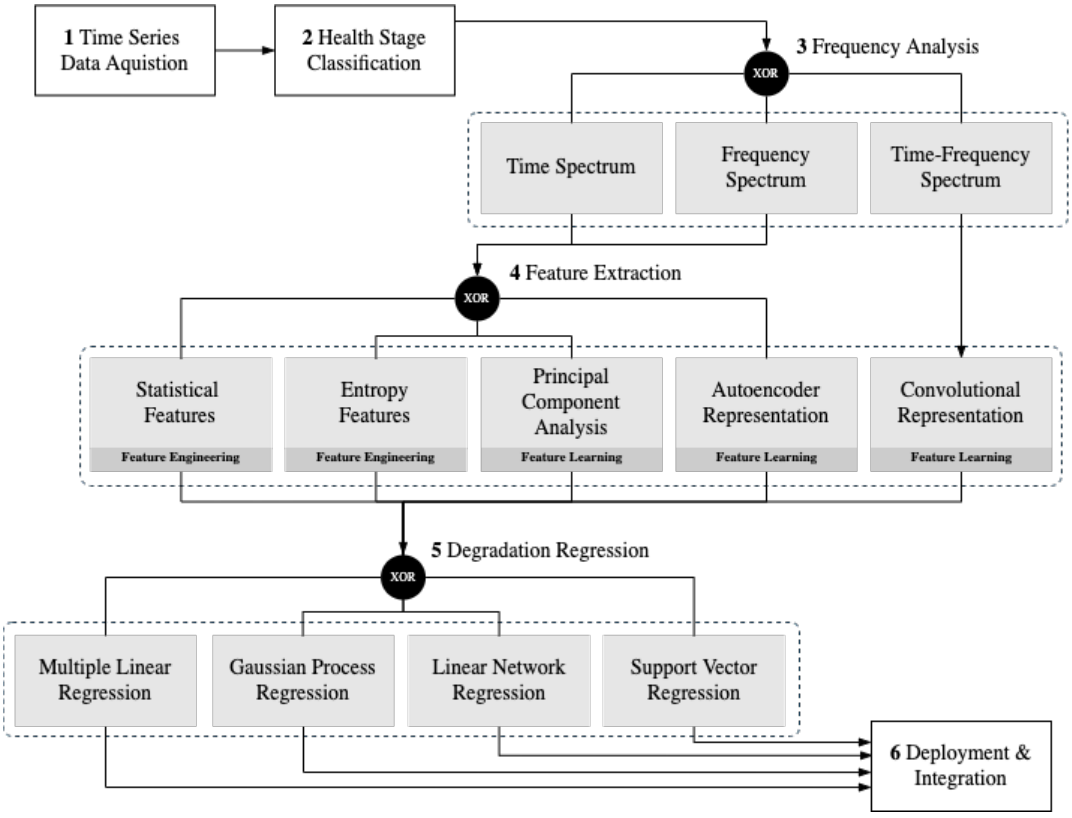


Figure 13.1: Six technical steps of RUL prediction (2 and 3 are optional).

kurtosis (Du *et al.*, 2012; Ahmad *et al.*, 2019; Pan *et al.*, 2020). A complete list of the *statistical features* for a univariate time series used in this work is presented in Appendix B.2.5. Another way of extracting features is by using information-theoretic measurements, i.e., entropies, as used by Boskoski *et al.* (2015); Zhang *et al.* (2016); Wang *et al.* (2019); Kim and Nam H. (2017). In this example, as entropy features, we specifically use Shannon entropy based on SAX and univariate PE. All statistical and entropy features can be applied directly to time or frequency spectrum. Compared to feature engineering, feature learning solve optimisation problems to learn features from a set of time series. Learned features can reveal task-specific patterns that are not obvious to humans, including non-linear patterns. There are numerous ways to learn feature representations, for example by using principal component analysis (see Section 5.2), autoencoders, or convolutional neural network (CNN) (see Appendices B.2.6 and B.2.7). Then, in **Step 5**, regression models, as one of the most popular data-driven techniques for RUL prediction, fit the derived features by regression functions. Related work in the context of data-driven models for degradation estimation includes polynomial regression (Loukopoulos *et al.*, 2019; Wu *et al.*, 2019), SVR (Kim *et al.*, 2012), or ANN for regression (Wang *et al.*, 2019). Furthermore, in this case study, we use the regression models introduced in the previous section,

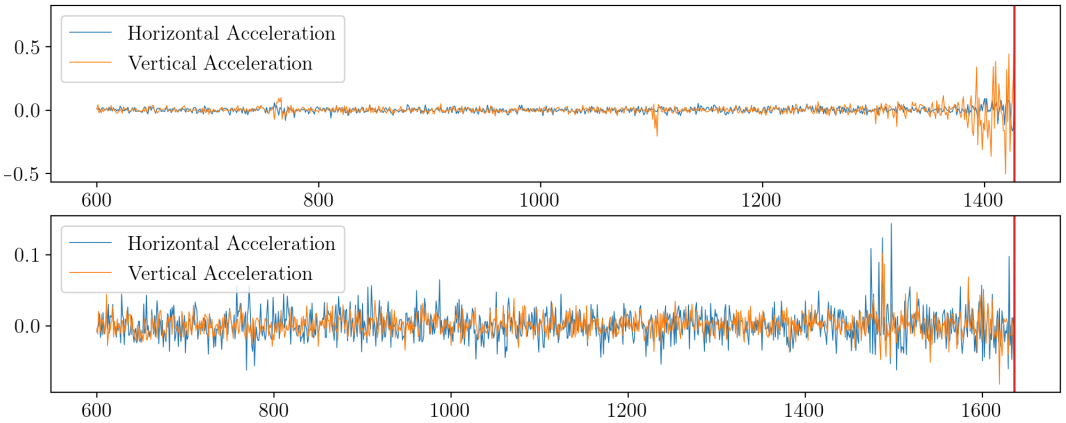


Figure 13.2: Horizontal and vertical acceleration (vibration) of bearings b_{14} (top) and b_{32} (bottom). The red line indicates end of useful lifetime.

namely MLR, GPR, linear network regression (ANN) and SVR.

Experimental Results. The dataset used for the case study is the well-known FEMTO bearing dataset that contains run-to-failure tests of 17 bearings each with time series data of vibration acceleration along the horizontal and vertical dimension. Degradation itself corresponds directly to increasing vibrations. Details such as sample rates or conditions can be found in Appendix A.1. Figure 13.2 visualises the horizontal and vertical vibration acceleration over time for two bearings.

In total, we conduct 104 experiments, whereas each experiment results from the combinations of the components visualised in Figure 13.1. Note that technically not all combinations of components from the frequency analysis and feature extraction step are possible, thus, we denote them explicitly as follows. We choose $Z_{i,j}$ to be an experiment, where $Z \in \{A, \ldots, M\}$ denotes a combination of preprocessing steps listed in Table 13.1, $i \in \{\text{true}, \text{false}\}$ denotes if the health stage classifier is used, and $j \in \{\text{MLR}, \text{GPR}, \text{ANN}, \text{SVR}\}$ denotes the selected regression model for prediction. For example, the notation $B_{\text{false,MLR}}$ represents no application of health stage classifier, followed by entropy feature extraction and multiple linear regression.

Note that the general aim of the following case study was not to achieve the best possible accuracy in prediction, but to investigate the relation of different components. For example, accuracy could be improved by performing hyperparameter tuning or appropriate feature selection methods, which we leave out on purpose. Instead, we investigate the interaction of steps and show that simple features from encodings of up and down movements can be equally successful in terms of RUL prediction accuracy compared to complex learned feature representations, for example using complex CNNs. Implementation details such as selected architectures or chosen (hyper-)parameters can be found in Appendix B.2.8. Each experiment is trained on a

	Statistical Features	Entropy Features	PCA	Autoencoder	Statistical+ PCA	Statistical+ Autoencoder	CNN
Time Spectrum	A	B	C	D	E	F	–
Frequency Spectrum	G	H	I	J	K	L	–
Time-Freq. Spectrum	–	–	–	–	–	–	M

Table 13.1: All combinations of preprocessing steps used in the case study.

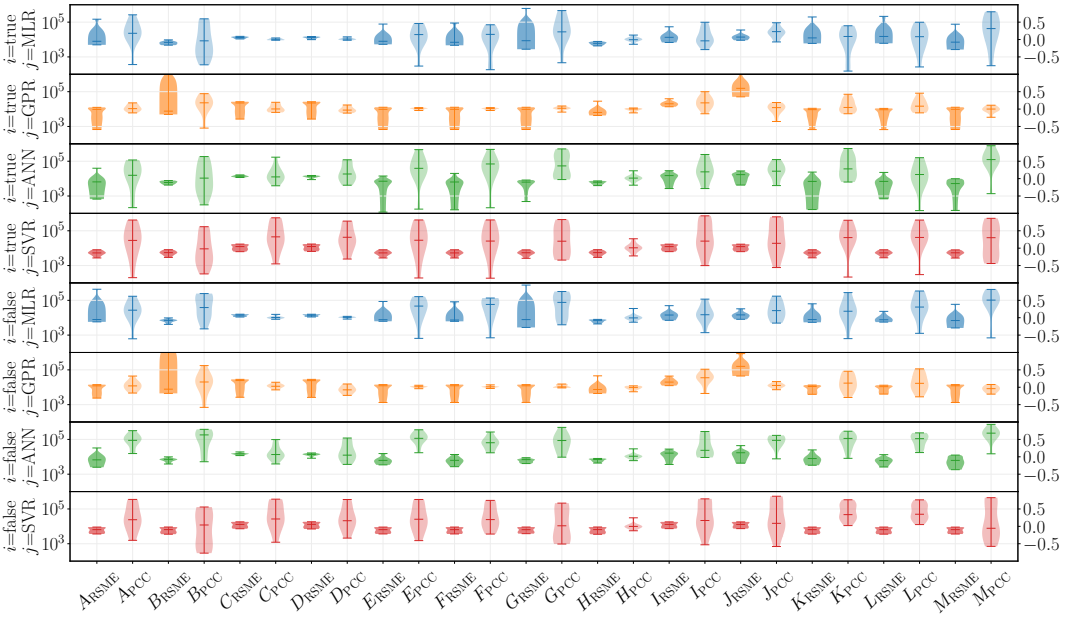


Figure 13.3: Violin plots for RMSE (dark) and PCC (light) for bearing data for each experiment. Note, bearing results are log-scaled for readability.

training dataset so that the RUL of an unseen sequence from the test dataset can be predicted before the results are then evaluated using root mean square error (RMSE) and Pearson correlation coefficient (PCC) (see Appendix B.2.9) between the observed and the estimated process of degradation. The experimental code and results can be found on GitHub[2]. Figure 13.3 shows violin plots for each experiment. Further results from this case study, particularly in relation to the remaining research questions on health stage classification (RQ1) and frequency analysis (RQ2), can be found in Appendix B.2.10.

To answer the research question **RQ3**, i.e., which feature set is most appropriate,

[2]https://github.com/inovex/RCIS2021-degradation-estimation-bearing-vessels

we compare RMSE and PCC of experiments $\{A, G\}_{i,j}$ vs. $\{B, H\}_{i,j}$ vs. $\{C, I\}_{i,j}$ vs. $\{D, J\}_{i,j}$ vs. $\{E, K\}_{i,j}$ vs. $\{F, L\}_{i,j}$ vs. $\mathrm{M}_{i,j}$ for $i = \{\text{true, false}\}$ and every j. The average RMSE per feature extraction method across all 8 experiments (with and without health stage classification and 4 regression methods) are 20.299, 93.429, 15.797, 50.555, 10.952, 8.449, and 15.986, respectively, suggesting that CNN as particularly effective or entropy feature particularly ineffective. However, when considering the effectiveness of the features in the context of different regression models, experiments $\{A, G\}_{i,j}$, $\{E, K\}_{i,j}$, $\{F, L\}_{i,j}$, and $\mathrm{M}_{i,j}$ perform worst with MLR, and $\{B, H\}_{i,j}$, $\{C, I\}_{i,j}$, and $\{D, J\}_{i,j}$ perform worst with GPR. Disregarding these two regression methods, average RMSEs are 6.962, 6.265, 13.603, 13.513, 7.082, 6.721, and 6.238. Learned features perform on average more than twice as bad as engineered features. Feature learning on engineered features, such as performing PCA or autoencoder on statistical features, is more efficient. In general, there is no free lunch, i.e., not every feature set is suitable for every regression model (Wolpert and Macready, 1997). Across all methods, CNN performs best, followed by entropy features, which only fail in the context of GPR. Comparing the model complexities of the two feature extraction methods, i.e., CNN and entropies, it is even more remarkable that the relatively simple entropy features perform so well.

To answer the research question **RQ4**, i.e., which regression model yields the highest accuracy, we compare the RMSEs of the experiments $\{A, \ldots, M\}_{i,\mathrm{MLR}}$ vs. $\{A, \ldots, M\}_{i,\mathrm{GPR}}$ vs. $\{A, \ldots, M\}_{i,\mathrm{ANN}}$ vs. $\{A, \ldots, M\}_{i,\mathrm{SVR}}$ for all i. Regarding all experiments, the average RMSE are $22.677, 87.091$, 9.578, and 8.041 for each different regression model $j = \{\mathrm{MLR}, \mathrm{GPR}, \mathrm{ANN}, \mathrm{SVR}\}$, respectively, with standard deviations of 19.166, 202.102, 3.663, and 3.042. GPR in particular turns out to be unsuitable at first glance, an affect that must be examined with regard to the outlier predictions. ANN and SVR prove to be particularly stable, which, together with the results of the third research question, indicates a good ability to generalise.

13.3 Interim Conclusion: Regression

All in all, in the experiments, entropies prove to be suitable representations for classical regression models. As introduced in Chapter 3, the concept of entropies is intuitive, simple and not particularly complex, especially when compared to other prominent feature representations, e.g., CNNs from the field of deep learning. It is all the more remarkable that entropy features are this effective compared to other features. Moreover, they work in combination with any classical regression model.

There are still numerous other methods for feature extraction and regression. We have limited ourselves here to the most popular ones. As the focus of this work was not to achieve the best possible performance, but to investigate the relation of different components, the application of regularisation, feature selection methods, a corresponding hyperparameter tuning, as well as the optimisation of network architectures are left for future work. As we only present a small part of a comprehensive case study in this work, some open challenges exist. Additional results for composing

data-driven prediction processes for degradation estimation based on the conducted experiments can be found in Appendix B.2.11 or in the original conference paper by Finke *et al.* (2021).

13.4 Classification as a Learning Task

To show that entropies are not only effective representations in the context of regression tasks, we focus on classification in the following. The aim of classification is to predict a certain class label and not, as in regression, a continuous value. In Section 2.3 we have already mentioned some models that can be used in supervised learning. An important criterion for the selection of supervised ML models for solving tasks is model complexity. Advanced and complex state-of-art models like ANNs can probably learn everything that a simple k-nearest neighbour (kNN) model can. However, this complexity also comes at a price: Developing or training an ANN, i.e., adjusting the training data, takes a lot of time. Furthermore, a more complex model needs much more training data to be fitted, and this data is not always available. Last but not least, more complex models are more difficult for us humans to interpret, whereas the interpretability of a model is often necessary and valuable. Therefore, in the following we focus on one of the least complex models, which is quickly trainable and easy to interpret: kNN. Conversely, kNN also has disadvantages, of course, in terms of its ability to adapt to highly complex relationships between independent and dependent variables. kNN is a non-parametric method, which means the model has a fixed number of parameters, and which do not grow with the amount of training data. Parametric models have the advantage that they are often faster to use out-of-the-box. However, stronger assumptions are made about the nature of the data distributions. In addition, the computational tasks are often computationally difficult with large data sets. For advanced tasks such as natural language processing or computer vision, kNN is less suitable. More details can be found in all classical machine learning textbooks, for example, by Murphy (2012).

kNN. The kNN algorithm is based on the assumption that similar things exist in close distance. Roughly speaking, kNN selects the $k \in \mathbb{N}_{>0}$ points in the training set X_{train} that are closest to the test input x_{test}. Euclidean distance is usually used to determine the distance, although other metrics can be used. The empirical proportion of members of each class y_i in this selected set is returned as an estimate. More formally,

$$p(y = c|x_{test}, X_{train}, k) = \frac{1}{k} \sum_{i \in N_k(x_{test}, X_{train})} [y_i = c] \tag{13.2}$$

where $N_k(x_{test}, X_{train})$ are the (indices of the) k nearest points to x_{test} in X_{train} and $[a]$ is the indicator function, i.e., $[a] = 1$ if a is true, and 0 otherwise. The class c with the highest proportion $p(y = c|x_{test}, X_{train}, k)$ is then assigned to the test input x_{test}.

13.5 MPE on the UEA MTSC Archive

In this section, we perform several experiments to investigate the relevance of the different ordinal pattern representations for multivariate time series. We challenge the introduced ordinal pattern representations in classification tasks on the UEA MTSC archive, a collection of different time series data sets from many real-world cases, released in October 2018 by Bagnall *et al.* (2018).

Experimental Details. We consider a representation as being *good*, if it is flexible and realiable applicable on different real-world data sets. The UEA MTSC archive consists of 30 data sets with a wide range of series lengths, dimensions and cases from human activity recognition, motion classification, ECG classification and EEG to audio spectra classification and many others. The properties of the data sets contained in the archive, such as training size n_{train}, test size n_{test}, number of spatial variables m, length T of the time series and number of classes C differ and can be found in App. A.3. Note that 2 out of 30 data sets of the archive could not be used due to the different length T_i of each spatial variable $i = 1, \dots, m$. We perform classifications on the UEA MTSC data sets to compare the ability of separation through eight ordinal pattern representations, namely PPE, MWPE, MOPPE, MPE-EUCL, MPE-MANH, MPE-NORM, MPE-PCA and MPE-PCA$_2$, where MPE-PCA$_2$ denotes the reduction of the dimension to two spatial variables before MOPPE is executed. Higher accuracy means that the representation is able to identify the underlying explanatory factors better than other representations and discriminatory properties can be identified as useful inputs for supervised predictors (Bengio *et al.*, 2013). The initial benchmarking on the UEA MTSC archive by Bagnall *et al.* (2018) is with the standard 1-NN classifier with three different distance functions, namely Euclidean, dimension-independent dynamic time warping and dimension-dependent dynamic time warping. To make a certain comparability we also use the 1-NN classifier as described in the previous section. As model input, all eight MPE-variants were used individually for the evaluation. Finding an optimal order d and time delay τ is a challenging problem in research (Riedl *et al.*, 2013; Myers and Khasawneh, 2020). For simplicity, we have done an extensive hyperparameter optimisation. Results can be found in Table B.2 in App. B.3.1. Before we start with a general evaluation of different data sets, we give a detailed insight in one specific data set out of the 30 for a better understanding.

Example 13.5.1 (Atrial Fibrillation Classification)**.** The *AtrialFibrillation* data set contains two-channel ECG recordings for predicting spontaneous termination of atrial fibrillation (AF). The class labels are: t, s and n. Class t contains data, where the AF terminates at the latest within one second after the recording ending. Class s is described as an AF that self-terminates at least one minute after the recording ending. In Class n, the AF does not terminate for at least one hour after the recording of the data. An example of the recordings for each class is shown in Figure 13.4. More details can be found in the original paper by Moody (2004).

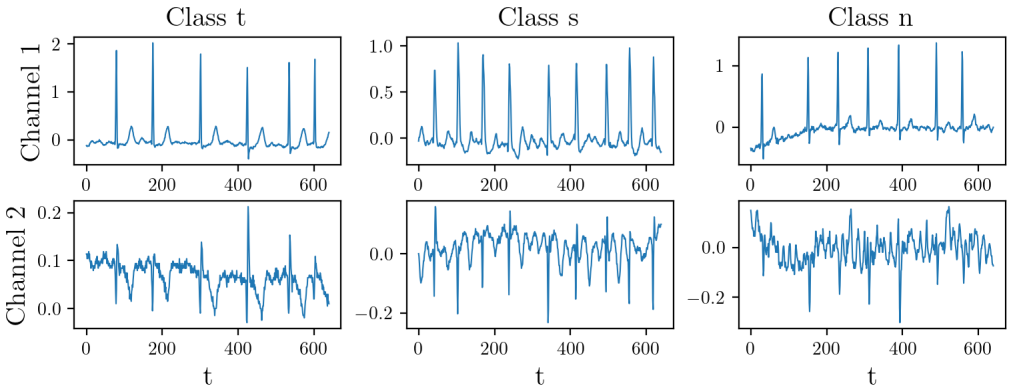


Figure 13.4: Example of 2-channel recordings for all three classes.

We examine the separability of the three classes based on the different MPE-variants, PPE, MWPE, MOPPE and MPE-PCA. The variance ratio for the first main component is greater than 74.17% for every $l = 1, .., n_{\text{train}}$, where n_{train} is the number of training samples. So a dimensionality reduction via PCA to a single dimension can be applied without losing too much information. Figure 13.5 shows boxplots for the calculated values of four different MPE-variants for each class. While PPE does not allow any separability of the classes, the weights of MWPE help to keep the classes n and t in a smaller range of MWPE values to make them identifiable. With MOPPE the class n can be separated relatively well by higher values from the class s by smaller values in comparison. An even better separation of classes n and s from t is achieved by MPE-PCA.

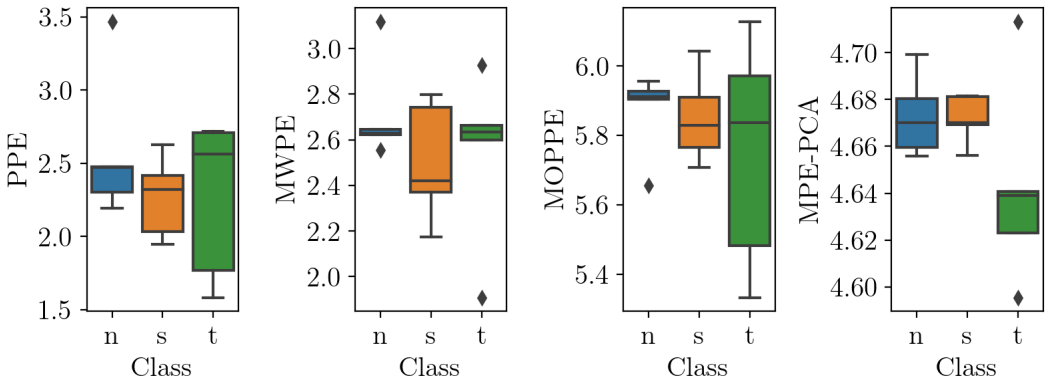


Figure 13.5: Boxplots for the values of four different MPE-variants of order $d = 5$ and delay $\tau = 1$ for classes n, s and t.

Experimental Results. Table 13.2 shows the mean accuracy for the given test data and labels for each UEA MTSC dataset when performing a 1-NN classification based on a single MPE-variant. Note that the results here may differ from those in the original paper (Mohr *et al.*, 2020b) as we have extended the hyperparameter search. The highest accuracy score per data set with respect to all eight MPE-variants is printed in bold. Besides, the names of data sets whose benchmark accuracy we outperform are marked in italics. Additional metrics, such as precision, recall and F_1-score can be found in Appendix B.3.2.

We start with a discussion of the results of the canonical variants PPE, MWPE, which is based on univariate ordinal patterns, and MOPPE, which is based on multivariate ordinal patterns. MWPE does not perform better than PPE for even half of the data sets. This is due to the fact that the amplitude information is not always relevant. Whether MWPE can be used profitably depends in particular on the application. MOPPE dominates in *Cricket* and *Eigenworms*, two of the longest records in the archive, confirming that the inclusion of the simultaneous motion of all spatial variables in the calculation is an important property for class separation. Nevertheless, the length T must be large and the number of spatial variables m small for this variant to be more successful than its averaged variant PPE, which discards the temporal commonality of spatial patterns in the computation. Next, we compare the results of the MPE-variants based on dimension reduction methods. MPE-PCA performs best on 10 data sets, while MPE-EUCL performs best only on 2 data sets and MPE-NORM only on 1 data set. MPE-MANH also achieves the highest accuracy only 1 times, along with MPE-NORM. All dimension-reducing procedures code certain properties of the spatial variables onto a projected single-dimensional variable on which the PE is subsequently calculated. The success of MPE-PCA shows that taking into account the correlations of the spatial variables is an important property to classify multivariate time series. In particular, a dimensionality reduction based on a decorrelation of the variables and then using the variable of greatest variance as a projection is more successful than a dimensionality reduction by classical distance measures such as Euclidean distance. The accuracy scores of MPE-PCA_2 confirm that adding more principal components does not necessarily give better results than MPE-PCA, so using the first principal component is sufficient in most cases.

	1-NN based on							
Data set	PPE	MWPE	MOPPE	EUCL	MANH	NORM	PCA	PCA_2
ArticularyWordRecog.	0.13	**0.14**	0.10	0.13	**0.14**	**0.14**	0.11	0.10
AtrialFibrillation	0.47	0.40	0.40	0.53	0.73	0.60	**0.80**	0.47
BasicMotions	0.55	**0.75**	0.53	0.58	0.38	0.43	0.55	0.58
Cricket	0.24	0.33	**0.36**	0.26	0.31	0.35	0.32	0.29
DuckDuckGeese	**0.36**	0.30	0.32	0.32	0.30	0.30	0.32	0.32

Eigenworms	0.50	0.47	**0.56**	0.47	0.47	0.53	0.49	0.46
Epilepsy	0.50	0.51	0.44	0.41	0.40	0.46	**0.54**	0.51
ERing	**0.49**	0.44	0.41	0.37	0.34	0.30	0.30	0.33
EthanolConcentration	0.29	**0.30**	0.29	0.29	0.29	0.28	**0.30**	0.28
FaceDetection	0.52	**0.53**	0.50	0.52	0.51	0.52	0.51	0.52
FingerMovements	0.54	0.56	0.49	0.54	0.58	0.58	**0.60**	0.57
HandMovementDir.	0.28	0.27	0.32	**0.36**	0.34	0.32	0.27	0.30
Handwriting	0.07	0.05	0.07	0.06	0.06	0.07	**0.08**	0.07
Heartbeat	0.67	0.65	0.65	0.63	0.63	0.62	**0.68**	0.62
JapaneseVowels	0.25	0.23	0.22	0.17	0.25	0.19	**0.26**	0.22
Libras	0.31	**0.37**	0.29	0.27	0.29	**0.37**	0.31	0.28
LSST	**0.20**	0.03	0.19	0.13	0.13	0.13	0.13	0.13
MotorImagery	0.61	**0.64**	0.60	0.55	0.61	0.63	0.57	0.59
NATOPS	0.19	**0.41**	0.22	0.23	0.25	0.24	0.23	0.22
PEMS-SF	**0.75**	0.61	0.56	0.61	0.60	0.64	0.63	0.63
PenDigits	0.22	0.17	0.18	**0.19**	0.18	0.18	**0.19**	0.15
PhonemeSpectra	**0.06**	0.06	0.06	0.06	0.06	0.06	0.06	0.05
RacketSports	0.38	0.33	0.32	0.32	0.32	0.30	**0.38**	0.32
SelfRegulationSCP1	0.59	0.56	0.57	0.61	0.61	0.58	**0.62**	0.59
SelfRegulationSCP2	**0.61**	0.55	0.56	0.55	0.55	0.57	0.53	0.54
SpokenArabicDigits	0.15	0.11	0.14	0.15	0.15	0.15	0.16	**0.17**
StandWalkJump	0.53	0.60	0.47	0.53	0.47	0.53	0.53	**0.67**
UWaveGestureLibrary	**0.24**	0.22	0.21	0.20	0.19	0.19	0.20	0.20

Table 13.2: Accuracy scores of all eight MPE-variants.

13.6 Interim Conclusion: Classification

In the second part of this chapter, we discussed ordinal pattern representations for multivariate time series in several classification tasks. We have shown that the newly introduced approaches, MWPE, MOPPE and MPE-PCA, outperform existing multivariate approaches on many real-world data sets. Which representation is chosen, however, depends in particular on the application. For example, MOPPE is a valuable representation in the case of a small number m of variables and a large length T of the time series. MWPE should only be used if the amplitude information is of relevance in the application. MPE-PCA takes into account the correlations of the spatial variables in the dimensional reduction and is thus more suitable than classical distance measures for reduction. Due to an improvement in prediction results and easy handling, the integration into toolboxes for representation learning of multivariate time series is indispensable.

Chapter 14

Conclusion

> Generally speaking, a *good* representation is one that makes a subsequent learning task easier.
>
> Goodfellow *et al.* (2016)

The success of a data-driven predictive model depends on the representation of the data used as input. Extracting representations from raw data – hand-crafted engineered or learned automatically – is an indispensable step when modelling time series, predicting the future or in other estimation or learning tasks. Representations must be informative and non-redundant in order to solve a learning task. Although the space of possible representations is infinitely large and at the same time infinitely exciting, this work focuses on one particular class, namely multivariate ordinal pattern representations. In this final chapter, we summarise the contributions of this work made in the context of multivariate ordinal pattern representations and give some directions for future work.

14.1 Summary of Contributions

In this work, we have focused on ordinal pattern representations in the context of multivariate time series. We have discussed limitations in existing approaches and proposed new approaches that overcome their main weakness: the incorporation of interdependencies or correlations of spatial variables of a multivariate time series. To show that the representations introduced are *good*, we have analysed them in different contexts. Overall, we summarise the contributions that are made in this work as follows.

(I) **New Approaches in Multivariate Ordinal Pattern Representations**

In Part I we reviewed previous research on multivariate ordinal pattern representations and uncovered their limitations. We have introduced three new multivariate ordinal pattern representations. Multivariate weighted permutation entropy (MWPE) fills the gaps by incorporating amplitude information. Multivariate ordinal pattern permutation entropy (MOPPE) and multivariate permutation entropy based on principle component analysis (MPE-PCA) overcome limitations of existing multivariate ordinal pattern representations by including dependencies or correlations between several spatial variables of the

multivariate time series. While MOPPE is based on a generalisation of the definition of univariate ordinal patterns to the multivariate case, MPE-PCA is based on a dimensional reduction of the spatial variables that allows the use of univariate patterns in its origin.

(II) **Investigations of Multivariate Ordinal Pattern Representations on Multivariate Fractional Brownian Motion**

In Part II we have studied the behaviour of both existing and newly introduced representations when applied to multivariate fractional Brownian motion. We have shown that there is a theoretical relationship between distributions of (multivariate) ordinal patterns and parameters of multivariate fractional Brownian motion such as the Hurst parameter, amplitude information or correlation, which can be expressed in formulae. Therefore, different multivariate ordinal pattern representations can be used to estimate different characteristics of multivariate fractional Brownian motion. Moreover, if it is known that an arbitrary time series is subject to the properties of mfBm, the statistical property of multivariate permutation entropy can be derived using simple formulas and does not have to be learned each time using complex (deep) learning methods.

(III) **Further Applications on Real-World Data**

In Part III we have proven the effectiveness of the introduced multivariate ordinal pattern representations in different learning tasks – unsupervised and supervised. In an unsupervised application, we used multivariate ordinal patterns for symmetry approximation in the context of dynamic probabilistic relational models. We have shown that MOP_4SA is a promising approach to avoid groundings and thus obtain runtime advantages without significantly reducing the accuracy in prediction. In a supervised application, we used multivariate ordinal pattern representations in regression as well as classification. Moreover, a representation is *good* if it is flexible and reliably applicable to different problems and real-world data sets. We have shown that permutation entropy representations overperform several other known classes of features and lead to better results in the context of different regression models for predicting the remaining useful lifetime life. Furthermore, we have shown that the newly introduced multivariate pattern representations outperform existing representations in several classification tasks on different real-world data sets of the UEA MTSC archive.

All in all, multivariate ordinal pattern representations are interesting because they are intrinsically motivated by interpretable upward and downward movements. In the context of this work, we have shown by example that by using multivariate ordinal pattern representations, specific subsequent learning tasks can be solved more easily. Nevertheless, the search for the *best possible* representation remains an exciting field of research.

14.2 Future Work

Starting from this work, there are various opportunities to investigate in the field of multivariate ordinal pattern representations and beyond. We highlight three topics.

Solving Inverse Problems. A main contribution of this work is the analysis of the theoretical relationship between the distribution of multivariate ordinal patterns and the Hurst parameter H of multivariate fractional Brownian motion. The direct relationship between these two can be used to solve an inverse problem, i.e., parameters such as self-similarity, cross-correlations or amplitude information are estimated from realised observations of mfBm and used to build a generalised model. How well multivariate ordinal pattern distributions are suited for this task must be evaluated accordingly within the framework of existing procedures.

Structures in Symmetries. In Chapter 12 we present MOP$_4$SA, an approach to approximate symmetries of multivariate time series in the context of dynamic probabilistic relational models. Rather, MOP$_4$SA is designed to be used in the context of arbitrary multivariate time series to cluster objects that behave the same or approximately the same. As is usual in machine learning, the success of MOP$_4$SA in other time series applications can only be proven by numerous evaluations on different real data sets. In addition, MOP$_4$SA opens up a new possibility to recognise structures such as periodicities in symmetry clusters. By looking at the similarity graph in MOP$_4$SA, suitable metrics and algorithms can uncover changes in or dependencies between the symmetry clusters. These metrics and algorithms need to be developed in future work.

Representation Learning. In classical machine learning, the extraction of representations or features, and the use of the model are considered as two separate steps that are performed one after the other. In this work, therefore, the extraction of multivariate ordinal representations and the modelling of the classification by k-nearest neighbours in Chapter 13 were also considered as two separate steps. Moreover, since this work investigates a special class of features, that of multivariate ordinal pattern representations, only one particular representation at a time was used for classification to compare them all. Usually, a high number of different features can increase the accuracy of the classification model. The set of optimal features can be determined by feature selection methods. In deep learning, on the other hand, internal representations are learned automatically within the model. Since intrinsic (multivariate) ordinal pattern representations have proven successful in recent years, the next step could be to learn them automatically. For example, one could try to learn the up and down movements as an internal representation of an autoencoder.

Part IV

Appendix

Appendix A

Datasets

For the evaluations conducted in this work, we use several real-world data sets, which we briefly present below.

A.1 FEMTO Bearing Dataset

In Section 2.3, we use a data set representing bearings of a machine. The data set is provided by FEMTO-ST institute within PRONOSTIA, an experimental platform dedicated to the testing and validation of bearing failure detection, diagnostic, and prognostic approaches (Nectoux *et al.*, 2012). The FEMTO bearing data set contains run-to-failure tests of 17 bearings each with time series data of vibration acceleration along the horizontal and vertical dimension as well as temperature. Temperature is not present in every run, thus, we exclude it in our experiments. The temperature and acceleration measurements, respectively, are sampled every ten seconds at 10Hz and 25.6kHz for 2560 time steps until failure. The bearings were operated under three different conditions (7 runs at 1800 rpm and 4000 N vs. 7 runs at 1650 rpm and 4200 N vs. 3 runs at 1500 rpm and 5000 N). The designation of samples b_{lm} is composed of the operating condition l and the number of runs m, where $l \in \{1, 2, 3\}$ and $m \in \{1, ..., 7\}$. The different operating conditions result in different test duration, ranging from 28min (b_{33}) to 7h 47min (b_{11}). The observed data are divided into a training and a test set with six and eleven bearings, respectively. Degradation itself corresponds directly to increasing vibrations. The fact that the bearings have different degradation patterns makes the building of a general model for RUL prediction more difficult. Further details can be found by Nectoux *et al.* (2012).

A.2 Automatic Identification System Data

In Chapter 12, we use historical vessel movements from 2020 based on automatic identification system (AIS) data[1] provided by the Danish Maritime Authority for the Baltic Sea to setup a DPRM as shown in Figure 12.1. AIS is an automatic tracking system that records data via a wireless system with the aim of improving the safety and guidance of vessel traffic by exchanging navigational and other vessel data. It was adopted as a mandatory standard by the International Maritime Organisation

[1] https://www.dma.dk/SikkerhedTilSoes/Sejladsinformation/AIS/

on 6 December 2000. Each AIS signal contains the current geo-position and the total cargo quantity of a vessel. As AIS data provides information about the position of a vessel, including specifications about the vessel itself, the actual supply and demand have to be calculated first. By dividing the Baltic Sea into zones, we derive the total amount of cargo transported between those zones based on the vessel movements and their cargo and thus obtain the data set for this evaluation. Preprocessing for retrieving variables *Supply* and *Idle* for 367 defined *Zones* can be found on GitHub[2].

A.3 UEA MTSC Archive

In Chapter 13, we challenge different MPE-variants in classification tasks on multivariate time series classification (MTSC) archive provided by University of East Anglia (UEA). Like the corresponding univariate archive, the archive was a collaborative effort between researchers at the UEA and the University of California, Riverside (UCR). The archive consists of different multivariate time series data sets from many real-world cases and was released in October 2018 by Bagnall *et al.* (2018). The 2018 vintage archive consists of 30 data sets with a wide range of series lengths, dimensions and cases from human activity recognition, motion classification, ECG, EEG and magnetoencephalography (MEG) classification to audio spectra classification and others. The data included should be formatted so that all time series in a data set are of equal length, no time series with missing data are included and there is a predefined split between training and testing. Table A.1 provides an overview of all data sets of the archive with regard to the number of training samples n_{train}, test samples n_{test}, variable dimensions m, time series length T and classes C.

Dataset	n_{train}	n_{test}	m	T	C
ArticularyWordRecognition	275	300	9	144	25
AtrialFibrillation	15	15	2	640	3
BasicMotions	40	40	6	100	4
CharacterTrajectories	1422	1436	3	182	20
Cricket	108	72	6	1197	12
DuckDuckGeese	60	40	1345	270	5
EigenWorms	128	131	6	17984	5
Epilepsy	137	138	3	206	4
EthanolConcentration	261	263	3	1751	4
ERing	30	30	4	65	6
FaceDetection	5890	3524	144	62	2
FingerMovements	316	100	28	50	2
HandMovementDirection	320	147	10	400	4
Handwriting	150	850	3	152	26

[2]https://github.com/FinkeNils/Processed-AIS-Data-Baltic-Sea-2020

Heartbeat	204	205	61	405	2
JapaneseVowels	270	370	12	29	9
Libras	180	180	2	45	15
LSST	2459	2466	6	36	14
InsectWingbeat	30000	20000	200	78	10
MotorImagery	278	100	64	3000	2
NATOPS	180	180	24	51	6
PenDigits	7494	3498	2	8	10
PEMS-SF	267	173	963	144	7
Phoneme	3315	3353	11	217	39
RacketSports	151	152	6	30	4
SelfRegulationSCP1	268	293	6	896	2
SelfRegulationSCP2	200	180	7	1152	2
SpokenArabicDigits	6599	2199	13	93	10
StandWalkJump	12	15	4	2500	3
UWaveGestureLibrary	120	320	3	315	8

Table A.1: A summary of the 30 datasets in the UEA Multivariate Time Series Classification Archive, 2018 (Bagnall *et al.*, 2018).

Appendix B

Experimental Details

In this work we conduct several experiments. In the following, we present corresponding details or additional results that only indirectly influence the results of the work, but are necessary to ensure understanding and reproducibility.

B.1 MPE-PCA Applied to mfBm

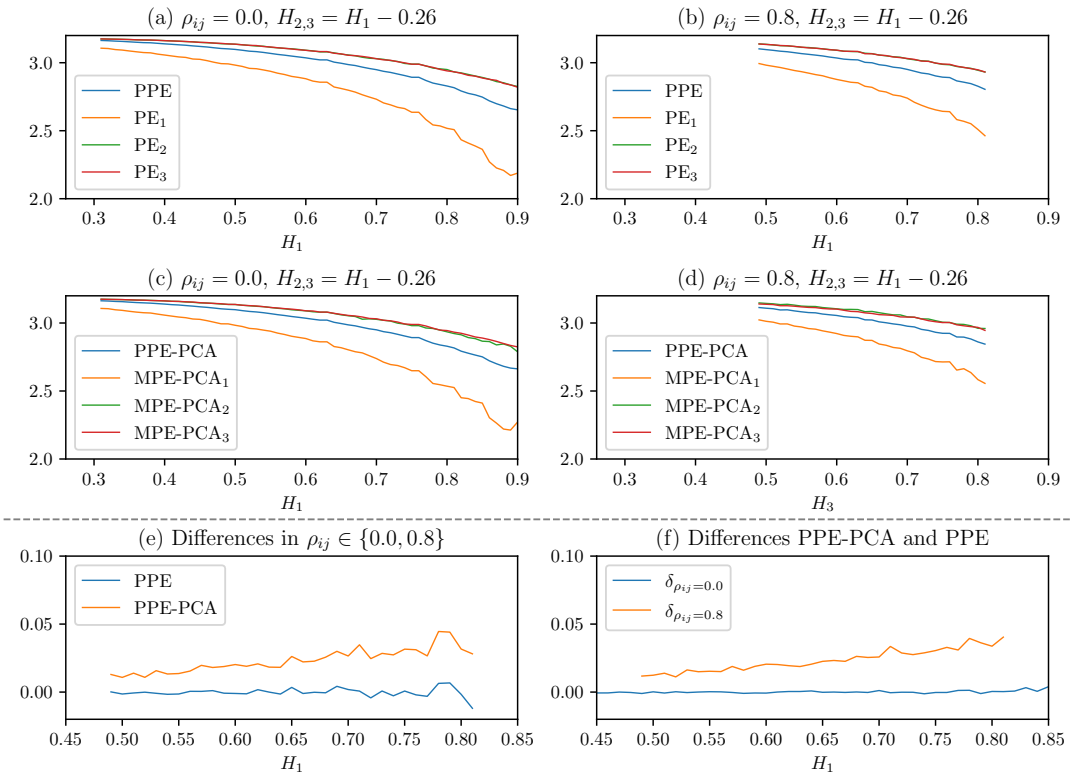


Figure B.1: Comparison of PPE and MPE-PCA of order $d = 4$ applied to mfBm.

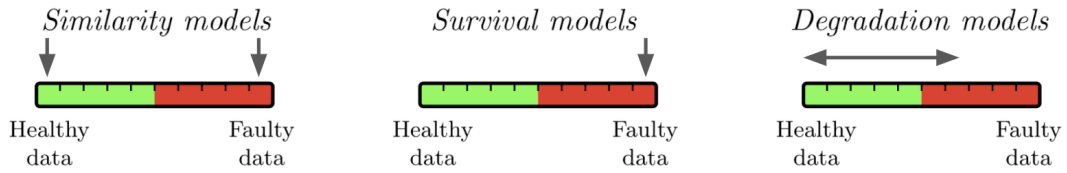


Figure B.2: Three families of models for the prediction of RUL.

B.2 PE in Degradation Estimation in Manufacturing

In Section 2.3, we present a small part of a broad case study on data-driven degradation estimation in manufacturing. In the following, we provide further details that are helpful for a comprehensive understanding.

B.2.1 Modelling the Remaining Useful Lifetime

Depending on the type of measurement data, three different model families are applied (Zheng, 2019):

- *Similarity models* use run-to-failure data from similar machines, starting during healthy operation and ending close to failure or maintenance. RUL is directly estimated from historical labeled training data by applying a pattern matching of trends or conditional indicator values.
- *Survival models* are used when the user does not have a complete history of run-to-failure data but instead has data about the life span of related components. Probability distributions are determined based on the behaviour of related components and used to estimate RUL.
- *Degradation models* estimate the degradation process without requiring faulty data. Historical behaviour of a machine condition indicator is used to extrapolate the damage progression to indirectly determine RUL.

The different families of data-driven models for predicting RUL are visualised in Figure B.2, with arrows indicating the types of training data available. As in real-world challenges complete run-to-failure data are rarely available, Section 2.3 focuses on degradation models.

B.2.2 Research Questions

(RQ1) Can health stage classification improve the accuracy of prediction?

(RQ2) Does the frequency spectrum of a time series provide more useful information than the raw data, i.e., time spectrum, itself?

(RQ3) Which feature sets are appropriate for the estimation of degradation?

(RQ4) Which data-driven regression method yields the highest accuracy?

B.2.3 Health Stage Classification

The second step of the overall process outlined in Fig. 13.1 is considered optional and is not explored further within the scope of this work. Nevertheless, we provide here an overview of what is meant by health stage classification. Caused by the fact that healthy data outweighs degradation data in regular operation, data-driven prediction of the degradation process can be impeded or even biased by healthy data. In order to distinguish between healthy and faulty stages in a time series, the point in time when the degradation starts has to be identified. The boundary of the two stages is called first prediction time. For simplicity, in this work we include an approach by Li *et al.* (2019), where kurtosis is used as classification indicator. The first prediction time corresponds to the time when the kurtosis of a sliding window over the time series exceeds the interval $\mu \pm 2\sigma$ for the second time, where μ is the mean and σ is the standard deviation at the beginning of the time series. After the identification of the first prediction time, observations in data classified as healthy are omitted from both training and prediction of degradation. The prediction by the regression model is initiated when data is classified as unhealthy.

B.2.4 Frequency Analysis

The second step of the overall process outlined in Fig. 13.1 is considered optional and is also not explored further within the scope of this work. Nevertheless, we provide here an overview of what is meant by frequency analysis. The analysis of the frequency range of a time series can provide further insights into the degradation process. In this step we distinguish between time spectrum, frequency spectrum and time-frequency spectrum analysis. By *time spectrum*, we denote the raw time series on which no frequency analysis is performed.

Fourier transform (FT) transforms an integrable function $f : \mathbb{R} \to \mathbb{C}$ from time to frequency domain. It is defined as

$$(\mathcal{F}f)(y) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i y \cdot t}\mathrm{d}t \tag{B.1}$$

for any real number t. The operator $\mathcal{F}$ decomposes the intensities of frequencies over time. Since in most real-world scenarios an integrable function $f(t)$ is not known, but only its values at discrete points in time, a discrete expansion is necessary. By *frequency spectrum*, we denote a time series that is transformed by discrete Fourier transform (DFT). DFT transforms a finite sequence of equally-spaced observed data points $(x_0, ..., x_T)$ into another sequence $(X_0, X_1, \ldots, X_T)$ that is a complex-valued function of frequency given by

$$X_k = \sum_{j=0}^{T} e^{-2\pi i \cdot \frac{jk}{T}} \cdot x_j \tag{B.2}$$

for $k = 0, ..., T$. The fast Fourier transform (FFT) is an efficient algorithm for computing DFT (Cooley and Tukey, 1965). Showing a trend, degradation time series

are inherently non-stationary, i.e., the mean is not constant over time. To analyse the frequency spectrum of non-stationary time series, short-time Fourier transform is used. To assume stationarity, the short-time Fourier transform uses a window function to select short time periods with constant mean. Several frequency spectra are calculated per window by DFT. By calculating several frequency spectra over the windows, in which the time series is assumed to be stationary, it is then possible to calculate a time-frequency spectrum via the FT. The short-time Fourier transform is defined as

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(t) w(t - \tau) e^{-i\omega t} \mathrm{d}t, \tag{B.3}$$

where $x(t)$ is the time series that the time-frequency distribution should be computed for, $w(\tau)$ is the sliding window function, and τ the delay in between windows. In most cases, the windows are selected with some overlap. By *time-frequency spectrum*, we denote a time series on which short-time Fourier transform is performed. Note that in the next step of the overall process, not every feature extraction method can be applied on every frequency analysis method.

B.2.5 Statistical Features

Mean	$\bar{x} = \frac{1}{T}\sum_{i=1}^{T} x_i$	Skewness	$\frac{\frac{1}{T}\sum_{i=1}^{T}(x_i-\bar{x})^3}{(\frac{1}{T}\sum_{i=1}^{T}(x_i-\bar{x})^2)^{\frac{3}{2}}}$
Max	$\max\{x_1, ..., x_T\}$	Kurtosis	$\frac{\frac{1}{T}\sum_{i=1}^{T}(x_i-\bar{x})^4}{(\frac{1}{T}\sum_{i=1}^{T}(x_i-\bar{x})^2)^2}$
Min	$\min\{x_1, ..., x_T\}$	Peak factor	$\frac{max(x)}{\sqrt{\frac{1}{T}\sum_{i=1}^{T} x_i^2}}$
Root mean square	$\sqrt{\frac{1}{n}\sum_{i=1}^{T} x_i^2}$	Change coefficient	$\frac{\bar{x}}{\sqrt{\frac{1}{T}\sum_{i=1}^{T} x_i^2}}$
Peak to peak value	$\max(x) - \min(x)$	Clearence factor	$\frac{max(x)}{\frac{1}{T}\sum_{i=1}^{T}(x_i)^2}$
Variance	$\frac{1}{T}\sum_{i=1}^{T}(x_i - \bar{x})^2$	Absolute Energy	$\sum_{i=1}^{T} x_i^2$

Table B.1: List of statistical features.

B.2.6 Autoencoder

A relatively new method for reducing dimensionality are *autoencoders*, a branch of ANNs. The architecture consists of two connected ANNs compressing the input variable into a reduced dimensional space, also called encoder, and re-creating the input data, also called decoder. Each node of the hidden "bottleneck" layer of compressed information can be treated as a feature in subsequent learning tasks. The autoencoder can be applied directly to the time and frequency spectrum.

B.2.7 Convolutional Neural Network

A CNN is another type of ANNs typically used for image recognition, but also for signal processing. The architecture of a classical CNN consists of one or more convolutional layers followed by a pooling layer. In a convolutional layer, a matrix, also called filter kernel, is moved stepwise over the input data calculating the inner product of both. The result is called feature map. Accordingly, neighbouring neurons in the convolutional layer correspond to overlapping regions such as similar frequencies in signals. In a pooling layer, superfluous information is discarded and a more abstract lower-dimensional representation of the relevant information is obtained by combining neighbouring elements of the map, e.g., by calculating the maximum. To feed the matrix output of the convolution layer and the pooling layer into a final fully connected layer, it must first be unrolled (flattened). The flatten layer is then treated as a feature. The CNN has to be applied to the time-frequency spectrum.

B.2.8 Implementation Details for the Bearing Experiment

For the purpose of reproducibility, we list the implementation details as follows. Outliers are removed based on Z-Score before data is normalised with Min-Max-Scaler by scikit-learn. FFT and short-time Fourier transform are implemented with SciPy with sampling rate 25.6kHz. For short-time Fourier transform, the Hann window function is used with a window length of 256 and an overlap of 128, meaning when given a time series with 2560 data points, the time series is separated into 21 different overlapping windows, and the Fourier transform is calculated for each of them resulting in the time-frequency spectrum of the time series. Statistical and entropy features are provided by tsfresh. For the calculation of Shannon entropy we use the classical symbolisation of the time series by SAX from pyts. For the calculation of permutation entropy we use the ordinal symbolisation by tsfresh with delay $\tau = 10$ and order $d = 5$. PCA is implemented using scikit-learn with encoding size 25. The autoencoder, CNN and ANN are implemented using Keras. The autoencoder architecture for feature learning consists of two encoding layers of size 160 and 80, followed by the coding layer of size 25 and two decoding layers of size 80 and 160. The CNN architecture for feature learning consists of 2 convolutional layers of dimension 6×6, each followed by a pooling layer of dimension 2×2 and a batch normalisation before the flattening layer is used for feature representation. The ANN architecture for the regression task consists of two hidden layers and an output layer, each of them with 512 hidden units. The activation function is chosen as rectified linear unit, i.e., $\mathrm{ReLu}(x) = \max(0, x)$. To avoid overfitting, the dropout rate is set to 0.5. The autoencoder, CNN, and ANN are trained using Adam optimizer with learning rate 0.001 and loss function as mean squared error. MLR, GPR and SVR are implemented by scikit-learn with default settings. For health stage classification only one of the available variables is used, namely horizontal vibration for bearing and log speed for the vessel dataset. Observations in each dataset are recorded until end of useful lifetime.

B.2.9 Metrics

RMSE is defined by

$$\mathrm{RMSE}(x,y) = \Big(\frac{1}{n}\sum_{i=1}^{T}(y_i - x_i)^2\Big)^{1/2}, \tag{B.4}$$

where $x = (x_1, ..., x_T)$ and $y = (y_1, ..., y_T)$ are time series and T is the length of both time series. Note that both time series are assumed to be of equal length. PCC measures the linear correlation of two time series x and y, and is defined by

$$\mathrm{PCC}(x,y) = \frac{\sum_{i=1}^{T}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{T}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{T}(y_i - \bar{y})^2}}, \tag{B.5}$$

where $\bar{x}$, $\bar{y}$ and s_x, s_y are the mean and the sample standard deviation of each respective time series. PCC has a value from -1 to 1, where 1 means that x and y are positively correlated, 0 means that there is no correlation at all, and -1 means a negative correlation. PCC describes the similarity of the behaviour of two time series, i.e., PCC indicates whether a learned model is able to correctly identify the degradation pattern (in case, PCC is close to 1). Note, PCC should be considered together with RMSE.

B.2.10 Further Experimental Results

To answer the first research question **(RQ1)** from App. B.2.2, whether a health stage classification can improve the accuracy of the prediction, we compare RMSE and PCC of experiments $\{A, \ldots, M\}_{\mathrm{true},j}$ vs. $\{A, \ldots, M\}_{\mathrm{false},j}$ for every regression model $j = \{\mathrm{MLR}, \mathrm{GPR}, \mathrm{ANN}, \mathrm{SVR}\}$. Experiments show that RMSE decreases if the HS classifier is applied in 59%, 78%, 64%, and 74% of the predictions, respectively. Thus, in general, an improvement is observed. This does not imply that the total RMSE over all bearings or vessels must also decrease. Indeed, for bearing data it even increases for $\{K, L, M\}_{\mathrm{true,MLR}}$ and $\{B, J\}_{\mathrm{true,GPR}}$, which can be taken from Figure 13.3 (top, blue and orange). Compared to RMSE, PCC increases in 41%, 63%, 34%, and 69% of the predictions, which indicates an overall improvement. Nevertheless, there is a deterioration of the average PCC in both case studies when using MLR. Hence, it should be checked individually whether there is an improvement in the functional relationship.

To answer the second research question research question **(RQ2)** from App. B.2.2, whether frequency analysis can provide additional information, we compare RMSE and PCC of experiments $A_{i,j}$ vs. $G_{i,j}$, $B_{i,j}$ vs. $H_{i,j}$, $C_{i,j}$ vs. $I_{i,j}$, $D_{i,j}$ vs. $J_{i,j}$, $E_{i,j}$ vs. $K_{i,j}$ and $F_{i,j}$ vs. $L_{i,j}$ for all i, j. The average RMSE and average PCC shows that only in the case of $B_{i,j}$ vs. $H_{i,j}$ an improvement is achieved, i.e., a reduction of the RMSE and an increase of the PCC. More specifically, we find that

the feature calculation on the frequency spectra leads to a reduction of RMSE only in $42\%, 50\%, 47\%, 38\%, 52\%$ and 55% of the predictions, which is close to random guessing. It is further to point out that $D_{\text{true,GPR}}$ and $E_{\text{true,MLR}}$ lead to an increase in RMSE in 100% of the predictions for bearing data, while in the case of vessel data they decrease in 100% of predictions. Therefore, we do not recommend blind use of frequency analysis, but rather use it wisely. Note, that we did not investigate whether combining features on the raw time spectrum in combination with features on the frequency spectrum gives better results. We leave this for future work in the context of feature selection.

B.2.11 Recommendations

We provide recommendations for composing data-driven prediction processes for degradation estimation based on the conducted experiments. Limits in the application depend on the individual use case that is to be implemented. Note that finding suitable degrading data directly related to the RUL of a machine part or complex system is not trivial. It requires initial analyses of the data and its correlations. The functional relationship have to be investigated or, if necessary, transformed by appropriate preprocessing such as creation of indicators. Along the research questions we recommend as follows.

1. *Health stage classifier*: We advise integrating a health stage classifier within the degradation estimation process, as in the vast majority of cases both RMSE and PCC are improved. Note that there are other health stage classifiers that may be more appropriate for your individual problem.

2. *Frequency analysis*: We do not recommend predicting the degradation solely by features calculated on frequency spectra. This does not mean that such features cannot add value in combinations with others.

3. *Feature set*: While CNN and entropy features are most suited for bearing data, classical statistical features are for vessel data. For getting started, we recommend using feature engineering before putting a lot of effort into feature learning and tuning its hyperparameters. The feature extraction method can be easily replaced in the process later. A good prediction depends on both, the choice of features, as well as the choice of a model.

4. *Regression model*: GPR may be used with caution and only be applied to appropriate data. Furthermore, we recommend more complex models than MLR. Not surprisingly, ANN and SVR perform best, with ANN being able to better represent the functional relationship. SVR is known for good generalisation ability, which is also shown here. Our comparison results confirm those of other authors, e.g., by Kim *et al.* (2012); Ozkat (2019).

B.3 MPE on the UEA MTSC Archive

B.3.1 Hyperparameter-Tuning

For each UEA MTSC dataset and each MPE-variant, we determined the best hyperparameters from a set of orders $d \in \{2, \ldots, 6\}$ and delays $\tau \in \{1, 2, 3, 5, 10\}$ in terms of classification. Results are shown in the following Table B.2.

Data set		1-NN based on PPE	MWPE	MOPPE	EUCL	MANH	NORM	PCA	PCA_2
ArticularyWordRecognition	d	3	5	2	4	5	4	6	4
	τ	5	5	1	1	1	1	1	10
AtrialFibrillation	d	2	2	4	6	3	6	2	2
	τ	10	5	1	5	1	10	1	5
BasicMotions	d	3	5	2	6	4	5	5	3
	τ	5	1	5	1	1	1	5	5
Cricket	d	5	5	3	3	5	4	4	5
	τ	5	10	5	10	10	1	1	1
DuckDuckGeese	d	2	4	2	5	2	3	3	4
	τ	1	1	1	10	10	10	1	1
Eigenworms	d	3	3	2	3	3	2	5	3
	τ	10	1	5	10	1	5	10	10
Epilepsy	d	6	4	4	4	6	4	5	5
	τ	1	1	10	5	5	1	5	10
ERing	d	2	3	3	2	2	6	6	4
	τ	1	3	3	3	3	3	1	1
EthanolConcentration	d	3	4	3	3	5	4	3	2
	τ	10	5	5	5	5	10	10	5
FaceDetection	d	4	4	2	2	3	5	4	4
	τ	1	3	3	2	3	3	1	2
FingerMovements	d	5	4	2	4	4	5	5	2
	τ	1	3	1	2	3	2	2	1
HandMovementDirection	d	4	4	3	3	4	4	3	4
	τ	5	1	1	5	1	5	5	1
Handwriting	d	5	5	4	5	3	5	5	3
	τ	10	10	10	10	1	10	1	1

Heartbeat	d	2	3	5	3	5	3	2	2
	τ	1	5	5	10	1	10	10	10
JapaneseVowels	d	3	3	2	4	4	2	2	2
	τ	2	2	3	1	1	1	1	1
LSST	d	3	4	3	4	3	3	4	3
	τ	2	1	1	1	2	2	1	1
Libras	d	5	4	4	6	5	5	5	3
	τ	3	2	1	1	3	2	3	3
MotorImagery	d	5	2	2	2	2	4	2	3
	τ	10	1	1	5	5	1	10	10
NATOPS	d	3	5	2	4	4	5	2	5
	τ	1	2	2	1	3	2	3	2
PEMS-SF	d	3	3	2	5	5	5	5	4
	τ	1	5	10	1	10	10	1	5
PenDigits	d	3	3	2	3	2	3	3	2
	τ	2	2	1	1	2	1	1	2
PhonemeSpectra	d	3	4	3	4	4	3	3	5
	τ	1	1	1	1	1	1	1	1
RacketSports	d	4	4	2	3	4	5	3	3
	τ	2	2	2	2	2	1	2	2
SelfRegulationSCP1	d	4	3	5	5	5	4	4	5
	τ	1	5	1	5	5	5	1	1
SelfRegulationSCP2	d	2	4	2	2	4	2	5	2
	τ	10	10	5	5	1	1	10	5
SpokenArabicDigits	d	2	4	2	3	3	3	3	2
	τ	2	1	3	2	2	2	2	2
StandWalkJump	d	4	3	2	3	2	4	3	3
	τ	1	10	1	1	5	10	5	1
UWaveGestureLibrary	d	2	3	3	2	3	6	2	4
	τ	10	1	1	5	1	10	1	1

Table B.2: Selected hyperparameters for classification.

B.3.2 Precision, Recall and F_1-Score

Precision and recall are then defined by

$$p = \frac{tp}{tp+fp}, \tag{B.6}$$

$$r = \frac{tp}{tp+fn}, \tag{B.7}$$

where tp denotes true positives, tn denotes true negatives, fp denotes false positives, and fn denotes false negatives. The traditional F_1-measure or balanced F_1-score is a measure that combines precision and recall, i.e., the harmonic mean of precision and recall. Specifically,

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{B.8}$$

In the experiments we use the weighted F_1-score, i.e., for each label the F_1-score is calculated and then their average is weighted with the accuracy. This can result in an F_1-score that is not between precision and recall. Table B.3 shows precision, recall and weighted F_1-score of the experiments performed for each data set of the UEA MTSC archive.

		1-NN based on							
Data set		PPE	MWPE	MOPPE	EUCL	MANH	NORM	PCA	PCA_2
ArticularyWordR.	r	0.13	0.14	0.09	0.15	0.13	0.14	0.11	0.10
	p	0.13	0.14	0.10	0.13	0.14	0.14	0.11	0.10
	F_1	0.13	0.13	0.10	0.13	0.13	0.13	0.10	0.10
AtrialFibrillation	r	0.51	0.44	0.57	0.57	0.74	0.61	0.61	0.45
	p	0.47	0.40	0.40	0.53	0.73	0.60	0.60	0.47
	F_1	0.47	0.39	0.39	0.53	0.73	0.59	0.59	0.45
BasicMotions	r	0.56	0.75	0.53	0.59	0.32	0.44	0.55	0.56
	p	0.55	0.75	0.53	0.58	0.38	0.43	0.55	0.58
	F_1	0.54	0.75	0.51	0.58	0.34	0.41	0.54	0.55
Cricket	r	0.31	0.33	0.37	0.23	0.27	0.37	0.28	0.27
	p	0.24	0.33	0.36	0.26	0.31	0.35	0.32	0.29
	F_1	0.24	0.32	0.36	0.24	0.28	0.34	0.30	0.27
DuckDuckGeese	r	0.39	0.30	0.31	0.18	0.28	0.42	0.35	0.31
	p	0.36	0.3	0.32	0.30	0.30	0.32	0.32	0.32
	F_1	0.35	0.29	0.31	0.21	0.28	0.33	0.32	0.30
Eigenworms	r	0.51	0.49	0.54	0.46	0.47	0.52	0.49	0.46
	p	0.50	0.47	0.56	0.47	0.47	0.53	0.49	0.46
	F_1	0.53	0.47	0.55	0.46	0.46	0.52	0.49	0.45

Epilepsy	r	0.49	0.51	0.44	0.40	0.40	0.45	0.55	0.51
	p	0.50	0.51	0.44	0.41	0.40	0.46	0.54	0.51
	F_1	0.49	0.51	0.44	0.40	0.39	0.45	0.54	0.51
ERing	r	0.49	0.45	0.40	0.33	0.32	0.30	0.27	0.29
	p	0.49	0.44	0.41	0.37	0.34	0.30	0.30	0.33
	F_1	0.47	0.45	0.40	0.34	0.31	0.30	0.27	0.29
EthanolConc.	r	0.29	0.29	0.29	0.30	0.29	0.28	0.30	0.28
	p	0.29	0.30	0.29	0.29	0.30	0.28	0.30	0.28
	F_1	0.29	0.29	0.29	0.29	0.29	0.28	0.30	0.28
FaceDetection	r	0.52	0.54	0.54	0.53	0.51	0.52	0.51	0.52
	p	0.51	0.52	0.50	0.52	0.51	0.52	0.51	0.52
	F_1	0.52	0.52	0.34	0.47	0.51	0.52	0.51	0.51
FingerMovements	r	0.54	0.57	0.48	0.54	0.58	0.59	0.60	0.57
	p	0.54	0.56	0.49	0.54	0.58	0.58	0.60	0.57
	F_1	0.54	0.55	0.37	0.54	0.58	0.57	0.60	0.57
HandMovementD.	r	0.31	0.28	0.36	0.42	0.37	0.40	0.32	0.32
	p	0.28	0.27	0.32	0.36	0.34	0.32	0.27	0.30
	F_1	0.29	0.27	0.33	0.37	0.34	0.34	0.27	0.30
Handwriting	r	0.07	0.05	0.06	0.07	0.06	0.07	0.06	0.07
	p	0.07	0.05	0.07	0.06	0.06	0.07	0.07	0.07
	F_1	0.06	0.05	0.06	0.06	0.06	0.07	0.06	0.06
Heartbeat	r	0.67	0.66	0.61	0.64	0.64	0.60	0.69	0.63
	p	0.68	0.65	0.65	0.63	0.63	0.62	0.68	0.62
	F_1	0.67	0.65	0.63	0.63	0.64	0.61	0.68	0.63
JapaneseVowels	r	0.30	0.25	0.28	0.19	0.24	0.21	0.27	0.24
	p	0.26	0.23	0.22	0.17	0.25	0.19	0.25	0.22
	F_1	0.27	0.23	0.24	0.14	0.22	0.19	0.25	0.22
Libras	r	0.32	0.37	0.29	0.24	0.34	0.38	0.29	0.27
	p	0.31	0.37	0.29	0.27	0.29	0.37	0.31	0.28
	F_1	0.31	0.36	0.29	0.25	0.29	0.36	0.29	0.27
LSST	r	0.20	0.18	0.10	0.17	0.17	0.18	0.17	0.17
	p	0.20	0.19	0.03	0.13	0.13	0.13	0.13	0.13
	F_1	0.20	0.18	0.03	0.14	0.13	0.14	0.14	0.13
MotorImagery	r	0.61	0.64	0.60	0.55	0.61	0.64	0.57	0.59
	p	0.61	0.64	0.60	0.55	0.61	0.63	0.57	0.59
	F_1	0.61	0.64	0.60	0.55	0.61	0.63	0.57	0.59
NATOPS	r	0.19	0.42	0.17	0.23	0.24	0.24	0.24	0.23
	p	0.19	0.41	0.22	0.23	0.25	0.24	0.23	0.22

	F_1	0.19	0.40	0.19	0.23	0.24	0.24	0.20	0.22
PEMS-SF	r	0.75	064	0.57	0.62	0.63	0.64	0.66	0.65
	p	0.75	0.61	0.56	0.61	0.60	0.64	0.63	0.63
	F_1	0.74	0.61	0.56	0.61	0.59	0.63	0.63	0.63
PenDigits	r	0.31	0.17	0.18	0.18	0.09	0.23	0.16	0.06
	p	0.22	0.17	0.19	0.20	0.18	0.18	0.19	0.15
	F_1	0.18	0.17	0.15	0.18	0.11	0.15	0.16	0.09
PhonemeSpectra	r	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07
	p	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.05
	F_1	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.04
RacketSports	r	0.38	0.33	0.31	0.32	0.33	0.31	0.34	0.33
	p	0.38	0.33	0.32	0.32	0.32	0.30	0.34	0.32
	F_1	0.37	0.32	0.30	0.31	0.32	0.30	0.33	0.32
SelfReg.SCP1	r	0.59	0.56	0.59	0.62	0.61	0.58	0.62	0.59
	p	0.59	0.56	0.57	0.62	0.61	0.58	0.61	0.59
	F_1	0.59	0.56	0.55	0.62	0.61	0.58	0.61	0.59
SelfReg.SCP2	r	0.61	0.55	0.56	0.55	0.55	0.57	0.53	0.54
	p	0.61	0.55	0.56	0.55	0.55	0.57	0.53	0.54
	F_1	0.61	0.55	0.55	0.55	0.55	0.56	0.53	0.54
SpokenArabicD.	r	0.18	0.11	0.15	0.15	0.15	0.15	0.15	0.17
	p	0.15	0.11	0.14	0.15	0.15	0.15	0.16	0.17
	F_1	0.13	0.11	0.10	0.15	0.15	0.15	0.15	0.16
StandWalkJump	r	0.57	0.59	0.51	0.72	0.48	0.67	0.53	0.68
	p	0.53	0.60	0.47	0.53	0.47	0.67	0.53	0.67
	F_1	0.53	0.59	0.47	0.51	0.45	0.66	0.53	0.65
UWaveGestureL.	r	0.24	0.22	0.20	0.18	0.19	0.19	0.20	0.21
	p	0.24	0.22	0.21	0.20	0.19	0.19	0.20	0.20
	F_1	0.24	0.22	0.20	0.19	0.19	0.19	0.19	0.20

Table B.3: Precision, recall and weighted F_1-scores for classifications.

Bibliography

Sophie Achard, Danielle S. Bassett, Andreas Meyer-Lindenberg, and Ed Bullmore. Fractal Connectivity of Long-Memory Networks. *Physical Review E*, 77(3), 2008.

Alejandro Agostini and Enric Celaya. Exploiting Domain Symmetries in Reinforcement Learning with Continuous State and Action Spaces. In *2009 International Conference on Machine Learning and Applications*, pages 331–336. IEEE, 2009.

Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient Similarity Search in Sequence Databases. In David B. Lomet (Ed.), *Foundations of Data Organization and Algorithms*, pages 69–84. Springer, 1993.

Wasim Ahmad, Sheraz Ali Khan, M M Manjurul Islam, and Jong-Myon Kim. A Reliable Technique for Remaining Useful Life Estimation of Rolling Element Bearings using Dynamic Regression Models. *Reliability Engineering and System Safety*, 184(C):67–76, 2019.

Alexander Alvarez. Financial Modelling with Multivariate Mixed Fractional Brownian Motion. *Investigación Operacional*, 42(2):127–137, 2021.

Pierre-Olivier Amblard and Jean-François Coeurjolly. Identification of the Multivariate Fractional Brownian Motion. *IEEE Transactions on Signal Processing*, 59(11):5152–5168, 2011.

Pierre-Olivier Amblard, Jean-François Coeurjolly, Frédéric Lavancier, and Anne Philippe. Basic properties of the Multivariate Fractional Brownian Motion. *Séminaires et congrès*, 28:65–87, 2013.

José Amigó. *Permutation Complexity in Dynamical Systems: Ordinal Patterns, Permutation Entropy and All That*. Springer Series in Synergetics. Springer, 2010.

Adriana Antonelli, Gustavo Meschino, and Virginia Ballarin. Mammographic Density Estimation Through Permutation Entropy. In Lenka Lhotska, Lucie Sukupova, Igor Lacković, and Geoffrey S. Ibbott (Eds.), *World Congress on Medical Physics and Biomedical Engineering 2018*, IFMBE Proceedings, pages 135–141. Springer, 2019.

Alexandra Antoniouk, Karsten Keller, and Sergiy Maksymenko. Kolmogorov-Sinai Entropy via Separation Properties of Order-Generated σ-Algebras. *Discrete & Continuous Dynamical Systems*, 34(5):1793–1809, 2014.

Sergio Arianos and Anna Carbone. Cross-Correlation of Long-Range Correlated Series. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(03):P03037, 2009.

Antoine Ayache and Jacques Lévy Véhel. Generalized Multifractional Brownian Motion: Definition and Preliminary Results. In Michel Dekking, Jacques Lévy Véhel, Evelyne Lutton, and Claude Tricot (Eds.), *Fractals*. Springer, 1999.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA Multivariate Time Series Classification Archive, 2018. *arXiv:1811.00075*, 2018.

Lucas Baier, Niklas Kühl, Gerhard Satzger, Marcel Hofmann, and Marisa Mohr. Handling Concept Drifts in Regression Problems - the Error Intersection Approach. In Norbert Gronau, Moreen Heine, Hanna Krasnova, and K. Poustcchi (Eds.), *Entwicklungen, Chancen und Herausforderungen der Digitalisierung: Proceedings der 15. Internationalen Tagung Wirtschaftsinformatik, WI 2020*, pages 210–224. GITO Verlag, 2020.

Christoph Bandt and Bernd Pompe. Permutation Entropy: A Natural Complexity Measure for Time Series. *Physical Review Letters*, 88(17):174102, 2002.

Christoph Bandt and Faten Shiha. Order Patterns in Time Series. *Journal of Time Series Analysis*, 28(5):646–665, 2007.

Christian Becker and Marisa Mohr. Federated Machine Learning: Über Unternehmensgrenzen hinaus aus Produktionsdaten lernen. *atp magazin 05/2020*, 62(5):28–30, 2020.

Richard Ernest Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton Legacy Library. Princeton University Press, 2015.

Albert Benassi, Daniel Roux, and Stéphane Jaffard. Elliptic Gaussian Random Processes. *Revista Matematica Iberoamericana*, 13:19–90, 1997.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Jan Beran, Yuanhua Feng, Sucharita Ghosh, and Rafal Kulik. *Long-Memory Processes: Probabilistic Properties and Statistical Methods*. Springer, 2013.

Andrea L. Bertozzi and Ekaterina Merkurjev. Graph-based Optimization Approaches for Machine Learning, Uncertainty Quantification and Networks. In Ron Kimmel and Xue-Cheng Tai (Eds.), *Processing, Analyzing and Learning of Images, Shapes, and Forms: Part 2*, volume 20 of *Handbook of Numerical Analysis*, pages 503–531. Elsevier, 2019.

Pavle Boskoski, Matej Gasperin, Dejan Petelin, and Dani Juricic. Bearing Fault Prognostics using Rényi Entropy Based Features and Gaussian Process Models. *Mechanical Systems and Signal Processing*, 52-53, 2015.

Tanya Braun and Ralf Möller. Lifted Junction Tree Algorithm. In *KI 2016: Advances in Artificial Intelligence*, pages 30–42. Springer, 2016.

Tobias Bux and Marisa Mohr. Blockchain-Lösungen für den produktionstechnischen Mittelstand/Blockchain Solutions for Medium-sized Production Engineering Companies – Proof of Confidence for Cross-Company Networking of Production and Process Data. *wt Werkstattstechnik online*, 110(04):201–204, 2020.

Tobias Bux, Jonas Groß, Constantin Lichti, and Marisa Mohr. Projekt KOSMoS: Mit Blockchain transparent und firmenübergreifend warten. *atp magazin 03/2021*, 63(3):28–30, 2021.

Bill Chiu, Eamonn Keogh, and Stefano Lonardi. Probabilistic Discovery of Time Series Motifs. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 493–498. ACM, 2003.

François Chollet. *Deep Learning with Python*. Manning Publications Company, 2017.

Jean-François Coeurjolly, Pierre-Olivier Amblard, and Sophie Achard. Normalized Causal and Well-balanced Multivariate Fractional Brownian Motion. https://hal.archives-ouvertes.fr/hal-00501720, 2010.

James W. Cooley and John W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, 19:297–301, 1965.

Madalena Costa, Ary L. Goldberger, and C.-K. Peng. Multiscale Entropy Analysis of Complex Physiologic Time Series. *Physical Review Letters*, 89(6):068102, 2002.

Ethan M. Coven and Zbigniew H. Nitecki. On the Genesis of Symbolic Dynamics as we Know it. *Colloquium Mathematicum*, 110:227–242, 2008.

Barbara D'Ambrogi-Ola. *Inverse Problem for Fractional Brownian Motion with Discrete Data*. PhD thesis, University of Helsinki, 2009.

James Davidson and Nigar Hashimzade. Alternative Frequency and Time Domain Versions of Fractional Brownian Motion. *Econometric Theory*, 24(1):256–293, 2008.

Didier Delignières. Correlation Properties of (Discrete) Fractional Gaussian Noise and Fractional Brownian Motion. *Mathematical Problems in Engineering*, 2015:485623, 2015.

Guy Van den Broeck and Mathias Niepert. Lifted Probabilistic Inference for Asymmetric Graphical Models. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, pages 3599–3605, 2015.

Gustavo Didier and Vladas Pipiras. Integral Representations and Properties of Operator Fractional Brownian Motions. *Bernoulli*, 17(1):1–33, 2011.

Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting Cyclic Symmetry in Convolutional Neural Networks. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1889–1898, 2016.

C. R. Dietrich and G. N. Newsam. Fast and Exact Simulation of Stationary Gaussian Processes through Circulant Embedding of the Covariance Matrix. *SIAM Journal on Scientific Computing*, 18(4):1088–1107, 1997.

Shichang Du, Jun Lv, and Lifeng Xi. Degradation Process Prediction for Rotational Machinery Based on Hybrid Intelligent Model. *Robotics and Computer-Integrated Manufacturing*, 28(2):190–207, 2012.

Pablo Duboue. *The Art of Feature Engineering: Essentials for Machine Learning.* Cambridge University Press, 2020.

Antonio Dávalos, Meryem Jabloun, Philippe Ravier, and Olivieri Buttelli. Theoretical Study of Multiscale Permutation Entropy on Finite-Length Fractional Gaussian Noise. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1087–1091, 2018.

Robert J. Elliott and John van der Hoek. Fractional Brownian Motion and Financial Modelling. In Michael Kohlmann and Shanjian Tang (Eds.), *Mathematical Finance*, pages 140–151. Birkhäuser Basel, 2001.

Bilal Fadlallah, Badong Chen, Andreas Keil, and José Príncipe. Weighted-Permutation Entropy: A Complexity Measure for Time Series Incorporating Amplitude Information. *Physical Review E*, 87(2):022911, 2013.

Nils Finke and Marisa Mohr. A Priori Approximation of Symmetries in Dynamic Probabilistic Relational Models. In Stefan Edelkamp, Ralf Möller, and Elmar Rueckert (Eds.), *KI 2021: Advances in Artificial Intelligence*, pages 309–323. Springer, 2021.

Nils Finke and Ralf Möller. On the Construction of Symmetries and Retaining Lifted Representations in Dynamic Probabilistic Relational Models. *Advances in Science, Technology and Engineering Systems Journal*, 7(1), 2022. In Submission.

Nils Finke, Marcel Gehrke, Tanya Braun, Tristan Potten, and Ralf Möller. Investigating Matureness of Probabilistic Graphical Models for Dry-Bulk Shipping. In Manfred Jaeger and Thomas Dyhre Nielsen (Eds.), *Proceedings of the 10th International Conference on Probabilistic Graphical Models*, volume 138 of *Proceedings of Machine Learning Research*, pages 197–208. PMLR, 2020.

Nils Finke, Marisa Mohr, Alexander Lontke, Marwin Züfle, Samuel Kounev, and Ralf Möller. Recommendations for Data-Driven Degradation Estimation with Case Studies from Manufacturing and Dry-Bulk Shipping. In Samira Cherfi, Anna Perini, and Selmin Nurcan (Eds.), *Research Challenges in Information Science*, pages 189–204. Springer, 2021.

Nils Finke, Ralf Möller, and Marisa Mohr. Multivariate Ordinal Patterns for Symmetry Approximation in Dynamic Probabilistic Relational Models. In Guodong Long, Xinghuo Yu, and Sen Wang (Eds.), *AI 2021: Advances in Artificial Intelligence*, pages 543–555. Springer International Publishing, 2022.

Brian J.W. Fleming, D. Yu, R.G. Harrison, and D. Jubb. Wavelet-based detection of coherent structures and self-affinity in financial data. *The European Physical Journal B: Condensed Matter and Complex Systems*, 20(4):543–546, 2001.

Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised Scalable Representation Learning for Multivariate Time Series. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché Buc, E. Fox, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Marcel Gehrke, Tanya Braun, and Ralf Möller. Lifted Dynamic Junction Tree Algorithm. In *Proceedings of the International Conference on Conceptual Structures*, pages 55–69. Springer, 2018.

Marcel Gehrke, Ralf Möller, and Tanya Braun. Taming Reasoning in Temporal Probabilistic Relational Models. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2592–2599. IOS Press, 2020.

Luis A. Gil-Alana. A Fractional Multivariate Long Memory Model for the US and the Canadian Real Output. *Economics Letters*, 81(3):355–359, 2003.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Yanping Guo, Yingying Chen, Qianru Yang, Fengzhen Hou, Xinyu Liu, and Yan Ma. Multi-Scale Permutation Entropy: A Potential Measure for the Impact of Sleep Medication on Brain Dynamics of Patients with Insomnia. *Entropy*, 23(9), 2021.

Jacques Hadamard. Les Surfaces à Courbures Opposées et leurs Lignes Géodésiques. *Journal de Mathématiques Pures et Appliquées*, 4:27–74, 1898.

Mattis Hartwig, Marisa Mohr, and Ralf Möller. Constructing Gaussian Processes for Probabilistic Graphical Models. In *Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference (FLAIRS-33)*, pages 57–62, 2020.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, 2nd edition, 2009.

Shaobo He, Kehui Sun, and Huihai Wang. Multivariate Permutation Entropy and its Application for Complexity Analysis of Chaotic Systems. *Physica A: Statistical Mechanics and its Applications*, 461:812–823, 2016.

Gustav A. Hedlund. Sturmian Minimal Sets. *American Journal of Mathematics*, 66(4):605–620, 1944.

Munaf Yousif Hmood and Amjad Hibtallah Hamza. Discrete Wavelet Based Estimator for the Hurst Parameter of Multivariate Fractional Brownian Motion. *Journal of Physics: Conference Series*, 1879(3):032033, 2021.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a Freely Accessible Critical Care Database. *Scientific Data*, 3:160035, 2016.

Karsten Keller and Heinz Lauffer. Symbolic Analysis of High-Dimensional Time Series. *International Journal of Bifurcation and Chaos*, 13(09):2657–2668, 2003.

Karsten Keller, Sergiy Maksymenko, and Inga Stolz. Entropy determination based on the ordinal structure of a dynamical system. *Discrete and Continuous Dynamical Systems - Series B*, 20(10), 2015.

Karsten Keller, Teresa Mangold, Inga Stolz, and Jenna Werner. Permutation Entropy: New Ideas and Challenges. *Entropy*, 19(3), 2017.

Karsten Keller. Permutations and the Kolmogorov-Sinai entropy. *Discrete & Continuous Dynamical Systems*, 32(3):891, 2012.

Kristian Kersting. Lifted Probabilistic Inference. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 33–38. IOS Press, 2012.

Joo-Ho Choi Kim and Dawn An Nam H. Remaining Useful Life Prediction of Rolling Element Bearings Using Degradation Feature Based on Amplitude Decrease at Specific Frequencies. *Structural Health Monitoring*, 2017.

Hack-Eun Kim, Andy C.C. Tan, Joseph Mathew, Eric Y. H. Kim, and Byeong-Keun Choi. Machine Prognostics Based on Health State Estimation Using SVM. In Joe E. Amadi-Echendu, Roger Willett, Kerry Brown, and Joseph Mathew (Eds.), *Asset condition, information systems and decision models*, pages 169–186. Springer, 2012.

Stefan Kramer. A Brief History of Learning Symbolic Higher-Level Representations from Data (And a Curious Look Forward). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4868–4876, 2020.

Joseph B. Kruskal and Mark Liberman. The Symmetric Time Warping Problem: From Continuous to Discrete. In *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley Publishing Co., 1983.

Frédéric Lavancier, Anne Philippe, and Donatas Surgailis. Covariance Function of Vector Self-Similar Processes. *Statistics & Probability Letters*, 79(23):2415–2421, 2009.

Xiang Li, Wei Zhang, and Qian Ding. Deep Learning-Based Remaining Useful Life Estimation of Bearings Using Multi-Scale Feature Extraction. *Reliability Engineering & System Safety*, 182:208–218, 2019.

Hongmei Li, Jinying Huang, Xiwang Yang, Jia Luo, Lidong Zhang, and Yu Pang. Fault Diagnosis for Rotating Machinery Using Multiscale Permutation Entropy and Convolutional Neural Networks. *Entropy*, 22(8), 2020.

Chun-Feng Li. Rescaled-Range and Power Spectrum Analyses on Well-Logging Data. *Geophysical Journal International*, 153(1):201–212, 2003.

Li Li. Response to Comments on PCA Based Hurst Exponent Estimator for fBm Signals Under Disturbances. *IEEE Transactions on Signal Processing*, 57(7):2840–2846, 2009.

Tiebing Liu, Wenpo Yao, Min Wu, Zhaorong Shi, Jun Wang, and Xinbao Ning. Multiscale Permutation Entropy Analysis of Electrocardiogram. *Physica A: Statistical Mechanics and its Applications*, 471:492–498, 2017.

Panagiotis Loukopoulos, George Zolkiewski, Ian Bennett, Suresh Sampath, Pericles Pilidis, X. Li, and David Mba. Abrupt Fault Remaining Useful Life Estimation Using Measurements from a Reciprocating Compressor Valve Failure. *Mechanical Systems and Signal Processing*, 121:359 – 372, 2019.

Dizhen Ma, Shaobo He, and Kehui Sun. A Modified Multivariable Complexity Measure Algorithm and Its Application for Identifying Mental Arithmetic Task. *Entropy*, 23:931, 2021.

Natallia Makarava. *Bayesian Estimation of Self-Similarity Exponent*. PhD thesis, Universität Potsdam, 2012.

Benoit B. Mandelbrot and John W. Van Ness. Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Review*, 10(4):422–437, 1968.

Benoit B. Mandelbrot and James R. Wallis. Noah, Joseph, and Operational Hydrology. *Water Resources Research*, 4(5):909–918, 1968.

Yuliya Mishura and Ûliâ S. Mišura. *Stochastic Calculus for Fractional Brownian Motion and Related Processes*. Number 1929 in Lecture Notes in Mathematics. Springer, 2008.

Marisa Mohr and Karsten Keller. Aus den Auf- und Abwärtsbewegungen einer Zeitreihe lernen. *Mitteilungen der Deutschen Mathematiker-Vereinigung*, 30(1):25–29, 2022.

Marisa Mohr and Ralf Möller. A Summary of Canonical Multivariate Permutation Entropies on Multivariate Fractional Brownian Motion. *Advances in Science, Technology and Engineering Systems Journal*, 6(5):107–124, 2021.

Marisa Mohr and Ralf Möller. Ordering Principal Components of Multivariate Fractional Brownian Motion for Solving Inverse Problems. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2021 (APSIPA-ASC)*, 2021.

Marisa Mohr and Frederik Timm. Ein sicherer KOSMoS für den unternehmensübergreifenden Austausch von Daten. *t3n Magazin*, 68, 2022. In Press.

Marisa Mohr, Nils Finke, and Ralf Möller. On the Behaviour of Permutation Entropy on Fractional Brownian Motion in a Multivariate Setting. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2020 (APSIPA-ASC)*, pages 189–196, 2020.

Marisa Mohr, Florian Wilhelm, Mattis Hartwig, Ralf Möller, and Karsten Keller. New Approaches in Ordinal Pattern Representations for Multivariate Time Series. In *Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference (FLAIRS-33)*, pages 124–129, 2020.

Marisa Mohr, Christian Becker, Ralf Möller, and Matthias Richter. Towards Collaborative Predictive Maintenance Leveraging Private Cross-Company Data. In Ralf H. Reussner, Anne Koziolek, and Robert Heinrich (Eds.), *INFORMATIK 2020*, pages 427–432. Gesellschaft für Informatik, 2021.

Marisa Mohr, Florian Wilhelm, and Ralf Möller. On the Behaviour of Weighted Permutation Entropy on Fractional Brownian Motion in the Univariate and Multivariate Setting. *The International FLAIRS Conference Proceedings*, 34, 2021.

Marisa Mohr. The Mystery of Entropy: Measuring Unpredictability in Machine Learning. *inovex Blog*, May 2019. `https://www.inovex.de/blog/the-mystery-of-entropy-how-to-measure-unpredictability-in-machine-learning/`.

Marisa Mohr. The Mystery of Time Series: Why Dealing with Time Series is Difficult. *inovex Blog*, July 2021. `https://www.inovex.de/de/blog/the-mystery-of-time-series/`.

George E. Moody. Spontaneous Termination of Atrial Fibrillation: A Challenge from Physionet and Computers in Cardiology 2004. In *Computers in Cardiology, 2004*, pages 101–104, 2004.

Francesco Carlo Morabito, Domenico Labate, Fabio La Foresta, Alessia Bramanti, Giuseppe Morabito, and Isabella Palamara. Multivariate Multi-Scale Permutation Entropy for Complexity Analysis of Alzheimer's Disease EEG. *Entropy*, 14(7):1186–1202, 2012.

Cristina Morel and Anne Humeau-Heurtier. Multiscale Permutation Entropy for Two-Dimensional Patterns. *Pattern Recognition Letters*, 150:139–146, 2021.

Kevin P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, Computer Science Division, 2002.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning Series. MIT Press, 2012.

Audun Myers and Firas A. Khasawneh. On the Automatic Parameter Selection for Permutation Entropy. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(3):033130, 2020.

Bhattacharya Sharad Nath and Bhattacharya Mousumi. Long Memory in Stock Returns: A Study of Emerging Markets. *Iranian Journal of Management Studies*, 5(2):67–88, 2012.

Patrick Nectoux, Rafael Gouriveau, Kamal Medjaher, Emmanuel Ramasso, Brigitte Chebel-Morello, Noureddine Zerhouni, and Christophe Varnier. PRONOSTIA: An Experimental Platform for Bearings Accelerated Degradation Tests. In *Proceedings of IEEE International Conference on Prognostics and Health Management*, 2012.

Nicoletta Nicolaou and Julius Georgiou. The Use of Permutation Entropy to Characterize Sleep Electroencephalograms. *Clinical EEG and Neuroscience*, 42(1):24–28, 2011.

Mathias Niepert and Guy Van den Broeck. Tractability through Exchangeability: A New Perspective on Efficient Probabilistic Inference. In *AAAI-14 Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 2467–2475. AAAI Press, 2014.

Tolga Esat Oezkurt and Tayfun Akgül. IS PCA Reliable for the Analysis of Fractional Brownian Motion? In *Proceedings of 13th European Signal Processing Conference*, pages 1–4, 2005.

Tolga Esat Oezkurt, Tayfun Akgül, and Suleyman Baykut. Principal Component Analysis of the Fractional Brownian Motion for $0 < H < 0.5$. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, 2006.

Erkan Ozkat. The Comparison of Machine Learning Algorithms in Estimation of Remaining Useful Lifetime. In *Proceedings of 9th International BTKS*, 2019.

Zuozhou Pan, Zong Meng, Zijun Chen, Wenqing Gao, and Ying Shi. A Two-Stage Method Based on Extreme Learning Machine for Predicting the Remaining Useful Life of Rolling-Element Bearings. *Mechanical Systems and Signal Processing*, 144:106899, 2020.

E. Perrin, R. Harba, C. Berzin-Joseph, I. Iribarren, and A. Bonami. n-th-Order Fractional Brownian Motion and Fractional Gaussian Noises. *IEEE Transactions on Signal Processing*, 49(5):1049–1059, 2001.

François Petitjean, Jordi Inglada, and Pierre Gancarski. Satellite Image Time Series Analysis Under Time Warping. *IEEE Transactions on Geoscience and Remote Sensing*, 50(8), 2012.

Albert B. Piek, Inga Stolz, and Karsten Keller. Algorithmics, Possibilities and Limits of Ordinal Pattern Based Entropies. *Entropy*, 21(6):547, 2019.

Vladas Pipiras and Murad S. Taqqu. *Long-Range Dependence and Self-Similarity.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2017.

David Poole. First-order Probabilistic Inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 985–991, 2003.

Hamed Rabiei, Olivier Coulon, Julien Lefevre, and Frederic J. P. Richard. Surface Regularity via the Estimation of Fractional Brownian Motion Index. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 30:1453–1460, 2021.

Yomna Rayan, Yasser Mohammad, and Samia A. Ali. Multidimensional Permutation Entropy for Constrained Motif Discovery. In Ngoc Thanh Nguyen, Ford Lumban Gaol, Tzung-Pei Hong, and Bogdan Trawiński (Eds.), *Intelligent Information and Database Systems*, Lecture Notes in Computer Science, pages 231–243. Springer, 2019.

Maik Riedl, Andreas Müller, and Niels Wessel. Practical Considerations of Permutation Entropy: A Tutorial Review. *The European Physical Journal Special Topics*, 222, 2013.

Stan Salvador and Philip Chan. FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space. *KDD Workshop on Mining Temporal and Sequential Data*, pages 70–80, 2004.

Taisuke Sato. A Statistical Learning Method for Logic Programs with Distribution Semantics. In *Proceedings of the 12th International Conference on Logic Programming*, pages 715–729, 1995.

Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. $E(n)$ Equivariant Graph Neural Networks. In *Proceedings of the 38th ICML*, 2021.

Venkat A. Setty and A. S. Sharma. Characterizing Detrended Fluctuation Analysis of Multifractional Brownian Motion. *Physica A: Statistical Mechanics and its Applications*, 419:698–706, 2015.

Diego F. Silva and Gustavo E. A. P. A. Batista. Speeding Up All-Pairwise Dynamic Time Warping Matrix Calculation. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 837–845. Society for Industrial and Applied Mathematics, 2016.

Parag Singla, Aniruddh Nath, and Pedro Domingos. Approximate Lifting Techniques for Belief Propagation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, page 2497–2504. AAAI Press, 2014.

Mathieu Sinn and Karsten Keller. Estimation of Ordinal Pattern Probabilities in Gaussian Processes with Stationary Increments. *Computational Statistics & Data Analysis*, 55(4):1781–1790, 2011.

Stilian A. Stoev and Murad S. Taqqu. How Rich is the Class of Multifractional Brownian Motions? *Stochastic Processes and their Applications*, 116(2):200–221, 2006.

Inga Stolz and Karsten Keller. A General Symbolic Approach to Kolmogorov-Sinai Entropy. *Entropy*, 19(12):675, 2017.

Cui Su, Zhenhu Liang, Xiaoli Li, Duan Li, Yongwang Li, and Mauro Ursino. A Comparison of Multiscale Permutation Entropy Measures in On-Line Depth of Anesthesia Monitoring. *PLOS ONE*, 11(10), 2016.

Francisco Traversaro, Francisco O. Redelico, Marcelo R. Risk, Alejandro C. Frery, and Osvaldo A. Rosso. Bandt-Pompe symbolization dynamics for time series with tied values: A data-driven approach. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):075502, 2018.

Harry van Zanten. When is a linear combination of independent fBm's equivalent to a single fBm? *Stochastic Processes and their Applications*, 117(1):57–70, 2007.

Deepak Venugopal and Vibhav Gogate. Evidence-Based Clustering for Scalable Inference in Markov Logic. In Toon Calders, Floriana Esposito, Eyke Hüllermeier, and Rosa Meo (Eds.), *Machine Learning and Knowledge Discovery in Databases*, pages 258–273. Springer, 2014.

Fengtao Wang, Xiaofei Liu, Gang Deng, Xiaoguang Yu, Hongkun Li, and Qingkai Han. Remaining Life Prediction Method for Rolling Bearing Based on the Long Short-Term Memory Network. *Neural Processing Letters*, 50(3), 2019.

Hermann Weyl. *Symmetry*. Princeton University Press, 1952.

David H. Wolpert and William G. Macready. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 1997.

Jun Wu, Chaoyong Wu, Shuai Cao, Siu Wing Or, Chao Deng, and Xinyu Shao. Degradation Data-Driven Time-To-Failure Prognostics Approach for Rolling Element Bearings in Electrical Machines. *IEEE Transactions on Industrial Electronics*, 66(1), 2019.

Jianan Xia, Pengjian Shang, Jing Wang, and Wenbin Shi. Permutation and Weighted-Permutation Entropy Analysis for the Complexity of Nonlinear Time Series. *Communications in Nonlinear Science and Numerical Simulation*, 31(1):60–68, 2016.

Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1317–1322, 2016.

Yi Yin and Pengjian Shang. Weighted Multiscale Permutation Entropy of Financial Time Series. *Nonlinear Dynamics*, 78:2921–2939, 2014.

Bin Zhang, Lijun Zhang, and Jinwu Xu. Degradation Feature Selection for Remaining Useful Life Prediction of Rolling Element Bearings. *Quality and Reliability Engineering International*, 32(2), 2016.

Yuhuang Zheng. Predicting Remaining Useful Life Based on Hilbert–Huang Entropy with Degradation Model. *Journal of Electrical and Computer Engineering*, 2019.

Luciano Zunino, Darío Gabriel Pérez, M. T. Martín, Mario Garavaglia, A. Plastino, and Osvaldo A. Rosso. Permutation Entropy of Fractional Brownian Motion and Fractional Gaussian Noise. *Physics Letters A*, 372(27):4768–4774, 2008.

List of Publications

The following blog articles, conference papers, journal articles, magazine articles and workshop contributions are peer-reviewed, accepted, and published during the period in which this dissertation was written, and thus are either part of the dissertation or have influenced the writing of it.

Blog Articles

Marisa Mohr. The Mystery of Time Series: Why Dealing with Time Series is Difficult. *inovex Blog*, July 2021. `https://www.inovex.de/de/blog/the-mystery-of-time-series/`

Marisa Mohr. The Mystery of Entropy: Measuring Unpredictability in Machine Learning. *inovex Blog*, May 2019. `https://www.inovex.de/blog/the-mystery-of-entropy-how-to-measure-unpredictability-in-machine-learning/`

Conference Papers

Nils Finke, Ralf Möller, and Marisa Mohr. Multivariate Ordinal Patterns for Symmetry Approximation in Dynamic Probabilistic Relational Models. In Guodong Long, Xinghuo Yu, and Sen Wang (Eds.), *AI 2021: Advances in Artificial Intelligence*, pages 543–555. Springer International Publishing, 2022

Marisa Mohr and Ralf Möller. Ordering Principal Components of Multivariate Fractional Brownian Motion for Solving Inverse Problems. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2021 (APSIPA-ASC)*, 2021

Nils Finke and Marisa Mohr. A Priori Approximation of Symmetries in Dynamic Probabilistic Relational Models. In Stefan Edelkamp, Ralf Möller, and Elmar Rueckert (Eds.), *KI 2021: Advances in Artificial Intelligence*, pages 309–323. Springer, 2021

Marisa Mohr, Florian Wilhelm, and Ralf Möller. On the Behaviour of Weighted Permutation Entropy on Fractional Brownian Motion in the Univariate and Multivariate Setting. *The International FLAIRS Conference Proceedings*, 34, 2021

Nils Finke, Marisa Mohr, Alexander Lontke, Marwin Züfle, Samuel Kounev, and Ralf Möller. Recommendations for Data-Driven Degradation Estimation with Case Studies from Manufacturing and Dry-Bulk Shipping. In Samira Cherfi, Anna Perini, and Selmin Nurcan (Eds.), *Research Challenges in Information Science*, pages 189–204. Springer, 2021

Marisa Mohr, Nils Finke, and Ralf Möller. On the Behaviour of Permutation Entropy on Fractional Brownian Motion in a Multivariate Setting. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2020 (APSIPA-ASC)*, pages 189–196, 2020

Marisa Mohr, Florian Wilhelm, Mattis Hartwig, Ralf Möller, and Karsten Keller. New Approaches in Ordinal Pattern Representations for Multivariate Time Series. In *Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference (FLAIRS-33)*, pages 124–129, 2020

Mattis Hartwig, Marisa Mohr, and Ralf Möller. Constructing Gaussian Processes for Probabilistic Graphical Models. In *Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference (FLAIRS-33)*, pages 57–62, 2020

Lucas Baier, Niklas Kühl, Gerhard Satzger, Marcel Hofmann, and Marisa Mohr. Handling Concept Drifts in Regression Problems - the Error Intersection Approach. In Norbert Gronau, Moreen Heine, Hanna Krasnova, and K. Poustcchi (Eds.), *Entwicklungen, Chancen und Herausforderungen der Digitalisierung: Proceedings der 15. Internationalen Tagung Wirtschaftsinformatik, WI 2020*, pages 210–224. GITO Verlag, 2020

Journal Articles

Marisa Mohr and Karsten Keller. Aus den Auf- und Abwärtsbewegungen einer Zeitreihe lernen. *Mitteilungen der Deutschen Mathematiker-Vereinigung*, 30(1):25–29, 2022

Marisa Mohr and Ralf Möller. A Summary of Canonical Multivariate Permutation Entropies on Multivariate Fractional Brownian Motion. *Advances in Science, Technology and Engineering Systems Journal*, 6(5):107–124, 2021

Magazine Articles

Marisa Mohr and Frederik Timm. Ein sicherer KOSMoS für den unternehmensübergreifenden Austausch von Daten. *t3n Magazin*, 68, 2022. In Press

Tobias Bux, Jonas Groß, Constantin Lichti, and Marisa Mohr. Projekt KOSMoS: Mit Blockchain transparent und firmenübergreifend warten. *atp magazin 03/2021*, 63(3):28–30, 2021

Christian Becker and Marisa Mohr. Federated Machine Learning: Über Unternehmensgrenzen hinaus aus Produktionsdaten lernen. *atp magazin 05/2020*, 62(5):28–30, 2020

Tobias Bux and Marisa Mohr. Blockchain-Lösungen für den produktionstechnischen Mittelstand/Blockchain Solutions for Medium-sized Production Engineering Companies – Proof of Confidence for Cross-Company Networking of Production and Process Data. *wt Werkstattstechnik online*, 110(04):201–204, 2020

Workshop Contributions

Marisa Mohr, Christian Becker, Ralf Möller, and Matthias Richter. Towards Collaborative Predictive Maintenance Leveraging Private Cross-Company Data. In Ralf H. Reussner, Anne Koziolek, and Robert Heinrich (Eds.), *INFORMATIK 2020*, pages 427–432. Gesellschaft für Informatik, 2021