UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE ESTUDIOS ESTADÍSTICOS

TESIS DOCTORAL

Quantifying chaos from time-series data through Lyapunov exponents:
a computational data science approach

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Julio Emilio Sandubete Galán

DIRECTORES

Lorenzo Escot Mangas
Simone Giannerini

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE ESTUDIOS ESTADÍSTICOS

TESIS DOCTORAL

QUANTIFYING CHAOS FROM TIME-SERIES DATA THROUGH LYAPUNOV EXPONENTS: A COMPUTATIONAL DATA SCIENCE APPROACH

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

JULIO EMILIO SANDUBETE GALÁN

DIRECTORES

LORENZO ESCOT MANGAS

SIMONE GIANNERINI

Quantifying chaos from time-series data through Lyapunov exponents: a computational data science approach

Memoria presentada para optar al título de Doctor por

D. Julio Emilio Sandubete Galán

Codirigida por

Dr. Lorenzo Escot Mangas
Dr. Simone Giannerini

Facultad de Estudios Estadísticos
Universidad Complutense de Madrid

Madrid 2020

A mis cuatro dimensiones

Agradecimientos

"Fuenteovejuna, todos a una".
— Lope de Vega

El embrión de esta tesis doctoral se gesta gracias a un regalo sorpresa de mi querida madre, un libro escrito por el Prof. D. Andrés Fernández Díaz, científico de pensamiento poliédrico, y a una inesperada llamada de un doctorando suyo, el Prof. D. Lorenzo Escot, mi co-supervisor, desencadenando, tras algo más de 9 meses, en la actual disertación que se presenta en esta memoria. El tiempo terrestre transcurre a la velocidad que la percepción humana le asigna, mientras que el espacio, se expande a medida que el universo se contrae ¿o no?. Tiempo y espacio, hermosa dicotomía!

Siento un amor incondicional y una eterna gratitud por mis *cuatro dimensiones*, sin ellos, quizás, no estaría ahora escribiendo estas escuetas líneas de agradecimiento ya que el nacimiento per sé, es condición necesaria pero no suficiente para culminar el proceso evolutivo. Sin ellos, sin ustedes, el ser y estar sería una simple concatenación de vocales y consonantes sin alma. Un recuerdo en el olvido. Despojarse de lo aprendido, de algo nos ha valido, sin caer en el descuido, de lo vivido. Afortunado me siento al haberos conocido. Como diría aquel, con ustedes, hasta el infinito y más allá!

A renglón seguido, aparece mi co-supervisor, el Prof. D. Lorenzo Escot. Cuando el camino se vuelve árido, una sola gota de agua sirve para calmar la sed. Gracias por ser ese manantial al que acudir cuando andaba algo distraído y sin saber muy bien a dónde ir ni desde dónde he venido. Tu humilde sabiduría honra a tu descarado sentido del humor. Gracias por ser y estar. Gracias por tu paciencia y comprensión. Gracias por compartir parte de tu tiempo conmigo. Este camino recién acaba de comenzar, caminante.

A la verita suya se encuentra mi otro co-supervisor, el Prof. D. Simone Giannerini. Grazie per l'accoglienza, la disponibilità e la vicinanza. Grazie per la vostra pazienza e comprensione. Grazie per aver partecipato a questo progetto. Grazie per i vostri consigli tecnici, il vostro aiuto e il vostro incoraggiamento in alcuni momenti difficili sono stati essenziali per realizzare questa tesi, grazie mille per la fiducia e per avermi dato questa opportunità. Il caos sarà il nostro punto d'incontro. Ci vediamo lì.

En el flanco derecho e izquierdo respectivamente se encuentran el Prof. D. Cristóbal Pareja Flores y el Prof. D. Daniel Gómez González. Mi enorme gratitud por hacerme sentir de la casa (y como en casa) a pesar de haber llegado con el tren en marcha. Gracias por vuestro entusiasmo, dedicación y entrega. Todo un ejemplo a seguir para las generaciones que os preceden.

Tras ellos, y no menos importante, aparece el resto del PDI y PAS de las distintas dependencias de nuestra querida Facultad de Estudios Estadísticos. Gracias por vuestro tiempo, predisposición, cariño y atención. Este agradecimiento lo hago extensible a mis compañeros de fatigas, gracias por vuestra compañía y tesón. Reiterar mi más profunda gratitud a todas las personas e instituciones que han participado de un modo u otro y me han acompañado a lo largo de este fructífero viaje. Gracias de corazón.

El trabajo de investigación que ha dado lugar a esta tesis doctoral ha sido benefactor de los fondos recibidos por parte de los proyectos de investigación RTI2018-094901-B-I00 del Ministerio de Ciencia e Innovación y FEM2014-56723-P del Ministerio de Economía y Competitividad. Además, el doctorando ha disfrutado de dos becas de ayuda a la movilidad para la realización de dos estancias de investigación predoctorales en el Real Colegio Complutense (Harvard University) y en el Dipartimento di Scienze Statistiche (Università di Bologna), dentro del programa propio de ayudas y acciones para el desarrollo de la investigación científica por parte de la Universidad Complutense de Madrid. Sirvan estas palabras para agradecer el recuerdo de lo vivido.

Finalmente, el doctorando ha sido benefactor de un contrato predoctoral como personal investigador en formación adscrito a la Facultad de Estudios Estadísticos. Mi más sincero agradecimiento y reconocimiento a nuestra Facultad de Estudios Estadísticos y por ende, a la Universidad Complutense. Mención especial a destacar serían las diferentes ayudas recibidas por parte del Grupo de Investigación UCM 'Data Analysis in Social and Gender Studies and Equality Policies'. Mi gratitud a los diferentes miembros del mismo y en especial, a su codirector el Prof. D. José Andrés Fernández Cornejo.

No me gustaría dejar pasar esta ocasión única sin antes hacer alusión a un breve extracto del pasional discurso declamado por el Prof. D. Lorenzo Escot en el Acto de Graduación el 20 de abril de 2018. Rezaba así: *"Ponédle ilusión a lo que hagáis, ponédle pasión, sin pasión, estaréis perdidos, sin pasión, se acabará el amor por lo que hacéis y sin amor, el día a día se volverá triste y tedioso, sin amor, no habrá esperanza"*.

Julio E. Sandubete Galán

Resumen

Cuantificando el caos en series temporales a través del exponente de Lyapunov: un enfoque desde la ciencia de datos computacional

Introducción

La aparición de la teoría del caos ha supuesto un cambio de paradigma en la ciencia. Los sistemas caóticos son sistemas dinámicos deterministas no lineales que pueden comportarse como un movimiento aparentemente errático e irregular. Tengan en cuenta que sí pudiésemos caracterizar un sistema caótico en algún sentido nos permitiría evidenciar que detrás de los mismos existe un sistema generador determinista a pesar de mostrar un comportamiento aparentemente aleatorio. Este hecho nos permitiría aprovechar ese carácter determinista para poder hacer predicciones y controlar las variables de estos sistemas dinámicos deterministas (caóticos). Los métodos y técnicas que persiguen contrastar la hipótesis del caos tratan de estimar los llamados *exponentes de Lyapunov* como una forma de caracterizar un sistema caótico. Hoy en día, cuantificar el caos a partir de datos de series temporales mediante este tipo de medidas cuantitativas de manera rigurosa está lejos de ser un ejercicio trivial y plantea una serie de retos teóricos y prácticos.

Objetivos y resultados

Esta tesis doctoral plantea una discusión progresiva de los aspectos teóricos y prácticos que constituyen, en la actualidad, las principales ideas sobre la cuantificación del caos en series temporales a través de los exponentes de Lyapunov por diversos métodos computacionales. En este sentido, cuando se conoce el sistema dinámico, podemos calcular directamente el exponente de Lyapunov teórico. Sin embargo, en gran parte de los fenómenos reales no tenemos la ventaja de observar directamente el estado del sistema y mucho menos de conocer la forma funcional que genera la dinámica asociada a él. En lugar de eso, habrá una función de observación que incluirá un error de medición aditivo que genera observaciones en forma de serie temporal contaminada por dichos errores. En este contexto hemos examinado los efectos de considerar un término de error de medición en la función de observación.

Además, hemos llevado a cabo una revisión detallada acerca del procedimiento de reconstrucción del atractor que nos permita extraer algunas propiedades invariantes e.g., los exponentes de Lyapunov del proceso generador desconocido de los datos. Obsérvese que un punto clave para crear una reconstrucción adecuada del espacio de estado es fijar un criterio correcto para estimar robustamente los parámetros que intervienen en dicho procedimiento. En este sentido, hemos propuesto extender los enfoques heurísticos tradicionales basados en prescripciones a un enfoque estadístico basado en procedimientos de selección de modelos que tengan en cuenta tanto el criterio de información bayesiano como las técnicas de validación cruzada. El proceso de reconstrucción del atractor supone que el sistema dinámico se muestrea con una frecuencia temporal uniforme. Luego, los datos provendrían de una serie de valores medidos periódicamente. Sin embargo, hay situaciones en las que las observaciones son medidas a intervalos de tiempo variable. En este contexto, hemos extendido el procedimiento de reconstrucción de la dinámica del proceso generador desconocido proponiendo un nuevo algoritmo.

Una vez reconstruído el atractor queremos saber si el proceso generador presenta una dinámica simple, se comporta de manera caótica o puramente aleatoria. Existen dos corrientes principales a la hora de estimar el exponente de Lyapunov a partir de series temporales. Los métodos tradicionales directos que no nos proporcionan estimadores consistentes y robustos ante la presencia de errores de medición como evidenciamos en esta tesis. Además, hay que notar que no permiten la construcción de test estadísticos formales. En este sentido, nos hemos centrado en los llamados métodos jacobianos indirectos que resuelven esas limitaciones. Hemos desarrollado un novedosa librería de R llamada **DChaos** que proporciona una interfaz para el análisis de series temporales caóticas. Proponemos un procedimiento de derivación analítica, en lugar de numérica del jacobiano necesario para la estimación de los exponentes de Lyapunov. Además, permite estimar consistemente el exponente de Lyapunov mediante diversos métodos computaciones globales y locales robustos y hacer inferencia estadística sobre su significación.

Conclusiones

Hemos considerado un marco global y local en el contexto de los métodos jacobianos indirectos a la hora de estimar el exponente de Lyapunov a partir de series temporales. En este sentido, los diferentes algoritmos propuestos presentan consistencia y robustez a la hora de caracterizar adecuadamente un sistema caótico en un entorno en el que se ha considerado el efecto provocado por errores de medición. Este hecho nos permitiría determinar si el proceso generador del fenómeno en cuestión presenta una dinámica simple, se comporta de manera caótica o puramente aleatoria a partir de datos de series temporales contaminadas por errores de medición.

Abstract

Quantifying chaos from time-series data through Lyapunov exponents: a computational data science approach

Introduction

Chaos theory has been hailed as a revolution of thoughts and attracting ever increasing attention of many scientists from diverse disciplines. Chaotic systems are nonlinear deterministic dynamic systems which can behave like an apparently erratic and irregular motion. Keep in mind that if we could characterise a chaotic system in some sense it would allow us to evidence that a deterministic generating system exists behind that chaotic system in spite of showing an apparently random behaviour. This fact would provide us to take advantage of this deterministic character to be able to make predictions and control over the variables of these (chaotic) deterministic dynamic systems. Methods and techniques related to test the hypothesis of chaos try to estimate the so-called *Lyapunov exponents* as a way of characterising a chaotic system. Nowadays quantifying chaos from time-series data through this kind of quantitative measure in a rigorous fashion is far from being a trivial exercise and poses a number of theoretical and practical challenges.

Objectives and results

This doctoral dissertation will lead us through a progressive discussion of those theoretical and practical aspects which constitute, at present time, the main and successful ideas about quantifying chaos from time-series data through the Lyapunov exponents by various computational methods. In this sense when dynamic systems are known, we can directly calculate the theoretical Lyapunov exponent. However in most real-world time series we do not have the advantage of observing directly the state of the dynamic system let alone knowing the functional form that generate the dynamic associated with it. Instead of that, there will be an observer function that include an additive measurement error which generates observations as a noise-contaminated time-series data. In this context we have discussed the effects of considering a measurement noise term on the observation function.

We have also provided a detailed review about the embedding procedure which allows us to get all the invariant properties e.g., Lyapunov exponents about the unknown data-generating process. Note that a key point to create a suitable reconstruction of the state space is to fix a right criteria for estimating the embedding parameters robustly. In this sense we have proposed extend the traditional heuristic approaches based on prescriptions to a statistical approach based on model selection procedures taking into account the Bayesian information criterion and leave one out cross-validation techniques. In addition, the attractor reconstruction procedure assumes that the dynamic system is sampled with uniform-time frequency. That is, the data does not come from a series of values measured periodically in time. However, there has been increasing interest in situations where observations are characterised by its almost continuity. In this context we have extended the embedding procedure from time-series data with non-uniform time-frequency proposing a novel algorithm based on those results.

Moving on, once we reconstruct the attractor we want to know if the unknown data-generating process presents a simple dynamic, behaves chaotically or comes from a purely stochastic process. There are two main methods in the literature for estimating the Lyapunov exponent from time-series data. On the one hand, the traditional direct methods which do not provide us consistent and robust estimators to the presence of measurement errors as we have illustrated empirically. The asymptotic distribution of the direct estimator does not exist which means that it does not allow the building of formal tests. On the other hand, we have focused on the so-called jacobian indirect methods which solve those disadvantages. In this context we have developed a novel R package called **DChaos** which provides an R interface for chaotic time series analysis. We propose an analytical derivation procedure, rather than numerically of the jacobian needed for the estimation of the Lyapunov exponents. R users might make statistical inference about Lyapunov exponents significance and test consistently the hypothesis of chaos from time-series data by various global and local jacobian indirect methods.

Conclusion

We have considered a global and local theoretical framework based on the jacobian indirect methods when estimating the Lyapunov exponent from time-series data. In this sense, the different algorithms and computational methods proposed present consistency and robustness for characterising correctly a chaotic system in an environment where the effects caused by measurement errors has been considered. This fact would allow us to determine whether the unknown data-generating process behind the phenomenon in question presents a simple dynamic, behaves chaotically or comes from a purely random motion from noise-contaminated time-series data.

Contents

Chapter 1

Introduction

"Remember to look up at the stars and not down at your feet. Try to make sense of what you see and wonder about what makes the universe exist. Be curious. Don't just give up!".

— Stephen Hawking

Abstract: This chapter presents a summary of the main arguments that have motivated the elaboration of this PhD thesis (Section 1.1), taking them as a starting point to achieve the aims set out in Section 1.2. In addition, the content and structure of this doctoral dissertation are established in Section 1.3.

There are PhD theses that are born under strict planning, they refer to a specific topic and their steps are almost determined in the previous scheme that sets them in motion. The doctoral student who breaks into them has *hardly anything to do* with getting on the train and devoting his effort to learning, adapting and carrying out that rhythm of work. The current PhD thesis does not belong to that group, but represents the other extreme of scientific reality. Four years ago there were many unconnected ideas and projects to be put into practice, more than one doctoral programme to be enrolled in, and so many ways of thinking about what would it be the first step. The only common point of all of them was the word *chaos*. The ideas proposed in this doctoral dissertation will be supported by the statistical analysis, the suitable computational tools and the know-how about chaotic dynamic systems. Let us remark the main arguments that have motivated the elaboration of this PhD thesis since taking them as a starting point as we will show in the remaining pages.

1.1 General motivation

Chaos theory has been hailed as a revolution of thoughts and attracting ever increasing attention of many scientists from diverse disciplines. Particularly in statistics and data science there is an increasing number of publications who are interested on chaos. Nowadays the explosive growth of data, zettabytes of information, are pouring into computers from many sources, both scientific and commercial, and there is an emerging needed to analyse and understand the dynamic behind this huge amount of data. Although data analysis is not something new, the volume, velocity, variety, veracity, value and sources of information have been transformed in such a way in the last century. Hence it is necessary to develop novel techniques and algorithms that respond to the great challenges in this context. Let us point out two inspiring examples which might provide us some research questions.

For instance, in cosmology, data is being generated at an awesome rate by telescopes, satellites and interferometers scanning the skies. Signals containing information on the dynamics behind millions of astronomical bodies and objects are stored. Astronomers need automated ways to analyse their data to detect and characterise such objects as planets, stars, constellations, galaxies, black holes and so on. In this sense we doubt if data models are applicable directly because the true data-generating process is unknown. Thus we do not have the advantage of observing directly the state of the system let alone knowing the functional form that generate the dynamic associated with it. Hence it is assumed that all information available is a noise-contaminated signal as a time-series data. So how can we extract relevant information about the unknown underlying generator system behind it from just time-series data?; how the measurement noise might affect the detection of possible chaotic behaviour inside those noise-contaminated signals?.

The analysis of financial data makes up another challenging issue related to the model-data paradox. The greatest difficulty in applying chaos theory in empirical studies is related to the quantity and quality of dataset. Unlike experimental sciences where one can carry out experiments and simulations in the laboratory obtaining a large and clean dataset, most economic time series consist of daily, weekly, monthly, quarterly or annual noise-contaminated data. Particularly, in financial markets, the emergence of new technologies have had important implications for both the quantity and quality of existing data in finance. Nowadays trades realised by traders are replaced by algorithmic trades executed in an automated way providing a huge amount of data over many intra-daily disaggregated time-frequencies. In this sense we were interested to know if some invariant properties about the unknown underlying dynamic system that generates the time-series data e.g., the Lyapunov exponents are affected or not by the time-frequency considered in the sampling and if so, how is it affected.

The untiring search for answering those questions (and others that came up later) motivated the beginning of this doctoral dissertation setting its theoretical framework. According to existing literature on dynamic systems and nonlinear time series analysis several methodologies, procedures and algorithms have been developed, applied and implemented to look into the complex dynamic behaviour behind time-series data. Generally speaking, these methods can be divided into three main groups which focus on different features, i.e., self-similarity, attractor invariants and entropy respectively. Although they are from different perspectives, these approaches are closely related to each other. As far as this PhD thesis is concerned we will focus exclusively on the second one which is mainly based on chaos theory.

Traditionally the study of a phenomena with a complex and irregular evolution has been carried out assuming that the underlying dynamic, which generates this complexity, should be represented through stochastic processes. This approach has been propitiated, in part, because the solutions from deterministic systems (perfectly regular, ordered and periodic) were incapable of reproducing the complex dynamics observed in real phenomena. In fact, the maximum degree of complexity that deterministic systems could describe was restricted to quasi-periodic movements. Nowadays, it is generally accepted that chaotic systems are nonlinear deterministic dynamic systems which can behave like an erratic and apparently random motion.

In this sense a challenging question in chaos theory from a statistical perspective would be how to differentiate chaotic motions from purely random fluctuations. That is, how to distinguish whether an apparently erratic, non-cyclical and aperiodic dynamic system is random or chaotic. Keep in mind that if we could characterise a chaotic dynamic system in some way it would allow us to evidence that a deterministic generating system exists behind that chaotic system in spite of showing an apparently random dynamic behaviour. This fact would provide us to take advantage of this deterministic character to be able to make predictions and control over the variables of these (chaotic) deterministic dynamic systems. Let us discuss the particular objetives set out in this doctoral dissertation.

1.2 Particular objectives

This section provides a summary of the main *aims* proposed by this PhD thesis taking them as the research questions which we will try to answer throughout the current dissertation. Particularly, we are interested in providing a response to the following issues.

As far as this doctoral dissertation is concerned we must point out that in deterministic dynamic systems is not possible to give a rigorous definition that account for all the aspects that the term chaos implies so far. The mathematical (and statistical) foundation of chaos theory are still not well established (Section 2.1). So we are going to adopt an operational approach in order to characterise a chaotic dynamic system based on the study of stability which plays a crucial role in the dynamics behind the system.

We will focus on local definitions which considers the behaviour of a system inside an attractor once the dynamic systems, i.e., the orbits, have reached some of the dynamic equilibria. In this context if a dissipative nonlinear deterministic system is locally unstable, then there is the possibility that no matter how close two initial values are to each other, they will lead to drastically different orbits or trajectories. In this case, the ultimate state cannot be a fixed point, a periodic point, a limit point, a limit cycle, a quasi-periodic limit cycle or a limit torus. Instead, each realisation is then almost indistinguishable from that of a stochastic process, see Section 2.2.

Thus, randomness can be generated by a strictly deterministic system. This attracting set called strange attractor can be described as a feature of chaos which denote a finite region of the state space by an infinite number of points where an orbit or trajectory inside it will show an aperiodic cycle. That is, there is no regular finite period (tends to infinity) and it is highly unstable. In order to study the stability of an orbit or trajectory inside this kind of attractor we refer to the so-called Lyapunov exponents. So quantifying chaos through this kind of quantitative measure is a key point for understanding chaos. Objective 1: *it will be to provide a detailed review of the state-of-the-art about the so-called Lyapunov exponent and how it can be used to characterise a chaotic dynamic system, see Section 2.3.*

Since it was realised that even simple and deterministic dynamic systems can produce trajectories which look as a random process, there occurred some obvious questions: how can one distinguish chaos from randomness in real datasets?; how can we extract from time-series data relevant information about the unknown underlying generator system of them?. When dynamic systems are known, we can directly calculate both the largest and the full spectrum of the theoretical Lyapunov exponent value to check when the dynamic system is in a regime of chaotic behaviour (Section 2.4). However in most real-world observed time series the data-generating process is rarely known a priori. Then we are also interested in studying what happens when we assume that the true underlying generator system is unknown. Unfortunately, those theoretical results are not directly applicable when moving into the field of nonlinear time serie analysis. The empirical analysis of chaotic dynamic systems is based on the study of observed time-series data.

Thus if we take into account the empirical environment we do not have the advantage of observing directly the state of the dynamic system let alone knowing the functional form that generates the dynamic associated with it. Instead of that, there will be an observer function that include an additive measurement error which generates observations. As far as this doctoral dissertation is concerned we will consider only one observable variable. Hence it is assumed that all information available is the noise-contaminated sequence as a scalar strictly stationary univariate time serie.

We consider that it is appropriate to add a measurement noise term in the observer function because most real-world observed time-series data are usually noise-contaminated signals. Objective 2: *it will be to know the effects of considering a normal multinomial measurement noise term on the observation function (Section 3.1). The presence of a measurement noise can lead to confusion between a chaotic deterministic system and a purely stochastic one.* In such cases the interplay between order and disorder, determinism and stochasticity, stability and instability or predictability and unpredictability could undergoing profound changes with very exciting implications on nonlinear time series analysis and chaos.

Generally speaking devise a dynamic system to model some time-evolving phenomena involves specifying a rule that governs the evolution of the process through time and the space in which the process takes values. If the rule that governs the evolution of the process is well understood it may be possible to write down the dynamic system more or less from first principles, without much need for observational data except perhaps to fix some parameters (Section 2.2). Otherwise we may be much less clear about the underlying rules. That is, we would have to deal with the following situations:

(*i*) we might suspect that the system is governed by e.g., a set of difference equations, but not know what these are, even we may be uncertain what dynamic variables the equations involve; (*ii*) we may have available a quantity of experimental data relating to how the system is observed to change through time. Then in this uncertainty context we will consider that all information available is this sequence of measurements in the form of time-series data because we have assumed that the true data-generating process is unknown (Section 3.1). Thus it will not be possible to consider the true orbit of the dynamic system in the original state space. Instead of that, it will be possible to obtain an approximation (reconstruction) of it that result equivalent in a topological sense (equivalence in the dynamic and geometric properties).

The embedding procedure will provide us a framework to reconstruct an unknown dynamic system which gave rise to a given observed scalar time-series data simply by reconstructing a new state space out of successive values from time series. Objective 3: *it will be to provide a detailed review about the attractor reconstruction procedure in order to make sure that the embedding process allows us to get all the relevant information about the unknown underlying dynamic system that generates the time-series data (invariant properties).* Particularly, we are interested in knowing if the Lyapunov exponents have approximately the same value in both the true and the reconstructed state space (Section 3.2). This fact would allow us to test the hypothesis of chaos in the unknown original dynamic system.

A key point to create a suitable reconstruction of the state space is to fix a right criteria in order to estimate robustly the embedding parameters: the embedding dimension and the reconstruction time-delay. Researchers in this area usually estimate them using heuristic approaches and prescriptions that mostly rely on physical or geometrical arguments. The main drawbacks of these heuristic approaches are the following: (*i*) they are not intrinsically statistical; (*ii*) they lead to estimators whose properties are unknown or largely unexplored; (*iii*) they do not take into account the results of any model fit. Objective 4: *it will be to extend these traditional techniques to more robust procedures from a statistical perspective when it comes to consistently estimating the embedding parameters.* Particularly, we will focus on a statistical approach which solves those three disadvantages, see Section 3.2.

The statistical approach to state space reconstruction can be viewed as a best subset selection problem within this context. The idea behind it is to select the embedding parameters that provide the best fit in the estimation of any quantitative measure e.g., the Lyapunov exponents taking into account some information criteria. As far this doctoral dissertation is concerned we are going to use the statistical approach based on model selection procedures taking into account the Bayesian information criterion instead of heuristic techniques, and in some cases we will use leave one out cross-validation techniques. In any case, we really think that the information derived from the heuristic approaches might be still useful and should not be disregarded as a complementary information. Keep in mind that the embedding parameters belong to the parameter set to be estimated when selecting the best model by several estimation methods of the Lyapunov exponent, see Section 4.4.

The attractor reconstruction procedure assumes that the dynamic system is sampled with uniform-time frequency e.g., 1-month, 1-day, 1-hour, 30-min, 5-min, 1-min and so on. However, there has been increasing interest in situations where observations are characterised by its almost continuity. That is, the data does not come from a series of values measured periodically in time. For instance, in financial markets the emergence of new technologies have had important implications for both the quantity and quality of existing data in finance. Nowadays trades realized by traders are replaced by algorithmic trades executed in an automated way providing a huge amount of data over many intra-daily disaggregated time intervals.

The information generated by the interactions between traders who buys and sells financial instruments such as stocks, bonds, commodities, currencies, derivatives and so on is apparently encoded in frequencies which are not usually equally spaced in time e.g., tick-by-tick. Thus if we observe a particular financial asset i.e., a currency pair on the Foreign Exchange Market, the quotes or rates of this financial asset are sampling by tick-by-tick intervals which do not follow a constant rhythm.

Each tick will appear when there is a change, upward or downward, in the trade price of each transaction. Can we use this kind of information to build a dynamic system model, and if so, how is it related to the true data-generating process?. Objective 5: *it will be to extend the reconstruction procedure from time-series data with non-uniform time-frequency proposing a novel algorithm based on those results, see Section 6.2.*

Continuing in the context of the financial markets recently some papers have been published looking for nonlinearities and chaos using intra-daily financial time series. All of them found little evidence to support the presence of chaotic behaviours in their datasets but they found strong evidences of nonlinearities. Objective 6: *it will be to find a solution to the following chaos model-data paradox: why it is usually difficult to detect chaos in economic time series whereas it is apparently easy to show a chaotic behaviour in many theoretical models (Section 6.1).*

In order to explain why chaos is generic in theoretical economics models but elusive in data we will discuss a possible answer to it, at least when studying financial time series. We are also interested to know if some invariant properties about the unknown underlying dynamic system that generates those financial time-series data (Section 6.3) e.g., the Lyapunov exponents are affected or not by the time-frequency considered in the sampling and if so, how is it affected, see Section 6.4.

Moving on, once we reconstruct the attractor we want to know if the unknown data-generating process presents a simple dynamic, behaves chaotically or comes from a purely stochastic process. Methods and techniques related to test the hypothesis of chaos try to quantify the initial-value sensitive property estimating the so-called Lyapunov exponents. Note that if one knows the data-generating process behind the time series the theoretical Lyapunov exponent can be calculated directly using its own definition (Section 2.3). However we have now assumed that the true dynamics of the system is unknown because in most real-world observed time series the data-generating process is rarely known a priori.

There are two main methods in the literature that provide the estimated Lyapunov exponent from time-series data. Objective 7: *it will be to provide an overview about the main theoretical and computational aspects of the traditional direct methods which directly measures the growth rate of divergence between two trajectories with an infinitesimal difference in their initial conditions (Section 3.3).* We want to contrast empirically if the traditional direct methods provide us (or not) consistent Lyapunov exponent estimators and robustness to the presence of (small) measurement errors, see Section 3.4. In addition, we will discuss its main advantages and limitations.

Regarding the second method we will provide an overview about the main theoretical and practical features of the so-called jacobian indirect methods which try to estimate the jacobian of the underlying generating system and then those partial derivatives are used to compute the Lyapunov exponent. Note that the different contributions that compose the global jacobian indirect methods differ in the algorithm used for the estimation of the jacobian. Particularly we are going to discuss the global jacobian indirect method based on a neural net approach which solve the disadvantages of direct methods. Objective 8: *it will be to provide some contributions based on the global jacobian indirect methods for estimating the Lyapunov exponents (Section 4.1).* We are going to propose an analytical derivation procedure, rather than numerically of the jacobian needed for the estimation of the Lyapunov exponents.

At this point, we will provide a discussion about the optimal sample size used for estimating the partial derivatives of the jacobian and the optimal block length considered for estimating the Lyapunov exponents. We want to propose a new way of subsampling by blocks in this context based on the bootstrap method (Section 4.2). Then a feasible test statistics will be introduced and a one-sided test will be implemented for the purpose of testing the hypothesis of chaos based on the theoretical asymptotic properties of the Lyapunov exponent and the consistent estimator of its variance, see Section 4.3. We also want to know if these indirect methods report better estimates than direct methods and if so, does it provide consistent estimators and robustness in the presence of measurement errors?, see Section 4.4.

Then we are going to introduce a discussion about the right procedure to obtain the partial derivatives based on the local jacobian approach. The main differences with the global jacobian approach will also be highlighted. In addition, we will illustrate how to get a consistent estimator of the Lyapunov exponent based on the local jacobian approach (Section 5.1). Traditionally the main contributions proposed by the scientific community regarding the local jacobian indirect methods have focused on local linear approaches. Objective 9: *it will be to extend those results by proposing two novel alternatives that differ in the algorithm used for the estimation of the unknown nonlinear functions in the local jacobian.*

Particularly, we are going to introduce the following two local approaches: polynomial kernel method (Section 5.2); neural net kernel method (Section 5.3). We will also provide the suitable algorithms in order to computes the local jacobian from the best-fitted polynomial kernel model and neural net kernel model. In addition, we are interested in knowing if local jacobian indirect methods report better estimates than direct methods and global jacobian indirect methods and if so, we want to contrast if the local jacobian indirect methods provide us (or not) consistent Lyapunov exponent estimators and robustness to the presence of measurement errors, see Section 5.4.

Last but not least, we are going to provide a detailed review of the main algorithms publicly available at the Comprehensive R Archive Network (CRAN) related to nonlinear time series analysis and chaos. There are three R packages recently developed about this topic. These R packages are based on ideas inspired by the time series analysis (TISEAN) project. All those packages implement its algorithms based exclusively on the direct methods for estimating the Lyapunov exponent. Objective 10: *it will be to develop a novel R package called* **DChaos** *which will be the first R library to provide the global and local jacobian indirect methods in this context.*

R users might make statistical inference about Lyapunov exponents significance and test consistently the hypothesis of chaos from time-series data by various data-based computational methods. To sum up, the **DChaos** package will provide a R interface for researchers interested in the field of chaotic dynamic systems and nonlinear time series analysis and professors (and students) who teach (learn) courses related to those topics. These algorithms are publicly available at `www.CRAN.R-project.org/package=DChaos`, see R documentation at the end of this PhD thesis. Now let us provide the outline and structure of this doctoral dissertation.

1.3 Content and structure

The current PhD thesis is presented as a compendium of *several essays* and it is structured around *seven* chapters as follows. Chapters $2-3$ provide the theoretical framework established in the field of chaotic dynamic systems and nonlinear time series analysis. The aim is to provide a broad conceptual basis for a better understanding of the results presented in subsequent chapters. Chapters $4-5$ focus on the main contributions proposed in this doctoral dissertation. We propose various data-based computational methods for quantifying chaos from time-series data through the Lyapunov exponents. Chapter 6 reports an applied data analysis based on different real-world financial time-series data. Concluding remarks, some contributions and future research questions are provided in Chapter 7. The pertinent abstract is also incorporated at the beginning of this doctoral dissertation. The DChaos package structure and documentation has been included as an appendix. Now let us point out the storyline contained in each chapter.

Chapter 2 presents a summary of the main concepts and definitions already established in the discipline of dynamic systems and chaos theory. Section 2.1 discuss briefly some ideas regarding the pursuit of an agreed definition of chaos. Section 2.2 gives an overview about the theoretical framework established in the discipline of deterministic dynamic systems. It is also discussed the stability of its solutions which plays a crucial role in the dynamics behind the systems. Section 2.3 provides some of the theoretical results that allow us to establish the existence of chaotic trajectories as solutions to certain finite-dimensional nonlinear dissipative deterministic dynamic systems. It is also present the quantitative measure called Lyapunov exponent which quantifies the initial-value sensitivity property in a deterministic environment. Section 2.4 reports the empirical results of this chapter and a discussion of them. Section 2.5 contains some concluding remarks about this chapter.

Chapter 3 provides a summary of the main concepts and definitions already established in the discipline of nonlinear time series analysis and chaos theory. Section 3.1 gives some preliminaries features for detecting chaos from real-world observed time-series data which are usually noise-contaminated signals. Section 3.2 illustrates the correct procedure for the reconstruction of the state space from time-series data with uniform time-frequency. It is also discussed some criteria in order to estimate the embedding parameters. Section 3.3 provides an overview about the main traditional direct methods for estimating the Lyapunov exponents from time-series data and its limitations. Section 3.4 reports the empirical results of this chapter and a discussion of them. Section 3.5 contains some concluding remarks about it.

Chapter 4 presents a summary of the main concepts and definitions related to the global jacobian indirect method for estimating the Lyapunov exponents from time-series data. Section 4.1 describes some key features about the global neural net approach by approximating the unknown nonlinear system through a feed-forward single hidden layer neural network. It also provide the analytical derivation, rather than numerical, of the jacobian needed for the estimation of the Lyapunov exponents. Section 4.2 illustrates the correct procedure in order to obtain a consistent estimator of the Lyapunov exponent from the partial derivates computed previously by two different procedures and four ways of subsampling by blocks. Section 4.3 provides a statistical framework for testing the hypothesis of chaos based on the theoretical asymptotic properties of the neural net estimator of the Lyapunov exponent and the consistent estimator of its variance. Section 4.4 reports the empirical results of this chapter and a discussion of them. Section 4.5 contains some concluding remarks about this chapter.

Chapter 5 provides a summary of the main concepts and definitions related to two new local jacobian indirect methods for estimating the Lyapunov exponents from time-series data. Section 5.1 provides a discussion about the right procedure to obtain the partial derivatives based on the local jacobian approach. The main differences with the global jacobian approach are also highlighted. In addition, it illustrates how to get a consistent estimator of the Lyapunov exponent based on the local jacobian approach. Section 5.2 describes the key features about the local polynomial kernel method by approximating the unknown nonlinear system through a local polynomial of order greater than one. Section 5.3 provides the main details about the local neural net method by approximating the unknown nonlinear system through a feed-forward single hidden layer local neural net. Section 5.4 reports the empirical results of this chapter and a discussion of them. Section 5.5 contains some concluding remarks about it.

Chapter 6 discusses an application to certain real-world financial time-series data which are characterised by its almost continuity on trade showing a variable time-frequency between each quote. Section 6.1 gives a possible explanation to the chaos model-data paradox in finance. Section 6.2 illustrates the correct procedure for the reconstruction of the state space from time-series data with non-uniform time-frequency. It also present an algorithm to get the non-uniform delayed-coordinate embedding vectors. Section 6.3 provides an overview about the theoretical framework for estimating the Lyapunov exponents from tick-by-tick time-serie data. Section 6.4 reports the empirical results of this chapter and a discussion of them. Section 6.5 contains some concluding remarks about this chapter.

Chapter 7 presents a summary of the main closing remarks derived from the arguments and results provided by this PhD thesis (Section 7.1). Then we have remarked the main contributions proposed, communications presented at national and international conferences and research stays developed during the elaboration of this doctoral dissertation (Section 7.2). To conclude, some future research questions are briefly discussed in Section 7.3. The DChaos package structure and documentation has been included at the end of this PhD thesis. Now let us begin by taking the first steps leisurely but surely on this fruitful journey, walker.

Chapter 2

Dynamic systems, stability and chaos

"Caminante, son tus huellas, el camino y nada más;
Caminante, no hay camino, se hace camino al andar".

— Antonio Machado

Abstract: This chapter presents a summary of the main concepts and definitions already established in the discipline of dynamic systems and chaos theory. The aim is to provide a broad conceptual basis for a better understanding of the results presented in subsequent chapters. Section 2.1 discuss briefly some ideas regarding the pursuit of an agreed definition of chaos. Section 2.2 gives an overview about the theoretical framework established in the discipline of deterministic dynamic systems. It is also discussed the stability of its solutions which plays a crucial role in the dynamics behind the systems. Section 2.3 provides some of the theoretical results that allow us to establish the existence of chaotic trajectories as solutions to certain finite-dimensional nonlinear dissipative deterministic dynamic systems. It is also present the quantitative measure called Lyapunov exponent which quantifies the initial-value sensitivity property in a deterministic environment. Section 2.4 reports the empirical results of this chapter and a discussion of them. Section 2.5 contains some concluding remarks.

The references mentioned in this chapter are not intended to be representative of the vast set of publications about *dynamic systems* and *chaos theory*; it is only a guide where the reader can consult to establish or expand the concepts provide by this PhD thesis.

2.1 Chaos and the lifelong pursuit of an agreed definition

Chaos theory has been considered as the third greatest discovery on science after relativity and quantum mechanics in the twentieth century. The term *chaos* has been hailed as a revolution of thoughts and attracting ever increasing attention of many scientists from diverse disciplines including mathematics, statistics, data science, physics, cosmology, computation, engineering, chemistry, biology, medicine, neurology, psychology, economics and many others. It has become a truly multi-disciplinary area of research, even it has captured the imagination of the general public.

It has also found applications in a vast number of research areas such as celestial mechanics, meteorology, fluids, lasers, nonlinear optical systems, solids, particle accelerators, plasmas, chemical reactions, biological systems, epidemiology, population dynamic, financial markets, cryptography and so on; the variety of systems in which chaotic behaviours appear encompasses all of nature at the macroscopic and microscopic levels.

Let us recommend some books and articles (and references therein) for anyone who wants to start getting involved in the field of *dynamic systems* and *chaos theory*, see e.g., Gleick (1987), Stewart (1991), Ruelle (1993), Fernández-Díaz (1994), Lorenz (1995), Alligood et al. (1996), Abarbanel (1996), Escot (2000), Ott (2002), Wiggins (2003), Broer and Takens (2011), Hirsch et al. (2012), Layek (2015), Devaney (2018), Fernández-Díaz (2019).

Particularly in statistics there is an increasing number of publications who are interested on chaos, see e.g., Cox and Smith (1953), Eckmann and Ruelle (1985), Bartlett (1990), Berliner (1992), Tong (1990, 1992, 1995, 1997), Giannerini and Rosa (2001), Chan and Tong (2001), Giannerini (2002), Fan and Yao (2003), Shintani and Linton (2004), Giannerini and Rosa (2004), Giannerini et al. (2007), Robinson (2012), Park and Whang (2012), Lasota and Mackey (2013), Ispolatov et al. (2015), Furquim et al. (2016), Lu et al. (2018), Sanjuán (2020).

The word chaos has a very long history and can already be found in the Ancient Greece as $\chi\alpha o\zeta$ whose meaning would be properly *hole*, although with a different meaning that the one we have currently. However the seeds to develop the modern notion of chaos can also be found there, in Aristotle, who was already aware of something like what we call *initial-value sensitivity*. Writing on his book 'On the Heavens' (350 BC) about methodology and epistemology he observed (translated into English): *« the least initial deviation from the truth is multiplied later a thousandfold. »*

Nowadays the twenty-third edition of the Royal Spanish Academy Dictionary (latest edition) provides an ambiguous definition about the word chaos which has the following meanings: «

- *Estado amorfo e indefinido que se supone anterior a la ordenación del cosmos.*
- *Confusión, desorden.*
- *Comportamiento aparentemente errático e impredecible de algunos sistemas dinámicos deterministas con gran sensibilidad a las condiciones iniciales.* »

There is no doubt that the meaning of chaos is misleading and may even seem contradictory, especially if one accepts the term *determinism* with the meaning and scope usually attributed to Pierre-Simon Laplace in his book 'Essai philosophique sur les probabilités' (1814). He wrote:

« Une intelligence qui, à un instant donné, connaîtrait toutes les forces dont la nature est animée et la situation respective des êtres qui la composent, si d'ailleurs elle était suffisamment vaste pour soumettre ces données à l'analyse, embrasserait dans la même formule les mouvements des plus grands corps de l'univers et ceux du plus léger atome; rien ne serait incertain pour elle, et l'avenir, comme le passé, serait présent à ses yeux. »

Two notions of unpredictability could be derived in principle from the ideas proposed by Laplace linked, respectively, to the degree of reliability of the measurements and to the initial-value sensitivity phenomena. First, regarding the degree of reliability of the measurements, the counterpoint seems to have put it the appearance of quantum mechanics and its consequences on the understanding of determinism. The *uncertainty principle* published by Werner Heisenberg on his paper 'Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik' (1927) could be a clear example of this because it would demonstrate the impossibility of applying deterministic equations to the microscopic world, since it would be impossible to know simultaneously with total determination the value of two conjugated variables of a particle, such as the position and velocity. That is, a precise measurement of one of them implies an indetermination in the value of the other. Then related to the second notion of unpredictability linked with the idea of initial-value sensitivity was Max Born who considered it in his book 'Natural Philosophy of Cause and Chance' (1949). He wrote:

« Determinism traditionally related to classical mechanics was not real for certain physical phenomena, given that it is not possible to know with infinite precision the initial conditions, only, we could obtain them with the maximum possible precision. »

Hence these two notions of unpredictability would lead one to think of the fact that a system whose behaviour was chaotic could be structurally deterministic but behave as if it were not, which could explain, in part, the apparent contradiction of the term chaos given by the Royal Spanish Academy Dictionary. The perception of unpredictability apparently inherent in a chaotic behaviour had been conceived in the publications made by numerous researchers so far e.g., H. Poincaré, G.D. Birkhoff, A.M. Lyapunov, R.E. von Mises, A. Kolmogorov, S. Smale, E.N. Lorenz, J. Yorke, J-P. Eckmann, D. Ruelle, F. Takens, H. Tong, K-S. Chan, A. Fernández Díaz, M.A. Sanjuán and co-researchers are noteworthy at the early stage of developments in chaos theory.

Even there is order in unpredictable motions how order and chaos could coexist. What are the laws underlying in chaotic motions? Professor Andrés Fernández Díaz in his book 'Economía dinámica caótica' (1994) provides a complementary point of view when dealing with the epistemological root of the term chaos, proposing to replace the classical dual approach with one of scale. He wrote:

« Insistiendo en la necesidad de buscar un compromiso razonable entre determinismo e indeterminismo, podría establecerse una escala o dosificación que iría desde un determinismo estricto hasta el indeterminismo sostenido por K. Pooper, tal que:

- *Determinismo controlable o básico (Laplace)*
- *Determinismo no controlable (Caos)*
- *Indeterminismo subjetivo (¿Caos?)*
- *Indeterminismo semi-objetivo (Física)*
- *Indeterminismo objetivo (Pooper)*

El caos como paradigma estaría situado entre los escalones b y c, y si se nos permite una analogía con el campo del análisis algebraico, equivaldría a la cuasi-identificación del valor superior y del valor inferior de una cortadura o dicho de otro modo, al elemento α de separación de infinitas sucesiones monótonas convergentes, tal y como se plantea en los postulados de continuidad de Dedekind y Cantor. »

The implications of the time scale really constitute the basis of his argument when he comes to define or delimit the scope of a chaotic behaviour. Therefore, such an approach could be useful for a better understanding of the term chaos. If it is considered as an intermediate step between determinism and indeterminism and the possibility of the simultaneous existence of both in a dynamic system is also considered, this would allow us to conjecture in a relation of strict complementarity between both concepts so that the study of a set of points in the state space with *different degrees* (or *probabilities*) to show a chaotic behaviour could be considered.

Over the last years there has been noticeable move toward statistical work on real-world problems and reaching out by statisticians toward collaborative work with other disciplines. Particularly, the fruitful relation between chaos theory and statistics can be supported by the following statements written by Prof. Howell Tong in his book 'Non-linear time series: a dynamical system approach' (1990). He wrote:

« Many of the ideas surrounding chaos have direct and sometimes quite profound contributions to statistics. The statisticians have an important role to play in clarifying and deepening the understanding of the notion of chaos in a noisy environment. The statisticians have much to offer in real data analysis with a view to extracting chaotic signals in noisy data. »

In this sense it is well-known that a chaotic dynamic system should displays the initial-value sensitivity property. However, ambiguity arises when applying this criterion to real-world observed time-series data which are usually noise-contaminated signals. So a challenging question in chaos theory from a statistical perspective would be how to distinguish chaotic motions from purely random fluctuations in experimental and observational dataset. In this sense Yao and Tong (1994a,b) also point out that it would be interesting quantify the amount of permissible noise without smearing the qualitative characteristics of the deterministic skeleton of the dynamic system.

Now staying on the storyline of this section let us discuss briefly some ideas regarding the pursuit of an agreed definition of chaos. An *ideal definition* should have at least the following five features: (*i*) it is simple and easy to apply by researchers in the field; (*ii*) it contains the essential features of the phenomenon defined; (*iii*) it is possible to derive all important and widely recognised features of the phenomenon from the definition; (*iv*) it includes all widely recognized examples of the phenomenon; (*v*) it excludes anything that is considered as not being an example of the phenomenon.

Generally speaking, there are several definitions of chaos that are commonly in use. For instance, a dynamic system would be *chaotic* if satisfied some of the following properties:

- It has a three-periodic limit cycle as a possible solution (Li and Yorke, 1975).
- It has positive topological entropy (Katok, 1980).
- The power spectral density of related time-series has a component which is absolutely continuous with respect to Lesbegue measure (Bergé et al., 1984).
- Its sequences have positive algorithmic complexity (Ford, 1986).
- It has positive Kolmogorov entropy (Shuster, 1988).
- It has a dense set of periodic orbits, is topologically transitive, and has sensitive dependence on initial conditions (Devaney, 1989).
- It has a strange attractor (Ruelle, 1993).
- It has a rapid loss of information over time and a rapid loss of correlation between the future and past (Brown and Chua, 1996).

As far as we know all definitions of chaos suffer from some defects but the most serious is that some definitions cannot be derived from each other. Note that it is almost impossible to give a precise definition of chaos which at the same time encapsulates all that the term implies in the diverse literature as we have just seen.

Philosophers and scientists are trying to understand logically how unpredictability occurs and how it can be expressed in mathematical setup. The mathematical (and statistical) foundation of chaos theory are still not well established. As far as this PhD thesis is concerned we must point out that in *deterministic* dynamic systems is not possible to give a rigorous one that account for all the aspects that the term implies so far. For *stochastic* processes this task is even harder because of dimensionality. So we have adopted an operational approach in the next sections. Let us focus on the deterministic case postponing for future work the purely stochastic case.

2.2 Deterministic dynamic systems and the stability of its solutions

Generally speaking dynamics refers to the mathematical study of motions and the forces that generate it. Dynamic analysis attempts to determine how systems change, why those changes occur, and how they can be controlled or modified. Dynamic systems are constituted by a set of elements, whose state in each instant of time is characterised by the value taken by a set of variables called *state variables*, among which there are mathematical relations called *laws* or *equations of evolution* (or *motion*). Hence a dynamic system will be determined by specifying a rule that governs the evolution of the process through *time* and the *space* in which the process takes values. Note that the fundamental idea is that the change of each state depends on the state that the dynamic system takes at some previous instant.

As far as this PhD thesis is concerned we will exclusively consider finite-dimensional dynamic systems. These finite-dimensional dynamic systems can be classified according to different non-exclusive criteria, such as systems in discrete-time ($t \in \mathbb{Z}^+$) or continuous-time ($t \in \mathbb{R}$); linear (the functional form of all the equations that characterise the dynamic system are linear) or nonlinear systems (otherwise); conservative (the volume of a given set is preserved in the finite-dimensional phase space under the action of the system over time) or dissipative systems (shrinks or reduces asymptotically to a compact set. That is, it converges over time towards attractor sets); deterministic (the evolution over time of each state variables is perfectly determined by the dynamic system) or stochastic systems (its temporal evolution is affected by at least one random state variable).

We have restricted our discussion in this PhD thesis mainly to discrete-time nonlinear dissipative deterministic dynamic systems for the following four reasons: (*i*) we shall concerned with the statistical analysis of *time-series data* (discrete steps in time); (*ii*) chaos is a phenomenon in nonlinear dynamical systems. It does not exist in linear systems; (*iii*) We are interested on dissipative systems which contracts the state space upon iterating their maps. The dissipative systems solutions converge to attractors excluding those orbits which go off to infinity, orbits will eventually settle around some attractors; (*iv*) ambiguity arises when it comes to (chaotic) stochastic process. So as to aid our discussion let us introduce some preliminaries and key definitions related to discrete-time nonlinear dissipative deterministic dynamic systems and the stability of its solutions. The discrete-time dynamic system may be determined by the solutions of ordinary difference equations (also called *maps*).

Definition 2.1 (Map)**.** A map is to be said a *one-dimensional deterministic difference equation* if satisfied the following equation (*state equation*)

$$X_t = F\left(X_{t-1}\right) \tag{2.2.1}$$

where the *state variable* $X_t \in \mathbb{R}$, $t \in \mathbb{Z}^+$ and $F : \mathbb{R} \to \mathbb{R}$. There are no difficulties that need to be addressed concerning the existence, uniqueness and continuity of solutions from equation (2.2.1). We may generalise this one-dimensional deterministic difference equation to higher dimensions $\mathbb{R}^d$ $(d > 1)$ by writing in the vector form as (*state equations*)

$$\boldsymbol{X}_t = \boldsymbol{F}\left(\boldsymbol{X}_{t-1}\right) \tag{2.2.2}$$

where the *state vector* $\boldsymbol{X}_t \in \mathbb{R}^d$, $t \in \mathbb{Z}^+$ and $\boldsymbol{F} : \mathbb{R}^d \to \mathbb{R}^d$.

Once we have represented the evolution of a phenomenon through a dynamic system, we have to solve it as in the case of static models. However the solution of the dynamic systems will not be a single value for each state variable, instead of that it will be an *orbit* (also called *trajectory*) defined by each state variable. That is, a set of values for each state variable and instant in time which will depend on the state equations and the initial conditions of the system. Thus, the solution of a dynamic system will be given by those trajectories for the state variables whose temporary evolution is determined by the equations that define the system. Generally speaking the solution of a dynamic system is usually defined as a function of time as follows.

Definition 2.2 (Dynamic system solution)**.** The *solution* of a discrete-time dynamic system is the functional $\boldsymbol{\Phi} : \mathbb{N} \to D \subset \mathbb{R}^d$ which assigns a system status for each instant k, $\boldsymbol{\Phi}\left(k\right) = \boldsymbol{F}\left[\boldsymbol{\Phi}\left(k-1\right)\right]; \forall k \in \mathbb{N}$. This solution will not be unique. To guarantee the existence and uniqueness of the solution we have to introduce the concept of *initial condition*. If for the instant $k_0 = 0$ is verified that $\boldsymbol{\Phi}\left(0\right) = \boldsymbol{X}_0$, then the solution $\boldsymbol{\Phi}\left(k\right)$ is said to satisfy the initial condition $\boldsymbol{X}_0$. Given an arbitrary point $\boldsymbol{X}_0$ that belongs to the domain of the system D, there is only one solution of the system which satisfies the initial condition $\boldsymbol{\Phi}\left(k; \boldsymbol{X}_0\right)$. Similarly the solution of a continuous-time dynamic system could be defined. Note that in stochastic systems there is no single solution because even if the system starts from the same initial condition, two different realizations can be obtained. In those systems the solution will be a stochastic process and we would have to make use of probabilistic analysis; instead, deterministic dynamic systems verify the criterion of uniqueness in their solutions. This fact will have important consequences in the following chapters.

Definition 2.3 (Orbit)**.** Let $F : \mathbb{R} \to \mathbb{R}$ be an one-dimensional map. The *forward orbit* starting from an *initial state* X_0 is defined by

$$\Phi^{+}(X_0) = \left\{X_0, F(X_0), F^2(X_0), \ldots, F^k(X_0), \ldots\right\}$$

Similarly, the *backward orbit* starting from X_0 is defined as

$$\Phi^{-}(X_0) = \left\{X_0, F^{-1}(X_0), F^{-2}(X_0), \ldots, F^{-k}(X_0), \ldots\right\}$$

The backward orbit exists if F is a homeomorphism (continuous and has continuous inverse). In general, the *orbit* (or *trajectory*) starting from X_0 under a homeomorphism F is defined as follows:

$$\begin{aligned}\Phi(X_0) &= \left\{F^k(X_0)\right\}_{k=-\infty}^{\infty} \\ &= \left\{\ldots, F^{-k}(X_0), \ldots, F^{-1}(X_0), X_0, F(X_0), \ldots, F^k(X_0), \ldots\right\}\end{aligned}$$

Similarly orbits of higher dimensional maps can be obtained in $\mathbb{R}^d$. Here the notation F^k represents the composition of F with itself k times, neither the k-th power of F nor its k-th order derivative. As we pointed out earlier we have focused on the so-called *dissipative* systems, which contracts the state space upon iterating their maps. On excluding those orbits which go off to infinity, orbits will eventually settle around some *attractors*. Hence attractors are key features for understanding the qualitative behaviour of a dynamic system and its solutions.

Definition 2.4 (Attractor)**.** We say that a compact set $A \subseteq \mathbb{R}$ is an *attractor* for the map F if it satisfies the following properties:

- The set A is an invariant set under the action of the dynamic system. That is, once the system reaches the attractor it remains in it indefinitely, unless an exogenous perturbation pushes it away.
- The initial points starting close to the set A have trajectories that stay close to A and tend asymptotically to it.
- The set A has a basin of attraction.

Definition 2.5 (Basin of attraction)**.** Let A be an attractor for the map F. The basin of an attractor A is the set of points in the state space such that initial conditions chosen in the subset $Y \not\subset A$ dynamically evolve to a particular attractor A of the map F. It can be defined as follows.

$$B = \left\{X : \lim_{k\to\infty} \inf_{Y \not\subset A} \left|F^k(X) - Y\right| = 0\right\}$$

The set B is called the *basin of attraction* or *domain of influence* of an attractor A. Here $|\cdot|$ denotes the Euclidean norm. Note that the unstable

attractors are the ones that have as their basins of attraction themselves. The globally stable attractors are those that have as their basins of attraction the domain D and the locally stable attractors are those that have as their basins of attraction only a subset of it. Such definitions in higher dimensional maps can be obtained similarly in $\mathbb{R}^d$ $(d > 1)$.

Hence an attracting set will be, in essence, an invariant set under the action of the dynamic system, non reducible and with a basin of attraction. That is, the subset of the state space to which the trajectories of the system converge after a transient time and it cannot be decomposed in other invariant disjunct subsets. An attractor may be classified mainly as a *fixed point*, a *periodic point*, a *limit point*, a *limit cycle*, a *quasi-periodic limit cycle*, a *limit torus* or even a more complicated set with a fractal structure (it has non-integer dimension) known as a *strange attractor*. Note that for d-dimensional deterministic systems with dimension less than 3 it is only possible to find any kind of attractor (even strange attractors) in discrete-time systems e.g., Logistic map, Gauss map or Hénon map, never in continuous-time systems as we will explain later.

Now let us focus on the study of stability which plays a crucial role in the dynamics behind the system. In fact, the stability of the dynamic system solutions is an important qualitative property in both linear and nonlinear systems. There are many stability theories in the literature but we will focus on the stability in the *Lyapunov sense*. We will focus on *local* definitions which means the behaviour of a system *inside* an attractor once the dynamical systems, i.e., the orbits, have reached some of the dynamic equilibria mentioned above. Therefore, stability will be determined by the *derivative rule* and the smaller magnitude of the derivative gives the faster rate or convergence will exist. Let us begin with the stability of a fixed point. For simplicity, we will consider the one-dimensional case. Such definitions in higher dimensional maps can be obtained similarly in $\mathbb{R}^d$ $(d > 1)$.

Definition 2.6 (Fixed point)**.** A point p is said to be a *fixed point* of a map $F : \mathbb{R} \to \mathbb{R}$ if $F(p) = p$. That is, if p is invariant under F. We denote it by $\bar{p}$. The fixed points of a map F can be obtained by finding the roots of the equation $F(X) - X = 0$. Let $\bar{p}$ be a fixed point of a map F, then $F(\bar{p}) = \bar{p}$. This yields $F^2(\bar{p}) = F(F(\bar{p})) = F(\bar{p}) = \bar{p}$. Similarly, $F^3(\bar{p}) = \bar{p}, \ldots, F^k(\bar{p}) = \bar{p}$ for all $k \in \mathbb{N}$. Thus the orbit of F starting from the fixed point $\bar{p}$ remains at $\bar{p}$ for all iterations. Hence the fixed point is a constant solution of the map F. This is why the fixed point is also called an *equilibrium point*. Note that this is the only possible attractor in linear systems.

Example 2.1. Let $F : \mathbb{R} \to \mathbb{R}$ be a map defined as $F(X) = 4X(1 - X)$ given $X = \bar{p}$. It has two fixed points $\bar{p} = \{0, 3/4\}$. Note that a map may have infinitely many fixed points. For instance, the identity map $I(X) = X$ has

infinitely many fixed points in $\mathbb{R}$. That is, every real number is a fixed point of this map. Consider another map $F(X) = X + \sin(\pi X)$. In this map every integer is a fixed point, and there are no others. The map $F(X) = X + 1$ has no fixed points.

Definition 2.7 (Stability of a fixed point)**.** A fixed point $\bar{p}$ of a map $F : \mathbb{R} \to \mathbb{R}$ is said to be *stable* or *attracting* if there exists a neighbourhood of $\bar{p}$ such that the orbit starting from any point in this neighbourhood tend to $\bar{p}$. That is, there exists a $\epsilon > 0$ such that for any point $p \in N_\epsilon(\bar{p}) = (\bar{p} - \epsilon, \bar{p} + \epsilon)$, the limit

$$\lim_{k \to \infty} \left| F^k(p) - \bar{p} \right| = 0$$

An attracting fixed point is also known as a *sink* under the flow generated by F. On the other hand, a fixed point $\bar{p}$ of F is said to be *unstable* or *repelling* if there exists a $\epsilon > 0$ such that for any point $p \in N_\epsilon(\bar{p}) = (\bar{p} - \epsilon, \bar{p} + \epsilon)$, there exists at least one point such that $F^k(p) \notin N_\epsilon(\bar{p})$ for a positive integer k. A repelling fixed point is also known as a *source* under the flow generated by F.

Theorem 2.1 (Stability theorem)**.** *Let $F : \mathbb{R} \to \mathbb{R}$ be a smooth function and $\bar{p}$ be a fixed point of F. Then, the following properties are satisfied: (i) if $|F'(\bar{p})| < 1$, then $\bar{p}$ is stable or attracting; (ii) if $|F'(\bar{p})| > 1$, then $\bar{p}$ is unstable or repelling; (iii) if $|F'(\bar{p})| = 1$, then we cannot figure out the stability of the equilibrium $\bar{p}$ from just the derivative $F'(\bar{p})$.*

Proof. Let $\bar{p}$ be a fixed point of the map F. Since F is smooth (continuously differentiable), $F'(\bar{p})$ is well-defined and it is given by

$$\left| F'(\bar{p}) \right| = \lim_{p \to \bar{p}} \left| \frac{F(p) - F(\bar{p})}{p - \bar{p}} \right|$$

(i) Suppose that $|F'(\bar{p})| < 1$. Then there exists a point $p' \in \mathbb{R}$ with $0 < p' < 1$ such that $|F'(\bar{p})| < p' < 1$. Thus, there is a $\epsilon > 0$ such that for any point $p \in N_\epsilon(\bar{p})$,

$$\left| \frac{F(p) - F(\bar{p})}{p - \bar{p}} \right| < p'$$

So whenever $|p - \bar{p}| < \epsilon$,

$$|F(p) - F(\bar{p})| < p' |p - \bar{p}|$$

Now,

$$\begin{aligned} \left| F^2(p) - F^2(\bar{p}) \right| &= |F(F(p)) - F(F(\bar{p}))| \\ &< p' |F(p) - F(\bar{p})| \\ &< p'^2 |p - \bar{p}| \end{aligned}$$

Similarly, using induction

$$\left|F^k(p) - F^k(\bar{p})\right| < p'^k |p - \bar{p}| \tag{2.2.3}$$

Since $0 < p' < 1$, $p'^k \to 0$ as $k \to \infty$. Again, as $\bar{p}$ is a fixed point of F implies $F^k(\bar{p}) = \bar{p}$ for all $k \in \mathbb{N}$. Therefore, from the inequality (2.2.3) we get $\left|F^k(p) - \bar{p}\right| \to 0$ as $k \to \infty$, whenever $p \in N_\epsilon(\bar{p}) = (\bar{p} - \epsilon, \bar{p} + \epsilon)$. This proves that $\bar{p}$ is a stable or attracting fixed point of F, see definition (2.17).

(*ii*) Suppose that $|F'(\bar{p})| > 1$. Then there exists a point $p' \in \mathbb{R}$ with $p' > 1$ such that $|F'(\bar{p})| > p' > 1$. Thus, there is a $\epsilon > 0$ such that for any point $p \in N_\epsilon(\bar{p})$,

$$\left|\frac{F(p) - F(\bar{p})}{p - \bar{p}}\right| > p' \Rightarrow |F(p) - F(\bar{p})| > p' |p - \bar{p}|$$

Now,

$$\begin{aligned}\left|F^2(p) - F^2(\bar{p})\right| &= |F(F(p)) - F(F(\bar{p}))| \\ &> p' |F(p) - F(\bar{p})| \\ &> p'^2 |p - \bar{p}|\end{aligned}$$

Similarly, using induction

$$\left|F^k(p) - F^k(\bar{p})\right| > p'^k |p - \bar{p}| \tag{2.2.4}$$

Since $p' > 1$, $p'^k \to \infty$ as $k \to \infty$. Again, as $\bar{p}$ is a fixed point of F implies $F^k(\bar{p}) = \bar{p}$ for all $k \in \mathbb{N}$. Therefore, from the inequality (2.2.4) we get $\left|F^k(p) - \bar{p}\right| \to \infty$ as $k \to \infty$, whenever $p \in N_\epsilon(\bar{p}) = (\bar{p} - \epsilon, \bar{p} + \epsilon)$. This proves that $\bar{p}$ is an unstable or repelling fixed point of F, see definition (2.17). This completes the proof. □

Example 2.2. Let $F : \mathbb{R} \to \mathbb{R}$ be a map defined as $F(X) = X + \sin(X)$. The fixed points of F are $\bar{p} = \{\varpi\pi\}$ for $\varpi = 0, \pm 1, \pm 2, \ldots$ given by $\sin(X) = 0$. Let $F'(X) = 1 + \cos(X)$ be the derivative such that

$$F'(\varpi\pi) = \begin{cases} 2 & \varpi = 2\varpi' \\ 0 & \varpi = 2\varpi' + 1 \end{cases}$$

for all $\varpi' = 0, \pm 1, \pm 2, \ldots$. Therefore, $F'(0) = 2$, $F'(\pi) = 0$, $F'(2\pi) = 2$. This shows that $\bar{p} = \pi$ is an attracting fixed point, while $\bar{p} = \{0, 2\pi\}$ are repelling fixed points. Then, the basin of attraction of π is $\mathrm{B}(\pi) = \{0, 2\pi\}$. Similarly, the basin of attraction of 3π is $\mathrm{B}(3\pi) = \{2\pi, 4\pi\}$. In general, we get $|F'(2\varpi'\pi)| = 2 > 1$ and $|F'((2\varpi' + 1)\pi)| = 0 < 1$. Thus, $2\varpi'\pi$ for all $\varpi' \in \mathbb{Z}$ are repelling fixed points, while $(2\varpi' + 1)\pi$ are attracting fixed points. The basin of attraction of the fixed points $(2\varpi' + 1)\pi$ are $\mathrm{B}((2\varpi' + 1)\pi) = \{2\varpi'\pi, (2\varpi' + 2)\pi\}$ for all $\varpi' \in \mathbb{Z}$.

Definition 2.8 (Periodic point)**.** A point p is said to be a *k-periodic point* of a map $F : \mathbb{R} \to \mathbb{R}$ if $F^k(p) = p$. The least positive integer k for which that relation is satisfied is called the prime period of the point p. Note that a k-periodic point of a map F is a fixed point $\bar{p}$ of the map F^k. But the converse is not true.

Example 2.3. Let $F : \mathbb{R} \to \mathbb{R}$ be a map defined as $F(X) = X^2 - 1$. If we fixed $p = 0$, then $F(p) = F(0) = -1$, $F^2(p) = F(F(0)) = F(-1) = 0$, $F^3(p) = F\left(F^2(0)\right) = F(0) = -1$ and $F^4(p) = F\left(F^3(0)\right) = F(-1) = 0$. So $p = 0$ is a 2-periodic point of the map F. Consider another map $F(X) = X^2$ given $X = p$. In this map $\bar{p} = \{0, 1\}$ are the fixed points of F. If $F^2(X) = F(F(X)) = X^4 \Rightarrow X^4 - X = 0 \Rightarrow X\left(X^3 - 1\right) = 0 \Rightarrow X(X-1)\left(X^2 + X + 1\right) = 0 \Rightarrow \{0, 1\}$ for all $p \in \mathbb{R}$. Therefore, $\bar{p} = \{0, 1\}$ are the fixed points of F^2 as well. But they are not 2-period points of F, since $F(\bar{p_1}) = 0$, $F(\bar{p_2}) = 1$, $F^2(\bar{p_1}) = 0$, $F^2(\bar{p_2}) = 1$. Maps may have infinite number of periodic points. For instance, let $F(X) = -X$ given $X = p$. Here $\bar{p} = 0$ is the only fixed point of F. For all $p \in \mathbb{R}$ we can see that $F^2(p) = F(F()) = F(-p) = -(-p) = p$. This shows that every real number is a fixed point of $F^2(X)$. But $\bar{p} = 0$ is the fixed point of F. Hence every non-zero real number is a 2-periodic point of this map.

Definition 2.9 (Stability of a periodic point)**.** A k-periodic point p of a map $F : \mathbb{R} \to \mathbb{R}$ is said to be *stable* or *attracting* if it is a stable fixed point of the map F^k. That is, if $\left|\left(F^k\right)'(X)\right| < 1$ given $X = p$ by theorem (2.1). On the other hand, a k-periodic point p of a map F is said to be *unstable* or *repelling* if it is an unstable fixed point of the map F^k. That is, if $\left|\left(F^k\right)'(X)\right| > 1$ given $X = p$

Example 2.4. Let $F : \mathbb{R} \to \mathbb{R}$ be a map defined as $F(X) = 4X(1 - X)$. For instance, if we take the point $p = \frac{1}{8}\left(5 + \sqrt{5}\right)$ we get

$$F\left(\frac{1}{8}\left(5+\sqrt{5}\right)\right) = \frac{1}{16}\left(5+\sqrt{5}\right)\left(3-\sqrt{5}\right) = \frac{1}{8}\left(5-\sqrt{5}\right)$$
$$F\left(\frac{1}{8}\left(5-\sqrt{5}\right)\right) = \frac{1}{16}\left(5-\sqrt{5}\right)\left(3+\sqrt{5}\right) = \frac{1}{8}\left(5+\sqrt{5}\right)$$

Again,

$$F^2\left(\frac{1}{8}\left(5+\sqrt{5}\right)\right) = F\left(F\left(\frac{1}{8}\left(5+\sqrt{5}\right)\right)\right) = F\left(\frac{1}{8}\left(5-\sqrt{5}\right)\right)$$
$$= \frac{1}{8}\left(5+\sqrt{5}\right)$$

This shows that the point $p = \left(5 + \sqrt{5}\right)/8$ is a fixed point of the map F^2. Hence it is a 2-periodic point of the given map. Now we shall use the

derivative rule for finding the stability character of this 2-periodic point of the map F. We have

$$F^2(X) = F(F(X)) = F(4X(1-X)) = 4 \cdot 4X(1-X)(1-4X(1-X))$$
$$= 16X - 80X^2 + 128X^3 - 64X^4$$

Since $\left(F^2\right)'(X) = 16 - 160X + 384X^2 - 256X^3$ given $X = p$ we have

$$\left|\left(F^2\right)'\left(\frac{1}{8}\left(5+\sqrt{5}\right)\right)\right| = 16\left(244 + 105\sqrt{5}\right) > 1$$

Therefore, the point $p = \left(5+\sqrt{5}\right)/8$ is a repelling 2-periodic point of this map.

Definition 2.10 (Limit cycle)**.** The set $\{p_1, p_2\}$ is said to be a *limit cycle* (2-periodic) of a map $F : \mathbb{R} \to \mathbb{R}$ if $F^2(p_1) = (F \circ F)(p_1) = F(F(p_1)) = F(p_2) = p_1$ and $F^2(p_2) = (F \circ F)(p_2) = F(F(p_2)) = F(p_1) = p_2$ where $F(p_1) = p_2$, $F(p_2) = p_1$. The points of a limit cycle (2-periodic) give the fixed points of F^2. Similarly, the set $\{p_1, p_2, p_3\}$ is said to be a *limit cycle* (3-periodic) of F if $F(p_1) = p_2$, $F(p_2) = p_3$, $F(p_3) = p_1$. Then, $F^2(p_1) = p_3$, $F^2(p_2) = p_1$, $F^2(p_3) = p_2$, but $F^3(p_1) = F\left(F^2(p_1)\right) = F(p_3) = p_1$, $F^3(p_2) = p_2$ and $F^3(p_3) = p_3$. The points of a limit cycle (3-periodic) give the fixed points of F^3. In general, a set $\{p_1, p_2, \ldots, p_k\}$ of a map F is said to be a *limit cycle* (k-periodic) if $F(p_1) = p_2, F(p_2) = p_3, \ldots, F(p_{k-1}) = p_k, F(p_k) = p_1$. Then, $F^k(p_1) = p_1, F^k(p_2) = p_2, \ldots, F^k(p_k) = p_k$. The points of the k-cycle are all fixed points of F^k. Note that every point in a cycle is different from each other. An 1-periodic limit cycle is also called a *limit point*.

Example 2.5. Let $F : \mathbb{R} \to \mathbb{R}$ be a map defined as $F(X) = 1 - X^2$. The fixed points of F are $\bar{p_1} = \left(-1+\sqrt{5}\right)/2$ and $\bar{p_2} = \left(-1-\sqrt{5}\right)/2$ given by $F(X) = 1 - X^2 \Rightarrow X^2 + X - 1 = 0 \Rightarrow \left\{\left(-1 \pm \sqrt{5}\right)/2\right\}$. Note that the point $\bar{p_2}$ lies outside the domain of F, $[-1, 1]$. Now, if $F^2(X) = F(F(X)) = F\left(1 - X^2\right) = 1 - \left(1 - X^2\right)^2 = 2X^2 - X^4$. Then, we have $X^4 - 2X^2 + X = 0 \Rightarrow X(X-1)\left(X^2 + X - 1\right) = 0 \Rightarrow \left\{0, 1, \left(-1 \pm \sqrt{5}\right)/2\right\}$. So the fixed points of $F^2(X)$ are $\bar{p} = \left\{0, 1, \left(-1-\sqrt{5}\right)/2\right\}$. But the points $p = \left(-1 \pm \sqrt{5}\right)/2$ are the fixed points of F, so we focus on the fixed points $\bar{p} = \{0, 1\}$. Then, $F(\bar{p_1}) = 1$, $F(\bar{p_2}) = 0$, $F^2(\bar{p_1}) = 0$ and $F^2(\bar{p_2}) = 1$. This shows that the set $\{0, 1\}$ forms the limit cycle of the map F (2-periodic).

Definition 2.11 (Stability of a limit cycle)**.** Let $\{p_1, p_2, \ldots, p_k\}$ be a limit cycle (k-periodic) of a map $F : \mathbb{R} \to \mathbb{R}$. From definition 2.10 each point p_i for $i = 1, 2, \ldots, k$, $k \in \mathbb{N}$, is a fixed point of the map F^k. The limit cycle (k-periodic) is stable (respectively unstable) if and only if the points p_i are

stable (respectively unstable) fixed points of the map $F^k(X_i)$ given $X_i = p_i$. By the chain rule we get

$$\begin{aligned}\left(F^k\right)'(X_i) &= \left(F^{k-1}\right)'(F(X_i)) \cdot F'(X_i) = \left(F^{k-1}\right)'(X_{i+1}) \cdot F'(X_i) \\ &= \left(F^{k-2}\right)'(X_{i+2}) \cdot F'(X_{i+1}) \cdot F'(X_i) \\ &\vdots \\ &= F'(X_{i+k-1}) \cdot F'(X_{i+k-2}) \cdot \ldots \cdot F'(X_{i+1}) \cdot F'(X_i)\end{aligned}$$

Since $\{p_1, p_2, \ldots, p_k\}$ is a limit cycle (k-periodic) of a map F, then $\left(F^k\right)'(p_i) = F'(p_1) \cdot F'(p_2) \cdot \ldots \cdot F'(p_{k-1}) \cdot F'(p_k)$. Hence if p_i are stable fixed points of $F^k(X_i)$ given $X_i = p_i$, then we get the following definition by theorem (2.1). A limit cycle (k-periodic) of a map F is said to be *stable* or *attracting* if $|F'(X_1) \cdot F'(X_2) \cdot \ldots \cdot F'(X_{k-1}) \cdot F'(X_k)| < 1$ given $X_i = p_i$ for $i = 1, 2, \ldots, k$. On the contrary, a limit cycle (k-periodic) of a map F is said to be *unstable* or *repelling* if $|F'(X_1) \cdot F'(X_2) \cdot \ldots \cdot F'(X_{k-1}) \cdot F'(X_k)| > 1$ given $X_i = p_i$ for $i = 1, 2, \ldots, k$.

Example 2.6. Let $F : \mathbb{R} \to \mathbb{R}$ be a map defined as $F(X) = -X^{1/3}$. For instance, if we take the points $p = \{-1, 1\}$ we have $F(-1) = -(-1)^{1/3} = 1$, $F(1) = -(1)^{1/3} = -1$, $F^2(-1) = F(F(-1)) = F(1) = -1$ and $F^2(1) = F(F(1)) = F(-1) = 1$. Then, the set $\{-1, 1\}$ form a limit cycle. This shows that the points $p = \{-1, 1\}$ are a limit cycle (2-periodic) of the map F. Now we shall use the derivative rule for finding the stability character of this 2-periodic point of the map F. We get

$$F'(X) = -\frac{1}{3}X^{-(2/3)} = -\frac{1}{3X^{(2/3)}}$$

given $X = p$ we have $F'(-1) = -1/3$ and $F'(1) = -1/3$. Hence the limit cycle $\{-1, 1\}$ will be stable if $|F'(-1) \cdot F'(1)| < 1$. Since $|F'(-1) \cdot F'(1)| = \left|\left(-\frac{1}{3}\right) \cdot \left(-\frac{1}{3}\right)\right| = \frac{1}{9} < 1$, then the points $\{-1, 1\}$ are an attracting limit cycle (2-periodic) of this map.

Example 2.7. Let $F : \mathbb{R} \to \mathbb{R}$ be a map defined as $F(X) = \mu X(1 - X)$. We are going to find all 2-periodic limit cycles of this map. Firstly, the fixed points of $F(X)$ are the following:

$$\bar{p} = \left\{0, \left(1 - \frac{1}{\mu}\right)\right\}$$

Clearly, the fixed points of $F^2(X)$ are the following $\{\bar{p_1}, \bar{p_2}, \bar{p_3}, \bar{p_4}\}$:

$$\bar{p} = \left\{0, \left(1 - \frac{1}{\mu}\right), \left(\frac{\mu + 1 \pm \sqrt{(\mu + 1)(\mu - 3)}}{2\mu}\right)\right\}$$

So we focus on $\bar{p_3}$ and $\bar{p_4}$. If $\mu = 4$ we get

$$\{\bar{p_3}, \bar{p_4}\} = \left\{ \left(\frac{4+1 \pm \sqrt{(4+1)(4-3)}}{2 \cdot 4} \right) \right\} = \left\{ \frac{5+\sqrt{5}}{8}, \frac{5-\sqrt{5}}{8} \right\}$$

Now, if $F(X) = 4X(1-X)$ given $X = \bar{p_3}$ we have

$$F(\bar{p_3}) = 4\left(\frac{5+\sqrt{5}}{8}\right)\left(1 - \frac{5+\sqrt{5}}{8}\right) = \frac{5+\sqrt{5}}{2} \cdot \frac{3-\sqrt{5}}{8} = \frac{5-\sqrt{5}}{8}$$

Thus we get $F(\bar{p_3}) = \bar{p_4}$, $F(\bar{p_4}) = \bar{p_3}$, $F^2(\bar{p_3}) = \bar{p_3}$ and $F^2(\bar{p_4}) = \bar{p_4}$. The 2-periodic limit cycles of F are the set $\{\bar{p_3}, \bar{p_4}\}$. Here $F'(X) = 4 - 8X$ given $X = \{\bar{p_3}, \bar{p_4}\}$ we have

$$F'(\{\bar{p_3}, \bar{p_4}\}) = 4 - 8\left(\frac{5 \pm \sqrt{5}}{8}\right) = \left\{ -\left(1+\sqrt{5}\right), -\left(1-\sqrt{5}\right) \right\}$$

Hence the limit cycle $\{\bar{p_3}, \bar{p_4}\}$ will be stable if $|F'(\bar{p_3}) \cdot F'(\bar{p_4})| < 1$ by definition (2.19). Since $|F'(\bar{p_3}) \cdot F'(\bar{p_4})| = \left|\left(1+\sqrt{5}\right) \cdot \left(1-\sqrt{5}\right)\right| = 4 > 1$, then the points $\left\{\left(5+\sqrt{5}\right)/8, \left(5-\sqrt{5}\right)/8\right\}$ are a 2-periodic repelling limit cycle of this map with $\mu = 4$.

Definition 2.12 (Quasi-periodic limit cycle)**.** In this case, it appears convenient to concentrate our attention on the following feature. There may be more than one frequency in the periodic trajectory of the system through the state of a limit cycle. If two of these frequencies form an irrational fraction (i.e., they are incommensurate), the trajectory is no longer closed, and the limit cycle becomes a *quasi-periodic limit cycle*. Something similar would be the case of a *limit torus*. An attractor is said to be a *limit torus* if satisfies the following properties: (i) it is composed by two or more overlapping limit cycles (the state space dimension will be $d \geqslant 2$); (ii) each limit cycle must be independent of the others and periodic; see e.g., Lorenz (1995) or Alligood et al. (1996). Note that the quasi-periodic process is an example of a process which is aperiodic but not chaotic.

To sum up if a globally bounded discrete-time nonlinear deterministic system is locally unstable, then there is the *possibility* that no matter how close two initial values are to each other, they will lead to drastically different orbits or trajectories. In this case, the ultimate state cannot be a fixed point, a periodic point, a limit point, a limit cycle, a quasi-periodic limit cycle or a limit torus. Instead, each realization is then almost indistinguishable from that of a stochastic process. Thus, *randomness can be generated by a strictly deterministic system.* This attracting set called *strange attractor* can be described as a feature of *chaos* which trajectories inside it may stay aperiodic for ever. Let us discuss briefly a definition of this kind of interesting attractor.

Definition 2.13 (Strange attractor)**.** It was Ruelle and Takens (1971) who first used this term to denote a finite region of the state space by an infinite number of points where an orbit or trajectory inside it will show an aperiodic cycle. That is, there is no regular finite period (tends to infinity) and it is highly unstable. As far as we know there is not a mathematical definition of a strange attractor and it is probably premature to give one at least as far as this PhD thesis is concerned. Instead, it appears convenient and relevant to concentrate our attention on the following features that are motivated by chaos theory. An attractor is said to be *strange* (or chaotic) if satisfies the following properties: (*i*) it has a fractal structure (it has non-integer dimension); (*ii*) it shows the initial-value sensitivity property. For instance, the Hènon map, the Rössler system or the Lorenz system show a strange attractor for certain parameter values of those systems as we will show later. In order to study the stability of an orbit or trajectory inside this kind of attractor we refer to the quantitative measure called *Lyapunov exponent* which we are going to explain in the next section.

2.3 Lyapunov exponent: measuring the initial-value sensitivity

As we have said at the beginning one of the main features of chaos is the well-known *initial-value sensitivity* property. Several authors have recognised this feature as the primary source of randomness as well, even in stochastic systems. The study of measures to quantify the initial-value sensitive property began with the introduction of Lyapunov exponent, a concept due to Oseledec (1968) and his *multiplicative ergodic theorem*. So quantifying chaos through this kind of quantitative measure is a key point for understanding chaos. In this section we will discuss this issue considering the deterministic dynamic systems.

Traditionally the study of a phenomena with a complex and irregular evolution has been carried out assuming that the underlying dynamic, which generates this complexity, should be represented through stochastic processes. This approach has been propitiated, in part, because the solutions from deterministic systems (perfectly regular, ordered and periodic) were incapable of reproducing the complex dynamics observed in real phenomena. In fact, the maximum degree of complexity that deterministic systems could describe was restricted to quasi-periodic movements. Nowadays, it is generally accepted that deterministic dynamic systems can generate chaos. As we said before it appears that for any definition of chaos, there may always be some chaotic systems which do not fall under some of them. Thus making chaos a twin to *Gödel's undecidability*. So it appears convenient and relevant to concentrate our attention on the following features.

Definition 2.14 (Deterministic chaos)**.** A deterministic dynamical system is said to be *chaotic* if satisfies the following properties: (i) initial-value sensitivity in nearby trajectories; (ii) global boundedness.

It was E.N. Lorenz who first show empirically the notion of initial-value sensitivity in his paper 'Deterministic Non-periodic Flow' (1963). In this paper he derived a nonlinear system for thermal convection in a simplified model of atmospheric flow and noticed a very strange thing that the solutions of the equations could be unpredictable and irregular despite being deterministic. In particular, what he discovered was that if he stopped of iterate the solutions of his dynamic system and did it again starting from a very similar initial condition, the solutions would immediately separate. It is novelly written in his book 'The Essence of Chaos' (1994). The sensitive dependence of the evolution of a system for an infinitesimal change of initial conditions is called popularly the *butterfly effect*. Hence an attribute for a chaotic deterministic system is to exhibit exponentially fast divergence of nearby trajectories for infinitesimally changed initial conditions. The notion of initial-value sensitivity might be expressed mathematically as follows.

Definition 2.15 (Initial-value sensitivity)**.** Let $X_t = F(X_{t-1})$ be an one-dimensional deterministic difference equation defined as equation (2.2.1). The map $F : \mathbb{R} \to \mathbb{R}$ is said to have the *sensitive dependence on initial conditions* property if there exists a $\delta > 0$ such that for any point $p \in \mathbb{R}$, there exists a point $p' \in \mathrm{N}_\epsilon(p) = (p - \epsilon, p + \epsilon)$ such that $\left|F^k(p) - F^k(p')\right| > \delta$ for a positive integer k. We may generalise this definition to higher dimensions $\mathbb{R}^d$ $(d > 1)$ by writing in the vector form. Let $\boldsymbol{F} : \mathbb{R}^d \to \mathbb{R}^d$ displays *initial-value sensitivity* if there exists a $\boldsymbol{\delta} > 0$, $\boldsymbol{\delta} \in \mathbb{R}^d$, such that for any vector $\boldsymbol{p} \in \mathbb{R}^d$, there exists a vector $\boldsymbol{p'} \in \mathbf{N}_\epsilon(\boldsymbol{p}) = (\boldsymbol{p} - \epsilon, \boldsymbol{p} + \epsilon)$ such that $\left|\boldsymbol{F}^k(\boldsymbol{p}) - \boldsymbol{F}^k(\boldsymbol{p'})\right| > \boldsymbol{\delta}$ for an integer $k > 0$.

This relation indicates that for any point $p \in \mathbb{R}$ there are points p' arbitrarily close ϵ which separate from p by at least δ under iterations of the map F. Here $F^k(p)$ denotes the k-th iteration of the map F starting from p. Let us explain this property by considering the following example.

Example 2.8. Let $F : \Theta \to \Theta \subseteq \mathbb{R}$ be a map defined as $F(X) = 2X$ on the unit circle Θ. Let $p_1 \in \Theta$ be a point and $\mathrm{N}_\epsilon(p_1) = (p_1 - \epsilon, p_1 + \epsilon)$ be a ϵ-neighbourhood of the point p_1. Let $\delta > 0$, then there exists an arbitrary point $p_2 \in \mathrm{N}_\epsilon(p_2)$ for $k > 0$ such that $\left|F^k(p_1) - F^k(p_2)\right| = \left|2^k p_1 - 2^k p_2\right| = 2^k |p_1 - p_2|$ where $2^k |p_1 - p_2| > \delta$; This implies that the map F has the sensitive dependence on initial conditions property. It can also be verified numerically by considering two neighbouring points, e.g. $p_1 = 0.25$ and $p_2 = 0.2501$ as shown in Table 2.3.1 (first three columns). This table shows that for increasing k (iterates) the difference $\left|F^k(p_1) - F^k(p_2)\right|$ becomes more and more larger than initial separation 0.0001. Hence the map F satisfies the

k	$F^k(p_1)$	$F^k(p_2)$	$\left\|F^k(p_1) - F^k(p_2)\right\|$	$F^k(p_1)$	$F^k(p_2)$	$\left\|F^k(p_1) - F^k(p_2)\right\|$
0	0.25	0.2501	0.0001	0.45	0.4501	0.0001
1	0.50	0.5002	0.0002	0.90	0.9002	0.0002
2	1.00	1.0004	0.0004	0.80	0.8004	0.0004
3	2.00	2.0008	0.0008	0.60	0.6008	0.0008
4	4.00	4.0016	0.0016	0.20	0.2016	0.0016
5	8.00	8.0032	0.0032	0.40	0.4032	0.0032
6	16.00	16.0064	0.0064	0.80	0.8064	0.0064
7	32.00	32.0128	0.0128	0.60	0.6128	0.0128
8	64.00	64.0256	0.0256	0.20	0.2256	0.0256
9	128.00	128.0512	0.0512	0.40	0.4512	0.0512
10	256.00	256.1024	0.1024	0.80	0.9024	0.1024
11	512.00	512.2048	0.2048	0.60	0.8084	0.2048
12	1024.00	1024.4096	0.4096	0.20	0.6096	0.4096
13	2048.00	2048.8192	0.8192	0.40	0.2192	0.1808
14	4096.00	4097.6384	1.6384	0.80	0.4384	0.3616
15	8192.00	8195.2768	3.2768	0.60	0.8768	0.2768
16	16384.00	16390.5536	6.5536	0.20	0.7536	0.5536
17	32768.00	32781.1072	13.1072	0.40	0.5072	0.1072
18	65536.00	65562.2144	26.2144	0.80	0.0144	0.7856
19	131072.00	131124.4288	52.4288	0.60	0.0288	0.5712

Table 2.3.1: The first 19 iterations provided by the doubling map on the unit circle starting from $p_1 = 0.25$ and $p_2 = 0.2501$ and the real line starting from $p_1 = 0.45$ and $p_2 = 0.4501$.

initial-value sensitive property. Note that this difference increases *linearly* with respect to k. This linearity property does not hold for all maps which dynamics have the initial-value sensitivity property as we can show in the following example.

Example 2.9. Let $F : \mathbb{R} \to \mathbb{R}$ be a map defined as

$$F(X) = 2X \,(\bmod 1) = \begin{cases} 2X, & 0 \leqslant X < 1/2 \\ 2X - 1, & 1/2 \leqslant X < 1 \end{cases}$$

on the real line $\mathbb{R}$. It represents the fractional part of $2X$ for $X \in [0, 1)$. Table 2.3.1 (second three columns) represents the first 19 iterations of F taking $p_1 = 0.45$ and $p_2 = 0.4501$ as two seeds. Clearly, as one can see the map F displays initial-value sensitivity. Note that after 12 iterations the difference $\left|F^k(p_1) - F^k(p_2)\right|$ becomes completely *uncorrelated* with respect to k. This is a feature that would be interesting to explore in the near future. As far as we know deterministic chaos seems to be tied to the notion of rates of divergence of orbits. The reliance on rates seems to open a *pandora's box of rates* e.g., higher, equal or lower than exponential. It would be also a challenging task know if these rates increase always *monotonically* (or not) with respect to iterates.

Remark 2.1. Notice that a map $F : \mathrm{M} \to \mathrm{M} \subseteq \mathbb{R}$ is said to be *expansive* if there exists a $r > 0$ such that for each pair of *every* points $p, p' \in \mathrm{M}$, there exists a positive integer k such that $\left|F^k(p) - F^k(p')\right| > r$. Higher dimensional maps can be similarly defined. The quantity r is called the constant of *expansiveness*. This is an effect in which any initial perturbation is always amplified when iterations are continued. The expansiveness property of a map cannot alone define a system to be chaotic. There are systems which are expansive but not sensitive to initial conditions. Nevertheless, we think the concept of expansiveness is too restrictive at least as far as this PhD thesis is concerned.

The standard notion in deterministic dynamic systems theory which quantifies initial-value sensitivity is the *Lyapunov exponent*. Generally speaking this exponent measures how fast a perturbation in a point moves down the trajectory in a finite number of steps. It can be defined as follows.

Definition 2.16 (Lyapunov exponent)**.** Let $X_t = F(X_{t-1})$ be an one-dimensional deterministic difference equation defined as equation (2.2.1). Let us assume that $F : \mathbb{R} \to \mathbb{R}$ is everywhere differentiable. Let X_0 and X'_0 be two close initial conditions in the state space $\mathbb{R}$ and X_k and X'_k their values after k time steps, respectively. Then after k steps using Taylor's series about X_0 the divergence on the trajectories may be expressed as

$$\begin{aligned}\left(X_k - X'_k\right) &= F^k(X_0) - F^k\left(X'_0\right) \\ &\approx \frac{d}{dX}F^k(X_0)\left(X_0 - X'_0\right)\end{aligned} \tag{2.3.1}$$

Thus, the growth of an infinitesimal perturbation in the initial condition as time evolves is governed by the partial derivatives of the map F. By the chain rule we have

$$\frac{d}{dX}F^k(X_t) = F'(X_0) \cdot F'(X_1) \cdot \ldots \cdot F'(X_{k-1}) \tag{2.3.2}$$

for $t = 0, \ldots, k-1$. If all the quantities of the right-hand side are of comparable magnitude, then $\frac{d}{dX}F^k(X_t)$ increases (or decreases) exponentially with k. That is, the separation $(X_k - X'_k)$ after k time steps behaves likewise. Let $(X_0 - X'_0)$ be the slight perturbation in the initial condition. The Lyapunov exponent gives the average exponential rate of divergence of infinitesimal nearby initial conditions on the attractor. Note that this interpretation coincides exactly with the definition of initial-value sensitivity, so we can use the Lyapunov exponent to quantify the sensitivity property of two neighbouring points. This exponent is governed by

$$\left|X_k - X'_k\right| \approx e^{\lambda k}\left|X_0 - X'_0\right| \tag{2.3.3}$$

It therefore makes sense to consider the average rate of change as follows

$$\begin{aligned}\lambda(X,k) &= \lim_{k\to\infty}\frac{1}{k}\ln\left|\frac{d}{dX}F^k(X_t)\right| \\ &= \lim_{k\to\infty}\frac{1}{k}\left[\ln\left|F'(X_0)\right| + \ln\left|F'(X_1)\right| + \ldots + \ln\left|F'(X_{k-1})\right|\right]\end{aligned} \tag{2.3.4}$$

$\forall F'(\cdot) > 0$. We call $\lambda(X,k)$ the *Lyapunov exponent*. We may generalise this definition to higher dimensions $\mathbb{R}^d$ $(d > 1)$ by writing in the vector form. We simply replace the partial derivatives $F'(X_t)$ by the Jacobian matrix $\boldsymbol{D}_X\boldsymbol{F}$ evaluated along all trajectory, with $\boldsymbol{D}_X\boldsymbol{F} = \frac{\partial F_i}{\partial X_{i,t}}$ where F_i denotes the ith component of $\boldsymbol{F}$, $X_{i,t}$ denotes the ith component of $\boldsymbol{X}$ for $i = 1,\ldots,d$ and $t = 0,\ldots,k-1$. That is,

$$\boldsymbol{D}_X\boldsymbol{F}(\boldsymbol{X_{i,t}}) = \begin{pmatrix} \partial F_1/\partial X_{1,t} & \partial F_1/\partial X_{2,t} & \cdots & \partial F_1/\partial X_{i,t} \\ \partial F_2/\partial X_{1,t} & \partial F_2/\partial X_{2t} & \cdots & \partial F_2/\partial X_{i,t} \\ \vdots & \vdots & & \vdots \\ \partial F_i/\partial X_{1,t} & \partial F_i/\partial X_{2,t} & \cdots & \partial F_i/\partial X_{i,t} \end{pmatrix} \tag{2.3.5}$$

Let $\Gamma_i(\boldsymbol{X})$ denote the *modulus* of the ith eigenvalue of $\boldsymbol{D}_X\boldsymbol{F}$ evaluated along all trajectory. This $i \times i$ matrix can be rewritten as a composition of k Jacobians by applying the chain rule for differentiation. Thus, we define the ith Lyapunov exponent $\lambda_i(\boldsymbol{X},k)$ as follows

$$\begin{aligned}\lambda_i(\boldsymbol{X},k) &= \lim_{k\to\infty}\frac{1}{k}\ln\left|\boldsymbol{D}_X\boldsymbol{F}^k(\boldsymbol{X_{i,t}})\right| \\ &= \lim_{k\to\infty}\frac{1}{k}\left[\ln\left|\boldsymbol{D}_X\boldsymbol{F}(X_{i,0})\right| + \ln\left|\boldsymbol{D}_X\boldsymbol{F}(X_{i,1})\right| + \ldots + \ln\left|\boldsymbol{D}_X\boldsymbol{F}(X_{i,k-1})\right|\right] \\ &= \lim_{k\to\infty}\frac{1}{k}\left[\ln\Gamma_i(X_{i,0}) + \ln\Gamma_i(X_{i,1}) + \ldots + \ln\Gamma_i(X_{i,k-1})\right]\end{aligned} \tag{2.3.6}$$

$\forall\Gamma_i(\cdot) > 0$, $i = 1,\ldots,d$. We call $\lambda_i(\boldsymbol{X},k)$ the *Lyapunov exponent spectrum*. There will be as many exponents as there are dimensions on $\mathbb{R}^d$, $d \geqslant 1$. Note that λ_1 will be called the *largest* Lyapunov exponent. It is the most important of all when it comes to test the initial-value sensitivity hypothesis.

In order to guarantee that the sensitive dependence on initial conditions property (at least one Lyapunov exponent is *positive*) becomes sufficient condition for the existence of chaos, it is necessary that the evolution of the solutions are limited in some region of the state space, that is to say, the orbits go to an attractor (*global boundedness*). This last condition is always verified when the dynamic system is dissipative.

Definition 2.17 (Dissipative system)**.** Let $\boldsymbol{X}_t = \boldsymbol{F}(\boldsymbol{X}_{t-1})$ be a finite-dimensional deterministic difference equation defined as equation (2.2.2). It is said to be *dissipative* if the sum of all the Lyapunov exponents spectrum values is negative ($\sum \lambda_i < 0$).

To sum up we have just explained a quantitative characteristic exponent which measure the initial-value sensitivity property in a deterministic environment. However, this is still some distance from a rigorous definition of chaos for deterministic dynamic systems and we are not aware of any universally agreed definition. As far as this PhD thesis is concerned we have considered an operational approach. A deterministic dynamic system is to be said *chaotic* if satisfied the following properties: (*i*) the dynamic system is dissipative; (*ii*) it has at least one positive Lyapunov exponent. Note that it is notoriously difficult to determine the Lyapunov exponents analytically in most nonlinear deterministic dynamic systems. In practice, the Lyapunov exponents have to be estimated numerically as we will discuss in the next chapter.

Remark 2.2. It is possible to have an *attractor classification criteria* to which the dissipative deterministic systems converge by making use of the Lyapunov exponent spectrum values. The sign of each exponent shows the properties of divergence ($\lambda_i > 0$), convergence ($\lambda_i < 0$) or neither ($\lambda_i = 0$) in the motions on infinitesimally different initial set-ups in each of the d directions of the state space $\mathbb{R}^d$, $i = 1, \ldots, d$. In general, for d-dimensional systems, when all exponents are negative the attractor will be a *fixed point*; when one of them is zero and the rest negative it will be a *limit cycle*; when $d > 2$ if $d - 1$ are null and the rest are negatives it will be a d-dimensional *limit torus*; finally when at least one of the exponents is positive, it will be a *strange attractor*.

For d-dimensional deterministic systems with dimension less than 3 it is only possible to find a chaotic behaviour in discrete-time systems e.g., Logistic map, Gauss map or Hénon map, never in continuous-time systems e.g., Rössler system or Lorenz system. The reason of this fact is the following: at least one Lyapunov exponent must be negative to guarantee that the dynamic system is dissipative (an attractor effectively exists); at least another must be positive (initial-value sensitivity); and, therefore, at least one Lyapunov exponent must be zero (the system is continuous and has some positive exponent). Let us show empirically this condition and discuss briefly some of the dynamic systems already mentioned in the next section.

2.4 Data analysis and simulations

In this section we provide some examples in order to illustrate the particular dynamic behaviour of several strange attractors from five well-known chaotic dynamic systems. We have considered in discrete-time the Logistic map proposed by May (1976), the Gauss map obtained by Gauss (1827) and the Hénon map discovered by Hénon (1976). In addition, we have considered in continuous-time the Rössler system provided by Rössler (1976) and the Lorenz system developed by Lorenz (1963), see table 2.4.1.

Dynamic system	Equations	Parameters	λ_{th}
Logistic	$x_t = \mu x_{t-1}\left(1 - x_{t-1}\right)$	$\mu = 4$	0.69314
Gauss	$x_t = e^{-\alpha x_{t-1}^2} + \beta$	$\alpha = 6.2, \beta = -0.5$	0.38367
Hénon	$x_t = 1 - a x_{t-1}^2 + y_{t-1}$	$a = 1.4$	0.41921
	$y_t = b x_{t-1}$	$b = 0.3$	-1.63479
Rössler	$\dot{x} = -y - z$	$a = 0.2$	0.07143
	$\dot{y} = x + ay$	$b = 0.2$	0.00000
	$\dot{z} = b + (x - c)\, z$	$c = 5.7$	-0.53943
Lorenz	$\dot{x} = \sigma\left(y - x\right)$	$\sigma = 10$	1.37514
	$\dot{y} = -xz + \rho x - y$	$\rho = 28$	0.00000
	$\dot{z} = xy - \beta z$	$\beta = 2.67$	-1.04327

Table 2.4.1: Theoretical Lyapunov exponents values (λ_{th}) from some well-known chaotic dynamic systems.

In order to understand the interrelationship between these strange attractors and its stability we have calculated theoretically the Lyapunov exponents as explained earlier. The data given in table 2.4.1 provide the following comments. First, all dynamic systems considered for certain parameter values show a chaotic behaviour. That is, all verify the condition that they have at least one positive Lyapunov exponent. Second, the number of Lyapunov exponents for each dynamic system is equal to the number of dimensions in the state space respectively. For instance, the Gauss map will have only one Lyapunov exponent value since its state space is $\mathbb{R}$. The Rössler system will have three different Lyapunov exponent values because its state space is $\mathbb{R}^3$. Third, for d-dimensional deterministic systems with a dimension less than 3 it is only possible to find a chaotic behaviour in discrete-time systems e.g., Logistic map, Gauss map or Hènon map, never in continuous-time systems. We can see this condition in the results provided above by the Rössler system and the Lorenz system: one exponent is negative to guarantee that the systems are dissipative, the second exponent is positive verifying the initial-value sensitivity property and the third exponent is zero (the Rössler system and Lorenz systems are continuous-time dynamic systems and both have some positive exponent).

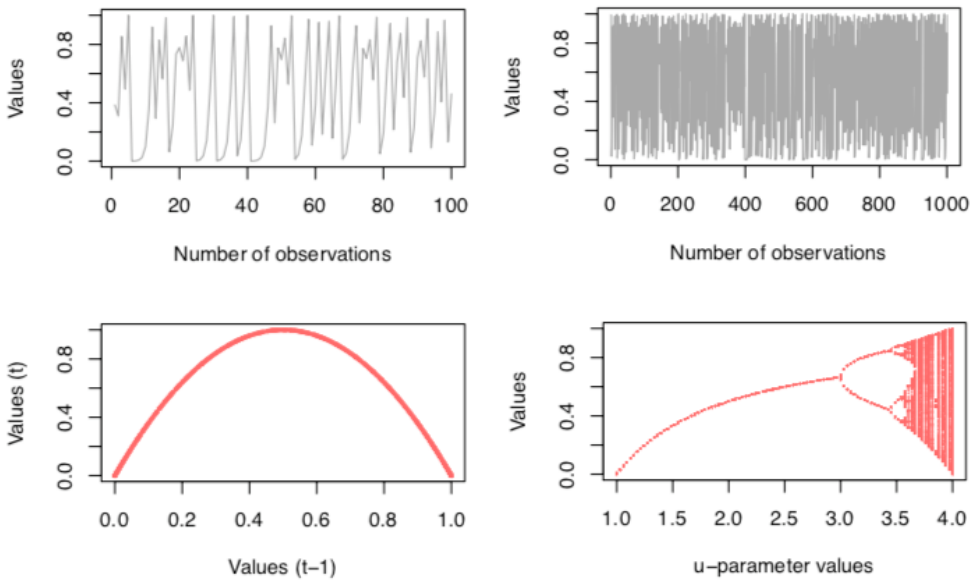


Figure 2.4.1: The evolution of the trajectories from the Logistic map ($\mu = 4$) for an initial period ($t = 100$) and the whole sample ($t = 1000$) are shown above. The Logistic map attractor ($\mu = 4$) and its bifurcation diagram for $1 \leqslant \mu \leqslant 4$ are shown below.

Now let us focus on the qualitative analysis showing a graphical exercise of comparative dynamics from the one-dimensional discrete-time system (Logistic map), the two-dimensional discrete-time system (Hénon map) and the three-dimensional continuous-time systems (Rössler system and Lorenz system). Firstly, we provides some snapshots about the Logistic map and the Hénon map in figure 2.4.1 and 2.4.3 respectively. We have illustrated the evolution of the temporal trajectories for an initial period, the evolution of these for the full sample, the attractor and its projections on the cartesian plane and the bifurcation diagram from a parameter set in which these systems exhibit a chaotic behaviour. We have represented the strange attractor of both maps showing the complex behaviour behind these kind of dynamic systems. In both cases the attractor is neither a *fixed point*, a *periodic point*, a *limit point*, a *limit cycle*, a *quasi-periodic limit cycle* nor a *limit torus*. Instead, there is a finite region of the state space characterised by an infinite number of points where an orbit or trajectory inside it will show an aperiodic cycle (*strange attractor*). That is, there is no regular finite period (tends to infinity) and it is highly unstable as we can see in figure 2.4.2.

The figures showed above allow us to observe empirically the transition from one *steady state* to another by a well-known chaotic dynamic system. Particularly the only fixed point of the Logistic map will be $\bar{p}_1 = 0$ for

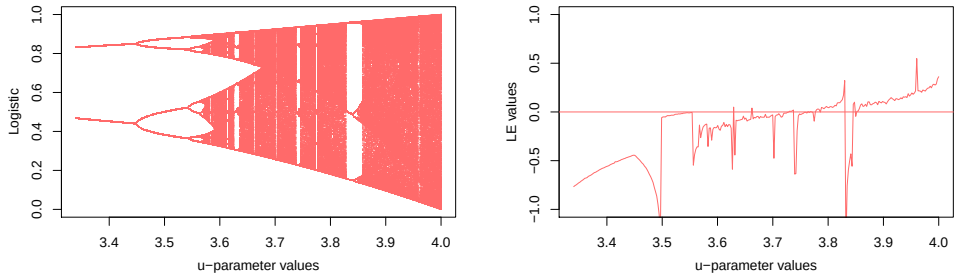


Figure 2.4.2: The bifurcation diagram and the theoretical Lyapunov exponents values from the Logistic map for $3.3 \leqslant \mu \leqslant 4$ are shown.

$0 < \mu < 1$. This steady state constitutes an attractor of the globally stable fixed-point type. That is, from any initial state, the system converges to the fixed point $\bar{p}_1 = 0$. Once this dynamic equilibrium is reached, and in the absence of exogenous disturbances that move it away from it, the system will remain stationary in that fixed point indefinitely. Note that when $1 < \mu < 3$ the fixed point $\bar{p}_1 = 0$ loses its stability becoming a repulsive point. The solutions from the logistic map for this range of the parametric space converge to the attractor (fixed point) $\bar{p}_2 = (\mu - 1)/\mu$. For $\mu > 3$ the two fixed points $\bar{p}_1$ and $\bar{p}_2$ become unstable, showing the solutions limited oscillatory behaviours with different properties regarding their periodicity and local stability or dependence on the initial conditions.

Thus, when $3 < \mu < 1 + \sqrt{6} \approx 3.45$ a new limit cycle type attractor appears, in which the period has doubled with respect to the previous situation, i.e. the new dynamic equilibrium will be a 2-periodic limit cycle, which the system solution will go through sequentially as time passes. This limit cycle will therefore be characterized by two states of the phase space such that

$$\{\bar{p}_3, \bar{p}_4\} = \left\{ \left(\frac{1 + \mu \pm \sqrt{\mu^2 - 2\mu - 3}}{2\mu} \right) \right\}$$

For $\mu > 1 + \sqrt{6} \approx 3.45$ the 2-periodic limit cycle $\{\bar{p}_3, \bar{p}_4\}$ becomes unstable, emerging new qualitative behaviours in the long term. In the first place, a 4-periodic limit cycle will appear, that is, an attractor formed by 4 states that will be repeated sequentially once the attractor has been reached. In particular, the 4-periodic limit cycle reached when $\mu = 3.5$ would be $\{0.8749, 0.3828, 0.8269, 0.5008\}$. This process of instability and appearance of new limit cycles with splitting, bifurcation or doubling of the period continues as we increase the value of parameter μ (period eight, sixteen, twenty-four, etc.). For $\mu > 3.5699$ the *cascade of bifurcations* makes the number of points tend to cover a larger region of state space, we have found the *route*

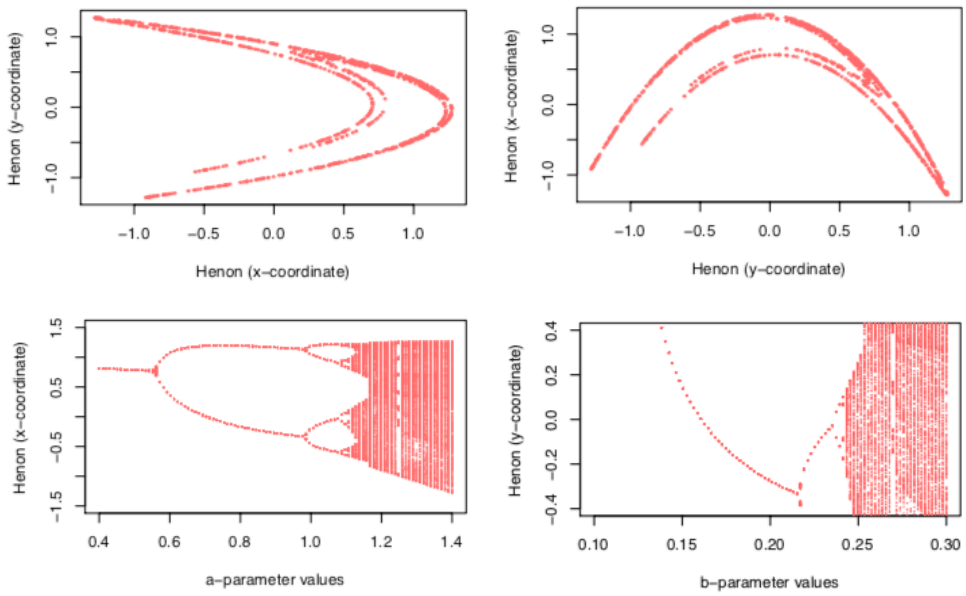


Figure 2.4.3: The Hénon map attractor (x-coordinate; y-coordinate) for $a = 1.4$ and $b = 0.3$ are shown above. The bifurcation diagrams for $0.4 \leqslant a \leqslant 1.4$ and $0.1 \leqslant b \leqslant 0.3$ are shown below.

to chaos, the attractor has become strange and the logistic map behaves chaotically. As one can see in figure 2.4.2 for $\mu > 3.5699$ the cascade of duplications is so rapid that the period tends to be infinite and the number of states to which the system tends to cover a whole region of the state space. In the chaotic region of the bifurcation diagram there are small blank areas, the regularity windows, where the system shows regular periodic behaviour. Some of them are of odd order, which bifurcate or double their period as the values of the parameter continue to increase until they enter again regions with chaotic behaviour.

Let us move on to the second discrete-time dynamic system. In general terms the Hénon attractor has a somewhat more complex structure than the logistic attractor. It is an attractor that fills the space more than a straight but less than a plane, and that presents the same complex appearance when viewed at different scales. Notice that when $b = 0$ the Hénon map is equivalent to the Logistic map. As one can see in the bifurcation diagram the long-term behaviour of the Hénon map depends on the value taken by their parameters. It is observed as in the logistic map that as the value of the parameters increases, there is a cascade of bifurcations with duplication of periods until falling into a chaotic region. Within this region there are regular behavioural gaps. For instance, for the parameter value $a = 1.27$ in the Hénon map there is a regular odd cycle limit (3-periodic) characteristic of

chaotic attractors. Then it is observed that after an increase in the parameter value of a, the cycle doubles until it returns to a chaotic behaviour.

Finally we are going to focus on the continuous-time dynamic systems. We show below some snapshots about the Rössler system and the Lorenz system in figure 2.4.4 and 2.4.5 respectively. Both dynamic system are three dimensional differential equations which we have solved it numerically using the fourth Runge-Kutta method proposed by Runge (1895) and Kutta (1901). We have illustrated the evolution of the temporal trajectories for the full sample ($t = 10000$) and the attractor and its projections on the cartesian plane from a parameter set in which these dynamic systems exhibit a chaotic behaviour. As one can see in these figures, the evolution of the systems within the attractors present a more complex dynamic than the simple dynamic systems as in the discrete-time case.

For instance, the dynamic from the Lorenz attractor draws an orbit with two circular loops in a limited region of the phase space (strange attractor) without the orbit crossing, passing twice through the same point. Although there is a strong recurrence in the global behaviour of the system, the orbit that describes its solution does not present a simple finite periodicity. This aperiodicity is the one that explains the irregularity and complexity that describe the temporal trajectories of the state variables. An irregularity and aperiodicity that resembles the one shown by the purely stochastic processes but in this case coming from a deterministic system. For these parameter values, the system behaves chaotically, presenting a significant dependence on the initial conditions as well as a high irregularity and aperiodicity. Keep in mind that the initial-value sensitivity property implies that two nearby trajectories diverge in their behaviour over time.

Note that for d-dimensional deterministic systems with dimension less than 3 it is only possible to find a chaotic behaviour in discrete-time systems e.g., Logistic map, Gauss map or Hénon map, never in continuous-time systems. Thus in continuous-time systems on $\mathbb{R}$ the Lyapunov exponent will be negative $(-)$, and the only possible attractor will be the *fixed point*. On $\mathbb{R}^2$ it is possible that *fixed points* $(-,-)$ and *limit cycles* $(0,-)$ will appear, see *Poincaré-Bendixson theorem* proposed by Poincaré (1881) and Bendixson (1901)). On $\mathbb{R}^3$ it will be possible to obtain attractors with more complex dynamics e.g., Rössler system or Lorenz system: $(+,0,-)$ *strange attractor*, see table 2.4.1; $(0,0,-)$ two-dimensional *limit torus*; $(0,-,-)$ *limit cycle*; $(-,-,-)$ *fixed point*. If $d > 3$ it will be necessary that at least one Lyapunov exponent be positive, another null and some negative as we explained before. For instance, on $\mathbb{R}^4$ any of the following combinations would guarantee that the system is chaotic: $(+,+,0,-)$; $(+,0,0,-)$; $(+,0,-,-)$.

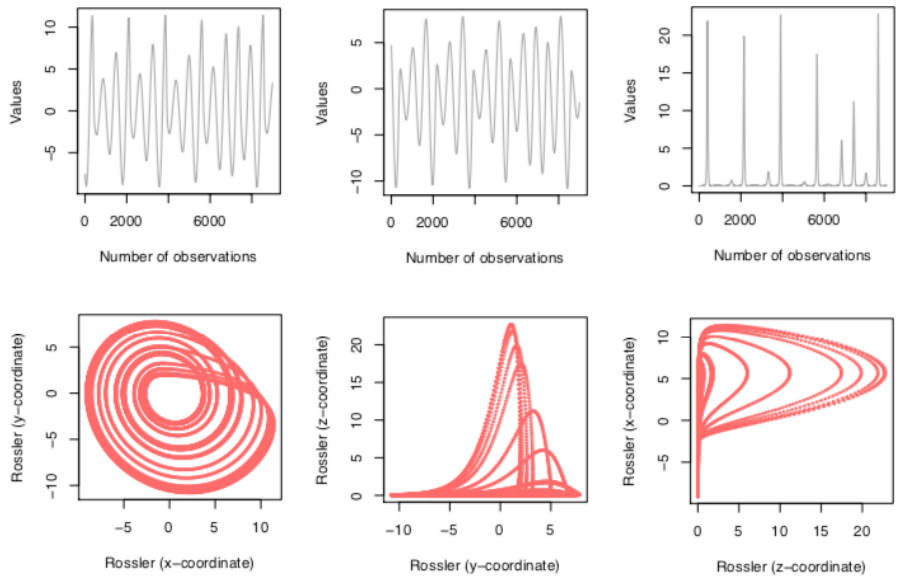


Figure 2.4.4: The evolution of the trajectories from the Rössler system for the whole sample ($t = 10000$) are shown above. The Rössler system attractor (x-coordinate; y-coordinate; z-coordinate) for $a = 0.2$, $b = 0.2$ and $c = 5.7$ are shown below.

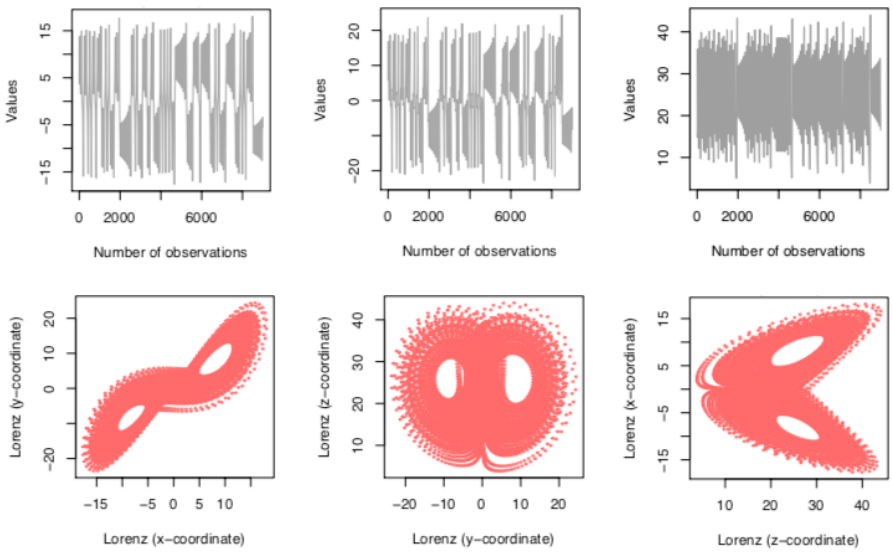


Figure 2.4.5: The evolution of the trajectories from the Lorenz system for the whole sample ($t = 10000$) are shown above. The Lorenz system attractor (x-coordinate; y-coordinate; z-coordinate) for $\sigma = 10$, $\rho = 27$ and $\beta = 8/3$ are shown below.

2.5 Discussion

In this chapter we have presented a summary of the main concepts and definitions already established in the discipline of dynamic systems and chaos theory. We have also provide some of the theoretical results that allow us to establish the existence of chaotic trajectories as solutions to certain finite-dimensional nonlinear dissipative deterministic dynamic systems.

As we have just seen if a globally bounded discrete-time nonlinear deterministic system is locally unstable, then there is the possibility that no matter how close two initial values are to each other, they will lead to drastically different orbits or trajectories. In this case, the ultimate state cannot be a *fixed point*, a *periodic point*, a *limit point*, a *limit cycle*, a *quasi-periodic limit cycle* or a *limit torus*. Instead, there is a finite region of the state space with an infinite number of points where an orbit or trajectory inside it will show an aperiodic cycle. That is, there is no regular finite period (tends to infinity) and it is highly unstable. The existence of this kind of attracting set called *strange attractor* can be described as an evidence of a chaotic behaviour.

In order to study the stability of an orbit or trajectory inside this kind of attractor we refer to the quantitative measure called Lyapunov exponent. When dynamic systems are known, we can directly calculate both the largest and the full spectrum of the theoretical Lyapunov exponent value to check when the dynamic system is in a regime of chaotic behaviour. However in *most real-world observed time series the data-generating process is rarely known a priori*. Then we are also interested in studying what happens when we assume that the true dynamics of the system is unknown. Unfortunately, these theoretical results are not directly applicable when moving into the field of nonlinear time serie analysis. The empirical analysis of chaotic dynamic systems is based on the study of observed time series of statistical data. These time-series data will have been generated by some deterministic (or stochastic) process that in most cases will be unknown.

Hence if we take into account the empirical environment we do not have the advantage of observing directly the state of the dynamic system let alone knowing the functional form that generate the dynamic associated with it. That is, the empirical analysis is precisely based on obtaining or inferring information about the properties of the unknown data-generating process. In the next chapter we want to know if the underlying generator system of the time-series data presents a simple dynamic, behaves chaotically or comes from a purely stochastic process. That is, we will focus on the theoretical framework about how to detect chaos from time-series data instead of well-known dynamic systems as we have seen in this chapter.

Chapter 3

Time-series data, attractor reconstruction and chaos

"Wahrlich es ist nicht das Wissen, sondern das Lernen, nicht Besitzen sondern das Erwerben, nicht das Da-Seyn, sondern das Hinkommen, was den grössten Genuss gewährt".

— Carl Friedrich Gauss

Abstract: This chapter presents a summary of the main concepts and definitions already established in the discipline of nonlinear time series analysis and chaos theory. The aim is to provide a broad conceptual basis for a better understanding of the results presented in subsequent chapters. Section 3.1 gives some preliminaries features for detecting chaos from real-world observed time-series data which are usually noise-contaminated signals. Section 3.2 illustrates the correct procedure for the reconstruction of the state space from time-series data with uniform time-frequency. It is also discussed some criteria in order to estimate the embedding parameters. Section 3.3 provides an overview about the main traditional direct methods for estimating the Lyapunov exponents from time-series data and its limitations. Section 3.4 reports the empirical results of this chapter and a discussion of them. Section 3.5 contains some concluding remarks.

The references mentioned in this chapter are not intended to be representative of the vast set of publications about *nonlinear time series analysis* and *chaos theory*; it is only a guide where the reader can consult to establish or expand the concepts provide by this PhD thesis.

3.1 Observer function and measurement noise

Since it was realised that even simple and deterministic dynamic systems can produce trajectories which look as a random process, there occurred some obvious questions: how can one distinguish chaos from randomness in *real datasets*?; how can we extract from *time-series data* relevant information about the *unknown underlying generator system* of them? Those questions were the starting point of what is nowadays called *nonlinear time series analysis*. First of all, let us recommend some books and articles (and references therein) for anyone who wants to start getting involved in the field of nonlinear time series analysis and chaos theory from a statistical perspective; Chan and Tong (2001), Kantz and Schreiber (2004), Chan (2009), Takens (2010), Teräsvirta et al. (2010), Bradley and Kantz (2015), Tang et al. (2015), Skokos et al. (2016), Huffaker et al. (2017), Tsay and Chen (2019).

In this section we are going to review some of the fundamental concepts which are required for answering those questions. Remember that devising a dynamic system to model some time-evolving phenomena involves specifying a *rule* that governs the evolution of the process through *time* and the *space* in which the process takes values as we mentioned in the previous chapter. If the rule that governs the evolution of the process is well understood it may be possible to write down the dynamic system more or less from first principles, without much need for observational data except perhaps to fix some parameters. Otherwise we may be much less clear about the underlying rules. That is, we would have to deal with the following situations: (i) we might suspect that the system is governed by e.g., a set of difference equations, but not know what these are, even we may be uncertain what dynamic variables the equations involve; (ii) we may have available a quantity of experimental (or observational) data relating to how the system is observed to change through time. Then in that context we will have to assume that the *true data-generating process* is *unknown*. That is, we will consider that all information available is this sequence of measurements in the form of *time-series data*. Let us fix some previous concepts.

Definition 3.1 (Observer function)**.** Let $X_t = F(X_{t-1})$ be an one dimensional deterministic difference equation defined as equation (2.2.1) where $F : \mathbb{R} \to \mathbb{R}$. We have called it the *state equation* of the dynamic system. Now we assume that the true data-generating process is unknown. Hence we do not have the advantage of observing directly the state of the system X_t let alone knowing the functional form F that generate the dynamic associated with it. Instead of that, there is an *observer function* $f : \mathbb{R}^2 \to \mathbb{R}$ that include an *additive measurement error* ε_t which generates observations as (*observation equation*)

$$x_t = f(X_t, \varepsilon_t) \tag{3.1.1}$$

where the *observable variable* $x_t \in \mathbb{R}$, $t \in \mathbb{Z}^+$ (discrete-time) and ε_t is a sequence of independent and identically distributed random variables such that ε_t is independent of X_j, $0 \leqslant j \leqslant t$, for $t = 1, 2, 3, \ldots, n$. We may generalise this one-dimensional observation equation to higher dimensions $\mathbb{R}^d$ $(d > 1)$ by writing in the vector form as

$$\boldsymbol{x}_t = \boldsymbol{f}\left(\boldsymbol{X}_t, \boldsymbol{\varepsilon}_t\right) \tag{3.1.2}$$

where the *obervable vector* $\boldsymbol{x}_t \subseteq \mathbb{R}^d$, $t \in \mathbb{Z}^+$ (discrete-time), $\boldsymbol{f} : \mathbb{R}^{d+1} \to \mathbb{R}^d$ and $\{\boldsymbol{\varepsilon}_t\}$ is a sequence of measurement errors. As far as this PhD thesis is concerned we will consider only *one observable variable*. Hence it is assumed that all information available is the noise-contaminated sequence $\{x_t\}_{t=1}^n$ as a *scalar strictly stationary univariate time serie*.

This setup is proposed as the general form for mathematical models in which dynamic systems generate time-series data in the following sense. The trajectories of the dynamic system are sequences of the form $\left\{X_t = F^t\left(X_0\right)\right\}$, where usually the index t runs through the nonnegative integers. We assume that what is observed, or measured, at each time t is not the whole state X_t, but just one real number $x_t = f\left(X_t, \varepsilon_t\right)$ where f is the observer function which assigns to each state the value that is observed when the system is in that state. So corresponding to each orbit $\left\{X_t = F^t\left(X_0\right)\right\}$ there is a time series of successive measurements $\left\{x_t = f\left(F^t\left(X_0\right), \varepsilon_t\right) = f\left(X_t\right), \varepsilon_t\right\}_{t=1}^n$. We have considered it appropriate to add a measurement noise term in the observer function because most *real-world observed time-series data* are usually *noise-contaminated signals*.

Let us illustrate the effect of adding just a (small) measurement noise to a well-known deterministic dynamic system as the Logistic map with a chaotic behaviour $(\mu = 4)$. We have added to each time-series data a normal multinomial error term denoted by $\varepsilon_t \sim N\left(0, s\right)$ with different variance values s. The data given in figure 3.1.1 provides the following comments. First, as the measurement error increases, the dispersion of the huge amount of points that describe the attractor is amplified with the consequent inaccuracy. So a challenging question would be to quantify the amount of permissible measurement noise without smearing the qualitative characteristics of the latter. Second, the noise amplification is uniform through the dynamics and it does not dynamically interact with the underlying generator system (it is independent on where we are). Third, when $s > 0.1$ the presence of a measurement noise can lead to confusion between a chaotic deterministic system and a purely stochastic one. In such cases the interplay between order and disorder, determinism and stochasticity, stability and instability or predictability and unpredictability is undergoing profound changes with very exciting implications on nonlinear time series analysis and chaos.

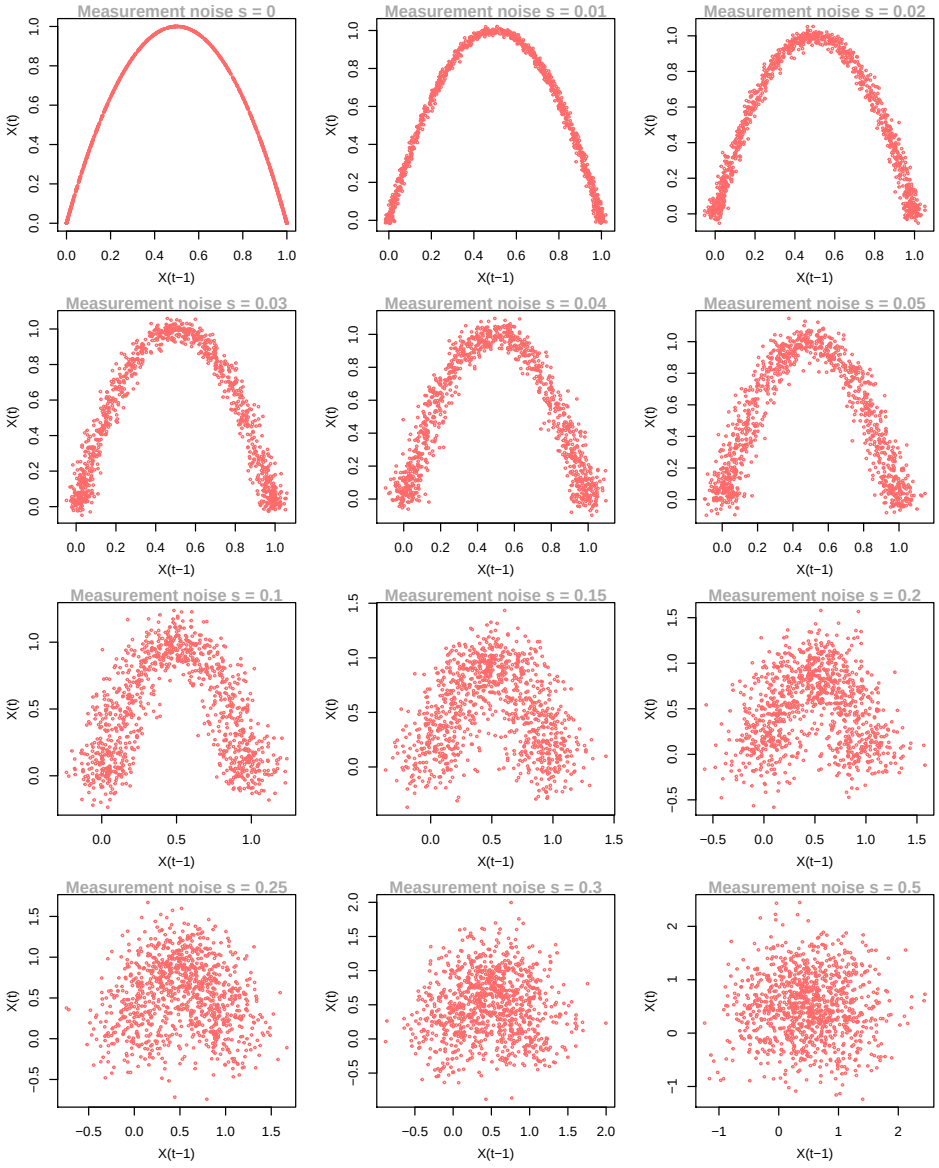


Figure 3.1.1: Logistic map attractor adding a measurement noise with several variance values.

From now on we assume that the *true data-generating process* is unknown. Thus it will not be possible to consider the true orbit of the dynamic system in the original state space. Instead of that, it will be possible to obtain an approximation (reconstruction) of it that result equivalent in a topological sense (equivalence in the dynamic and geometric properties). So this *reconstruction procedure* allows to extract all the relevant information about the unknown underlying dynamical system that generates the time series as the topology of the state space, the number and types of periodic orbits or certain invariant ergodic measures. This is a key point regarding the results that will be shown below. Let us explain it in more detail in the next section.

3.2 Uniform delayed-coordinate embedding procedure

The embedding (or reconstruction) procedure allows us to get all the relevant information about the unknown underlying dynamic system that generates the time-series data (*invariant properties*) e.g., the Lyapunov exponents defined previously must have approximately the same value in both the true and the reconstructed state space. This fact allows us to test the hypothesis of chaos in the unknown original dynamic system. Any method for estimating the Lyapunov exponent from some observed time-series data are based previously on the state space reconstruction procedure. The *embedding theorem* proposed by Takens (1981) provides a framework to reconstruct any unknown dynamic system which gave rise to a given observed scalar time series simply by reconstructing a new state space out of successive values of the time series. This fact enables scientists to construct models for complex and non-linear systems using just *a single observable*. Let us define previously two important concepts; an *embedding* and the *uniform delayed-coordinate embedding vectors*.

Definition 3.2 (Embedding)**.** A smooth map $\psi : \mathcal{M} \to \mathcal{N}$ between two manifolds (topological spaces) $\mathcal{M}$ and $\mathcal{N}$ is called an *embedding* if ψ yields a homeomorphism (ψ is bijective, continuous and its inverse is continuous) between $\mathcal{M}$ and its image $\psi(\mathcal{M})$ where $\psi(\mathcal{M}) \subset \mathcal{N}$. We have considered the *method of delayed-coordinates* proposed by Ruelle and Takens (1971) to get the *uniform delayed-coordinate embedding vectors* as follows.

Definition 3.3 (Uniform delayed-coordinate embedding vector)**.** Let $\{x_t\}_{t=1}^{n}$ be a scalar strictly stationary univariate time serie given by definition 3.1. We form a sequence of *uniform delayed-coordinate embedding vectors* $\boldsymbol{x}_t^m$ by associating for each time period t a vector in a *reconstructed state space* $\mathbb{R}^m$, whose coordinates satisfy the following (*reconstruction equation*)

$$\boldsymbol{x}_t^m = \left(x_t, x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-2)\tau}, x_{t-(m-1)\tau}\right) \tag{3.2.1}$$

where m is the *embedding dimension* and τ is the *reconstruction time-delay* (or *lag*). Despite the fact that we have measured just a single observable, we can construct a vector space whose axes represent all the relevant variables given by

$$\begin{aligned}
\boldsymbol{x}_1^m &= \left(x_1, x_{1-\tau}, x_{1-2\tau}, \ldots, x_{1-(m-2)\tau}, x_{1-(m-1)\tau}\right) \\
\boldsymbol{x}_2^m &= \left(x_2, x_{2-\tau}, x_{2-2\tau}, \ldots, x_{2-(m-2)\tau}, x_{2-(m-1)\tau}\right) \\
&\vdots \\
\boldsymbol{x}_n^m &= \left(x_n, x_{n-\tau}, x_{n-2\tau}, \ldots, x_{n-(m-2)\tau}, x_{n-(m-1)\tau}\right)
\end{aligned}$$

where the number of embedding vectors will be equal to $n-(m-1)\,\tau$ and its length equal to the embedding dimension m. The underlying idea is to make copies of the single observable signal with *uniform-time frequency* ($x_i - x_{i-1} = x_s - x_{s-1}\ \forall i \neq s$) and consider those delayed values as coordinates of a reconstructed state space retrieved from the time-series data. Let us explain the reconstruction procedure by considering the following example.

Example 3.1. Let $\mathrm{z}_t = \{3, 4, 5, 1, 2, 3, 3, 8, 6, 9\}$ be the measurements of a certain variable z. We assume that the measurements are taken every fixed time interval $\Delta = 10$ seconds. Then, the first measure, taken at time $t_1 = 0$ is $z_1 = 3$. After 10 seconds, at $\mathrm{t}_2 = t_1 + \Delta$, the second measure is $z_2 = 4$. The third measure is taken after 20 seconds at $\mathrm{t}_3 = t_1 + 2\Delta$, it is $z_3 = 5$ and so on. A point at time t in the reconstructed state space $\mathbb{R}^m$ is given by a vector $\boldsymbol{z}_t^m$ with the following components $\left(z_t, z_{t-\tau}, z_{t-2\tau}, \ldots, z_{t-(m-1)\tau}\right)$.

In this example we choose arbitrarily the following embedding parameters $m = 3$ and $\tau = 2$. The number of embedding vectors will be $n-(m-1)\,\tau = 10 - (3-1)\cdot 2 = 6$. Thus we have six points in the 3-dimensional reconstructed state space given by

$$\begin{aligned}
z_1 &= (3, 5, 2) \quad z_2 = (4, 1, 3) \quad z_3 = (5, 2, 3) \\
z_4 &= (1, 3, 8) \quad z_5 = (2, 3, 6) \quad z_6 = (3, 8, 9)
\end{aligned}$$

Hence we can write the reconstructed trajectory joining the points $z_1, \ldots, z_6$ where $\boldsymbol{z}_t = (z_t, z_{t-\tau}, z_{t-2\tau})$ for $t = 1, \ldots, 6$, see definition 3.3.

Remark 3.1 (Embedding parameter estimation)**.** A key point to create a suitable reconstruction of the state space is fix a right criteria in order to estimate robustly the *embedding parameters* (τ and m). Researchers in this area usually estimate them using two different alternatives: a *heuristic approach* that mostly rely on physical or geometrical arguments and by a *statistical approach*. Under the heuristic approach regarding the estimation of the time-delay τ although there are other criteria, see e.g.Abarbanel (1996), Kantz and Schreiber (2004), $\tau = 1$ is commonly used following the prescription proposed by Takens (1981).

Concerning the embedding dimension m most of the papers published consider the false nearest neighbours criteria proposed by Kennel et al. (1992). Another criteria widely used by the scientific community is to estimate the correlation dimension as a proxy of the embedding dimension using the algorithm proposed by Grassberger and Procaccia (1983).

The main drawbacks of these heuristic approaches are the following: (*i*) they are not intrinsically statistical; (*ii*) they lead to estimators whose properties are unknown or largely unexplored; (*iii*) they do not take into account the results of any model fit. The alternative proposed by the statistical approach solves those 3 disadvantages. The statistical approach to state space reconstruction can be viewed as a best subset selection problem within the nonparametric regression context as argued Chan and Tong (2001). The idea behind it is to select the embedding parameters, τ and m, that provide the best fit in the estimation of any quantitative measure e.g., the Lyapunov exponents taking into account some information criteria. For instance, the Akaike's information criterion (AIC) proposed by Akaike (1973), the Bayesian information criterion (BIC) provided by Schwarz (1978) or the Focused information criterion (FIC) obtained by Claeskens and Hjort (2003).

There are also model selection methods based on some information criteria that use cross-validation techniques which split the dataset into training, validation and test set. That is, we partition the time-series data into subsets, training the data on a subset and use the other subset to evaluate the model's performance. To reduce variability we should do multiple rounds of cross-validation with different subsets from the same dataset. Then we have to combine the validation results from these multiple rounds to come up with an estimate of the model's fit. There are some cross-validation techniques as e.g., *leave one out cross-validation* (the most recommended cross-validation option in the reconstruction procedure), *k-fold cross validation* or *stratified cross-validation*.

As far this PhD thesis is concerned we are going to use the statistical approach based on model selection procedures taking into account the Bayesian information criterion instead of heuristic techniques, and in some cases we will use leave one out cross-validation techniques. In any case, we think that the information derived from the heuristic approaches might be still useful and should not be disregarded as a complementary information. Keep in mind that the embedding parameters belong to the parameter set to be estimated when selecting the best model by several estimation methods of the Lyapunov exponent as we will show in subsequent chapters. Now, let us introduce some preliminar information in order to understand the reconstruction theorem.

The *reconstruction theorem* was presented in a symposium held at the University of Warwick in 1980. It was obtained independently by Aeyels (1981) and Takens (1981) as point out Broer and Takens (2011). It seems that Aeyels would have gotten the same type of results in his thesis in 1978 which was published later. As far as this PhD thesis is concerned we will follow the framework proposed by Takens which extends to the field of dynamic systems the classic embedding theorem proposed by Whitney (1936) in topology. In this connection, we would like to mention four interesting papers by Tong and Lim (1980), Packard et al. (1980), Mañé (1981) and Sauer et al. (1991).

Let us define the *delay embedding map* denoted by $\Phi_{h,\varphi} : \mathcal{M} \to \mathbb{R}^m$ as follows $\Phi_{h,\varphi}(x) = \left(\varphi(x), \varphi(h(x)), \ldots, \varphi\left(h^{m-2}(x)\right), \varphi\left(h^{m-1}(x)\right)\right)$. Suppose that $\mathcal{M}$ is compact, let $\mathcal{D}^l(\mathcal{M})$ be the set of diffeomorphisms (bijectives, l times continuously differentiables, their inverses are l continuously differentiables) of $\mathcal{M}$, and $\mathcal{C}^l(\mathcal{M})$ be the set of observation functions on $\mathcal{M}$. We assume following Takens (1981) that there is a subset of diffeomorphisms $h \in \mathcal{D}^l(\mathcal{M})$ that have only a finite number of periodic orbits of period less than m, and the eigenvalues of each such periodic orbit are distinct. If $m \geqslant 2d+1$, then there is an open and dense subset of observation functions $\varphi \in \mathcal{C}^l(\mathcal{M})$ for which the delay embedding map $\Phi_{h,\varphi}$ is an *embedding*, see definition 3.2. The reconstruction theorem may be described as follows.

Theorem 3.1 (Reconstruction theorem)**.** *Let $\mathcal{M} \subset \mathbb{R}^d$ be a compact d dimensional manifold. Then if $m \geqslant 2d+1$, the set of (h, φ) for which the delay embedding map $\Phi_{h,\varphi}$ is an embedding, is open and dense in $\mathcal{D}^l(\mathcal{M}) \times \mathcal{C}^l(\mathcal{M})$ for $l \geqslant 1$.*

In other words, let $\{x_t\}_{t=1}^n$ be a scalar strictly stationary univariate time serie given by definition 3.1. We consider the sequence of its m-dimensional reconstruction vectors $\left(x_t, x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-2)\tau}, x_{t-(m-1)\tau}\right)$. These uniform delayed-coordinate embedding vectors are *diffeomorphics* to the evolution of the unknown dynamical system if satisfied the following two conditions: (i) the embedding dimension m is larger than twice the dimension of the *true* state space ($m \geqslant 2d+1$); (ii) the state function F and the observer function f belongs to the open and dense subset defined by theorem 3.1 (are smooth functions).

The sense in which the sequence of reconstruction vectors is diffeomorphic to the underlying evolution of the dynamic system could be seen as follows. The relation between the time series $\{x_t\}_{t=1}^n$ and the underlying generator system $\left\{X_t = F^t(X_0)\right\}_{t=1}^n$ is that $x_t = f(X_t, \varepsilon_t)$, see definition 3.1. This fact means that the reconstruction vectors of the time series are just the images under the delay embedding map of the successive points of the evolution X_t. So this map which is an embedding of $\mathcal{M} \subset \mathbb{R}^d$ into its image (by theorem 3.1), sends the orbit of X_t to the sequence of reconstructions vectors $\left(x_t, x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-2)\tau}, x_{t-(m-1)\tau}\right)$ of the time series $\{x_t\}_{t=1}^n$.

That is, despite the state being hidden from direct observation, the topology of the attractor that characterises the dynamic system can be preserved in the scalar strictly stationary univariate time serie when it is arranged into uniform delayed-coordinate embedding vectors provided that the above conditions are satisfied.

Remark 3.2. The embedding theorems as the reconstruction theorem 3.1 usually take a well-defined class of functions and assert that, for a *large subset* of this class, an embedding is obtained by the method of delays or some other specified procedure. Since the theorems do not cover the *entirety* of the class, they give no assurance that any particular member of the class will produce an embedding. This would have to be verified for the particular case, for a review see Stark (1999), Deyle and Sugihara (2011). Thus these kind of theorems do not prove that applying the method of delays to a particular experimental dataset will result in an embedding. What they do is proving some evidences that the method of delays is an operational approach to successfully obtain the uniform delayed-coordinate embedding vectors. Hence although the conclusions of the reconstruction theorem are strong, its assumptions are hard to verify in general and may in fact fail in a number of theoretical applications.

For instance, the time could have been *continuous*; instead of measuring just a single observable, one could measure several variables (*multivariate time series*); the dynamics could have been given by an *endomorphism* instead of a diffeomorphism; the manifold could be *non-compact*. We refer the interested reader to this research area see e.g., Sauer et al. (1991) who extended it to dynamic systems in higher dimensions, Stark et al. (2003) who proved a number of embedding theorems for stochastic dynamic system or Robinson (2005) who extended it to infinite-dimensional dynamic systems. As far as this PhD thesis is concerned we will considered one-dimensional strictly stationary univariate time series which uniform delayed-coordinate embedding vectors are diffeomorphics to the evolution of its unknown finite-dimensional discrete-time non-linear dissipative deterministic systems.

Note that the reconstruction theorem assumes that the dynamic system is sampled with uniform-time frequency e.g., 1-month, 1-day, 1-hour, 30-min, 5-min, 1-min and so on. However, there has been increasing interest in situations where observations are characterised by its almost continuity. That is, the data does not come from a series of values measured periodically in time. For instance, a motivating example would be the case of financial markets. Nowadays trades realized by traders are replaced by algorithmic trades executed in an automated way providing a huge amount of data over many intra-daily disaggregated time intervals. The information generated by the interactions between traders who buys and sells financial instruments such as stocks, bonds, commodities, currencies, derivatives and so on is apparently encoded in frequencies which are not usually equally spaced in time.

Thus if we observe a particular financial asset i.e., a currency pair on the Foreign Exchange Market, the quotes or rates of this financial asset are sampling by tick-by-tick intervals which do not follow a constant rhythm. Each tick will appear when there is a change, upward or downward, in the trade price of each transaction. Can we use this kind of information to build a dynamic system model, and if so, how is it related to the true data-generating process?. Huke and Broomhead (2007) showed how to extend the reconstruction theorem under certain conditions when the dynamic system is sampled non-uniformly in time. Chapter 6 provides a discussion about it and illustrates how to get the non-uniform delayed-coordinate embedding vectors from time-series data with non-uniform time-frequency. Now once we have shown how to deal with the state space reconstruction we are going to focus on the next step. We will provide an overview about the main traditional direct methods for estimating the Lyapunov exponents from time-series data.

3.3 Estimating Lyapunov exponents by traditional direct methods and its limitations

According to the literature countless techniques have been developed and used to estimate the complexity of time-series data, for a review see e.g., Faggini (2014), Bradley and Kantz (2015) or Tang et al. (2015). We have focused on methods derived from chaos theory which estimate the complexity of a dataset through exploring the structure of the attractor. Particularly, we have been interested in the so-called *Lyapunov exponent* (λ) as an attractor invariant measure, for a review see Section 2.3. Quantify chaos through this kind of quantitative measure is a key point for understanding a chaotic behaviour. Hence our interest will be to test the *hypothesis of chaos* defined as follows:

$$\begin{aligned} H_0 &: \hat{\lambda}_k > 0 \\ H_1 &: \hat{\lambda}_k \leqslant 0 \end{aligned} \tag{3.3.1}$$

for $k = 1, 2, 3, \ldots$ on a k-dimensional system. Reject the null hypothesis $H_0 : \hat{\lambda}_k > 0$ means lack of chaotic behaviour. That is, the data-generating process does not have a chaotic attractor because of it does not show the property of sensitivity to initial conditions, see Gençay and Dechert (1992).

Methods and techniques related to test the hypothesis of chaos try to quantify the initial-value sensitive property estimating the so-called Lyapunov exponents. If one knows the data-generating process behind the time series the *theoretical* Lyapunov exponent can be calculated directly using its own definition, see Section 2.4. However we have now assumed that the true dynamics of the system is unknown because in most real-world observed time series the data-generating process is rarely known a priori.

There are two main methods in the literature that provide the *estimated* Lyapunov exponent from time-series data. The first one, the so-called *direct* approach which directly measures the growth rate of the divergence between two trajectories with an infinitesimal difference in their initial conditions. The second one, the so-called *indirect* approach (or jacobian-based method) which try to estimate the jacobian of the underlying generating system and then those partial derivatives are used to compute the Lyapunov exponent applying their analytical definition. We will discuss both methods although we have focused in greater detail into the jacobian indirect methods because provide us consistent estimators and robustness to the presence of (small) measurement noise as we will show in subsequent chapters. In this section we are going to discuss briefly the main traditional direct methods for estimating the Lyapunov exponents from time-series data.

The idea behind the *direct approach* can be summarising as follows. The presence of a sharp linear region in the plot of the evolution of the logarithm of the mean distance between nearby points would be a strong signal of chaos. Some *advantages* of this method are the following: (*i*) their low number of estimation parameters; (*ii*) easy implementation; (*iii*) they provide direct visual feedback to the user whether the available time series really exhibits exponential divergence on small scales; (*iv*) do not involve any kind of modelling; (*v*) do not require assumptions on the nature of the process.

The main *drawbacks* of the direct methods are the followings: (*i*) it does not allow the estimation of the full spectrum of Lyapunov exponents; (*ii*) it is not robust to the presence of (small) measurement noise because these estimators can produce a linear scaling region giving a wrong chaotic Lyapunov exponent value even for non-chaotic systems; (*iii*) it does not have a satisfactory performance in detecting existing nonlinearities on time-series data of moderate sample sizes; (*iv*) the asymptotic distribution of the estimator does not exist which means that it does not allow the building of formal tests. Let us discuss briefly the *algorithms* proposed in this context.

The direct method was first provided by Wolf et al. (1985), then revisited by Rosenstein et al. (1993) and Kantz (1994). The underlying algorithm is explained in details in Kantz and Schreiber (1997). There are three R packages recently developed related to nonlinear time series analysis and chaos. The **tseriesChaos** package written by Narzo (2019), the **nonlinearTseries** package proposed by Garcia (2019) and the **fNonlinear** package provided by Wuertz et al. (2017). These R packages are based on ideas inspired by the time series analysis (TISEAN) project suggested by Hegger et al. (1999). All of them implement the algorithm written by Kantz (1994) using exclusively the *direct method* for estimating the Lyapunov exponent. Those R packages are publicly available at the Comprehensive R Archive Network (CRAN) `www.CRAN.R-project.org`. The idea behind those algorithms can be described as follows.

Algorithm 3.1 (Kantz (1994))**.** Given a time series of length n, embedded in a m-dimensional space reconstructed with a time delay τ, the main aim is to compute a local average of the distances between every point x_i in the embedding space and its neighbours for $1 \leqslant i \leqslant n - (m-1)\tau$. That is, the evolution of the logarithm of this (local) mean distance $L(\Delta)$ is monitored for a finite number Δ of step ahead in time. The formula for this distance can be written as

$$L(\Delta) = \frac{1}{T} \sum_{i=1}^{T} \ln \left(\frac{1}{\#\mathcal{U}(x_i, \epsilon)} \sum_{x_j \in \mathcal{U}(x_i,\epsilon)} |x_{i+\Delta} - x_{j+\Delta}| \right)$$

where $T = n - (m-1)\tau$ is the number of points in the state space that are involved in the computation, $\#\mathcal{U}(x_i, \epsilon)$ is the number of neighbours of each point that are closer than a distance equal to ϵ and have a temporal separation greater than a certain value. We compute the log-average distance between every point and its neighbours, and we follow it for Δ time steps ahead. Thus, for every point x_i we have approximately a straight line which represents the evolution of the logarithm of the local mean distance. The average line of all these straight lines gives the evolution of the mean distance for the whole attractor. The slope of this average line gives us the estimated values of the *Lyapunov exponent*. Note that the algorithm suggested by Kantz (1994) includes that proposed by Rosenstein et al. (1993) as a special case (their number of neighbours are equal to 1). All of them emphasise that their method must be considered an improvement with respect to earlier approaches.

As we said before a strong disadvantage of this kind of direct estimators is that there are no available theoretical results for its consistency and asymptotic distributions. Then as the asymptotic distribution of the estimator obtained from direct methods do not exist, there is not chance to make statistical inference about chaos. One way to overcome this problem is to make use of empirical approaches. Several attempts based on resampling techniques have been put forward as bootstrap methods allow the construction of empirical distributions of estimators from available data. Giannerini and Rosa (2001, 2004) proposed a resampling scheme that allows to derive confidence interval, in a rigorous statistical sense, of the Lyapunov exponent estimator suggested by Kantz (1994). They also established its consistency and a rate of convergence for such estimators by means of simulations. They applied their algorithms on continuous-time dynamic systems. Giannerini et al. (2007) also implemented those algorithms for testing the presence of two positive Lyapunov exponents of comparable magnitudes through rigorous assessment of confidence intervals. The idea behind this approach can be described as follows.

Algorithm 3.2 (Giannerini and Rosa (2001))**.** Given a time series of length n with a sampling rate equal to Δ. A new sampling rate Δ' is derived by dividing Δ by a factor ς. Then a point p in the interval $]\Delta, 2\Delta[$ is selected at random. A new series call it *replicated series* $\boldsymbol{x}_1^r = x_1^r, x_2^r, \ldots, x_{p'}^r$ (r stands for replicated) is obtained by interpolating the original series at the points $(p, p+1 \times \Delta', \ldots, p+p' \times \Delta')$ where p' is the last point less than $n \times \varsigma$, that is

$$p' = n \times \varsigma - \left[p \times \frac{\varsigma}{\Delta}\right]$$

where $[\cdot]$ indicates the integer part of a number. We can think of $\boldsymbol{x}_1^r$ as the *first bootstrap sample*. Then, we have to chose B times another points and create new series, obtaining B replicated series $x_1^r, \ldots, x_B^r$. The different replicated series are obtained by choosing the initial condition randomly. In each of these series, it is possible to estimate the statistic of interest, the direct estimator of the Lyapunov exponent proposed by Kantz (1994), obtaining its empirical distribution.

Remark 3.3. Note that the algorithm suggested by Giannerini and Rosa (2001) consists essentially in interpolating in the time domain between measured values, in order to obtain a number of new time series of increased size. That is, given the length n of the original series, the above procedure generates series of length $n \times \varsigma$. Usually, from such a replicated sequence we extract n elements in order to have series with the same length as the original one. They have shown that there is an identifiable range of values of ς for which the bootstrap distribution of the Lyapunov exponent estimator is in statistical agreement with the distribution of the true series. The distribution of the true series is obtained by partitioning a long series, which is equivalent to consider the replicated series generated from different initial conditions.

Now, let us discuss briefly the noise environment in this context. Between the 1980s and 1990s several papers appeared that tried to incorporate a purely stochastic component (*dynamic noise*) in the underlying generator system to obtain a direct estimator of the Lyapunov exponent from noisy data, see e.g., Crutchfield et al. (1982), Kifer (1986), Gerrard in association with Tong in an unpublished note at the Edinburgh International Workshop on Nonlinear Time Series in 1989 as outline Tong (1995), Jensen (1993) or Blackburn et al. (1995). But it seems that Herzel et al. (1987) were the first ones to suggest the idea of measuring directly the separation of the trajectories originated from two nearby initial values when disturbed by the same noise realisation. It was called the *identical-noise realization* approach. As far as this PhD thesis is concerned we will not enter in detail into this approach, leaving the interested reader to this particular research area see Chan and Tong (2001). In any case, it is important to make some comments.

Note that if two nearby initial states share the same realisation of the dynamic noise, then the above calculation and interpretation remain *almost unchanged*. The only difference lies in the fact that the average is now taken with respect to the invariant measures of the stochastic system instead of the original deterministic system, assuming that the stochastic system has a unique invariant measure.

The main *drawbacks* of the identical-noise realization approach are the followings: (*i*) it is not realistic to assume that the same random shocks or dynamic noise sequence will be applied as excitations even if we start a dynamic system with different initial values. Hence the assumption of identical noise realisations is further from the truth on real-world observed time-series data; (*ii*) the direct estimator of the Lyapunov exponent provided by the identical-noise realization approach is not invariant under one-to-one differentiable coordinate transformations (necessary condition of an embedding to preserve the dynamic properties) in contrast to the noise-free case, see e.g., Takens (1994) or Chan and Tong (2001); (*iii*) it lacks a phenomenological basis and its interpretation can be controversial, especially when random shocks are not small as point out Giannerini (2002). We postpone for future work how to extend this ideas in order to incorporate different stochastic components (dynamic noise) in the underlying generator system to obtain a robust estimator of the Lyapunov exponent from noisy data (*non-identical noise realization* approach). Now, let us focus on a very important feature that we think it is relevant to remark at this point.

Remark 3.4 (Spurious Lyapunov exponents)**.** Since Lyapunov exponents are invariant with respect to diffeomorphic changes of the coordinate system, the Lyapunov exponents estimated from the reconstructed system will coincide with those of the original system. Technically, their estimation remains a delicate task mainly due to the existence of *spurious* Lyapunov exponents. Since the m-dimensional reconstructed system has a larger dimension than the d-dimensional original system, the number of spurious Lyapunov exponents should be equal to $m - d$. They have to be identified (or avoided) because their values are not related to the dynamics to be characterized. Hence without taking precautions spurious Lyapunov exponents can occur between *true* exponents resulting in false conclusions about the number of positive exponents. This issue is discussed in detail by many authors through different approaches, for a review see e.g., Parlitz (1992), Gençay and Dechert (1996), Sauer et al. (1998), Tempkin and Yorke (2007), Yang et al. (2012), Kantz et al. (2013) or Pathak et al. (2017). We have to keep this fact in mind. We will now move on to the empirical results of this chapter.

3.4 Data analysis and simulations

In this section we are going to focus on the following practical issues: (*i*) how to get the delayed-coordinate embedding vectors from time-series data with uniform-time frequency; (*ii*) test the robustness of the traditional direct methods available for estimating the Lyapunov exponent. Remember that any method for estimating the Lyapunov exponent from some observed time-series data are based previously on the state space reconstruction procedure. The three R packages mentioned above have implemented their algorithms by the following functions.

The **fNonlinear** package has a function called `embeddPSR`, `buildTakens` is the function that belongs to the **nonlinearTseries** package and the **tseriesChaos** package includes the function `embedd`. The underlying idea is to make copies of the single observable signal and consider those delayed values as coordinates of a reconstructed state space retrieved from the time-series data. Let us illustrate the reconstruction procedure by the following example using the `embedd` function from the Logistic map. In this example we choose arbitrarily the following embedding parameters $m = 5$, $\tau = 2$ and $n = 1000$. The number of embedding vectors will be $n-(m-1)\,\tau = 1000-(5-1)\cdot 2 = 992$. Thus we have 992 points in the 5-dimensional reconstructed state space given by (only the first five values are showed):

```
## The time-series data from the Logistic map with chaos
[1] 7.47e-01 7.57e-01 7.37e-01 7.76e-01 6.95e-01 8.48e-01

## The uniform delayed-coordinate embedding vectors forward
data  <- tseriesChaos::embedd(ts, m=5, d=2)
show(head(data, 5))

        V1/0      V1/2      V1/4       V1/6       V1/8
[1,] 0.7466701 0.7365940 0.69509054 0.51625546 0.00422337
[2,] 0.7566155 0.7760930 0.84775873 0.99894304 0.01682213
[3,] 0.7365940 0.6950905 0.51625546 0.00422337 0.06615659
[4,] 0.7760930 0.8477587 0.99894304 0.01682213 0.24711960
[5,] 0.6950905 0.5162555 0.00422337 0.06615659 0.74420600
```

Hence we can write the reconstructed trajectory joining the points from $x_1, \ldots, x_{992}$ where $\boldsymbol{x}_t = (x_t, x_{t-\tau}, x_{t-2\tau}, x_{t-3\tau}, x_{t-4\tau})$ for $t = 1, \ldots, 992$, see definition 3.3. Keep in mind that the embedding procedure allows us to get all the relevant information (invariant properties) about the unknown underlying dynamical system that generates the time-series data e.g., the Lyapunov exponents must have approximately the same value in both the true and the reconstructed state space. This fact allows us to test the hypothesis of chaos in the unknown original dynamic system.

Now we are going to test the robustness of the traditional direct methods available for estimating the Lyapunov exponent. We have considered some noise-contaminated time-series data from four well-known chaotic dynamic systems. The Logistic map, the Gauss map, the Hénon map (in discrete-time) and the Rössler system (in continuous-time), for a review see table 2.4.1. We have added to each time-series data a normal multinomial error term denoted by $\varepsilon_t \sim N(0, s)$ with different variance values s. We have considered it appropriate to add a measurement noise term because most real-world observed time-series data are usually noise-contaminated signals. In this sense we want to know if as the measurement noise increases, the error committed in obtaining the estimator is amplified with the consequent inaccuracy and inconsistency or instead it is reduced.

The algorithms considered in this empirical analysis for estimating the Lyapunov exponents are the following. The **tseriesChaos** package includes the function `lyap.k`, the **nonlinearTseries** package has a function called `maxLyapunov` and `lyapunovPlot` is the function that belongs to the **fNonlinear** package. We are going to compare the results from these packages for different measurement noise levels. Note that the function `lyapunovPlot` have implemented exactly the same code than `lyap.k`. For this reason, we are going to consider only the algorithms included in the R packages **tseriesChaos** (D1) and **nonlinearTseries** (D2).

The command `set.seed(34)` will set the seed for reproducibility. To save CPU time we have set the embedding dimension $1 \leqslant m \leqslant 7$, the time-delay $\tau = 1$ and the length of all time-series data is $n = 1000$. Those R packages follow the heuristic approach for estimating the embedding parameters. The mean square error (MSE) are calculated between the theoretical value (see Section 2.4) and the estimated value provided by the traditional direct methods (see Section 3.3).

Logistic map	$s = 0$	$s = 0.01$	$s = 0.02$	$s = 0.03$	$s = 0.04$	$s = 0.05$
D1 direct method	0.0000220	0.0056643	0.0030120	0.003006	0.0033485	0.0030913
D2 direct method	0.0802315	0.4765133	0.4814125	0.4815446	0.4790305	0.4830895
Gauss map						
D1 direct method	0.0005270	0.0111180	0.0205349	0.0293853	0.0275621	0.0336681
D2 direct method	0.0474216	0.1480353	0.1477251	0.1464405	0.1481204	0.1476371
Hénon system						
D1 direct method	0.0005650	0.0067221	0.0092761	0.0100339	0.0141379	0.0189926
D2 direct method	0.0121588	0.3133259	0.3145991	0.3115671	0.3226997	0.3178221
Rössler system						
D1 direct method	0.0004471	0.0049521	0.0063189	0.0072719	0.0127326	0.0174911
D2 direct method	0.0398841	0.6412752	0.6388524	0.6396631	0.6451333	0.6499127

Table 3.4.1: The mean square error (MSE) values based on the estimation of the largest Lyapunov exponent from direct methods provided by the **tseriesChaos** (D1) and **nonlinearTseries** (D2) packages are showed.

We have used the Monte Carlo method. We have done 1000 repetitions by different initial conditions when simulating the time-series data from the four dynamic systems and six measurement noise levels considered. We have had to estimate 24000 Lyapunov exponents in order to obtain the results displayed in table 3.4.1.

The data shown provide the following comments. First, we can remark that both traditional direct methods do not provide us consistent estimators and robustness to the presence of (small) measurement errors because the results obtained are not comparable to those which are noise-free; Second, as the noise increases the error committed in obtaining the estimator is amplified with the consequent inaccuracy and inconsistency. Third, the algorithm proposed by the **tseriesChaos** package (D1) provide better estimates than those of the **nonlinearTseries** package (D2) in all the experiments we have conducted. Fourth, the **tseriesChaos** package (D1) has less computational complexity and the computation time is lower than the **nonlinearTseries** package (D2). Finally we have only focused on the largest Lyapunov exponent as direct methods do not estimate the full spectrum. In addition those methods do not allow us to make inference about it. For those reasons we will focus in subsequent chapters on the so-called *jacobian indirect methods* which solve those disadvantages.

3.5 Discussion

In this chapter we have presented a summary of the main concepts and definitions already established in the discipline of nonlinear time series analysis and chaos theory. We have also provide some theoretical features for detecting chaos from real-world observed time-series data which are usually noise-contaminated signals. We have illustrated the effect of adding just a (small) measurement noise to a well-known chaotic deterministic dynamic system as the Logistic map. As we have seen as the measurement error increased, the dispersion of the huge amount of points that describe the attractor was amplified with the consequent inaccuracy. Then the presence of a measurement noise can lead to confusion between a chaotic deterministic system and a purely stochastic one. In such cases the interplay between order and disorder, determinism and stochasticity, stability and instability or predictability and unpredictability is undergoing profound changes with very exciting implications on nonlinear time series analysis and chaos.

Remember that devising a dynamic system to model some time-evolving phenomena involves specifying a rule that governs the evolution of the process through time and the space in which the process takes values as we mentioned in the previous chapter, for a review see Section 2.2. If the rule that governs the evolution of the process is well understood it may be possible to write down the dynamic system more or less from first principles, without much need for observational data except perhaps to fix some parameters. Otherwise we may be much less clear about the underlying rules. That is, we might suspect that the system is governed by e.g., a set of difference equations, but not know what these are, even we may be uncertain what dynamic variables the equations involve.

Hence in this context we will have to assume that the true data-generating process is unknown. Thus it will not be possible to consider the true orbit of the dynamic system in the original state space. Instead of that, it will be possible to obtain an approximation (reconstruction) of it that result equivalent in a topological sense (equivalence in the dynamic and geometric properties). As we have seen the embedding (or reconstruction) procedure allows us to get all the relevant information about the unknown underlying dynamic system that generates the time-series data (invariant properties) e.g., the Lyapunov exponents defined in the previous chapter (for a review, see Section 2.3) have approximately the same value in both the true and the reconstructed state space.

That fact has allowed us to test the hypothesis of chaos in the unknown original dynamic system. Note that any method for estimating the Lyapunov exponent from some observed time-series data are based previously on the state space reconstruction procedure. We have illustrated how to get the uniform delayed-coordinate embedding vectors from time-series data with uniform time-frequency. It also been discussed some heuristic and statistical approaches in order to estimate the embedding parameters robustly.

Once we have shown how to deal with the state space reconstruction we have focused on the next step. Methods and techniques related to test the hypothesis of chaos try to quantify the initial-value sensitive property estimating the so-called Lyapunov exponents. There are two main methods in the literature for estimating the Lyapunov exponent from time-series data. In this chapter we have provided an overview about the main traditional direct methods for estimating the Lyapunov exponents from time-series data which directly measures the growth rate of divergence between two trajectories with an infinitesimal difference in their initial conditions. Then the underlying algorithms have been described in detail. We have reported the results obtained by some R packages recently developed related to nonlinear time series analysis and chaos. Those R packages are publicly available at the Comprehensive R Archive Network (CRAN) `www.CRAN.R-project.org`.

Finally we have illustrated empirically the main drawbacks of the direct estimators of the Lyapunov exponent. To sum up, the traditional direct methods do not provide us consistent estimators and robustness to the presence of (small) measurement errors; they do not have a satisfactory performance in detecting existing non-linearities on time-series data of moderate sample sizes; they do not allow the estimation of the full spectrum of Lyapunov exponents; the asymptotic distribution of the estimator does not exist which means that it does not allow the building of formal tests. For those reasons we will discuss in subsequent chapters the so-called *jacobian indirect methods* which solve those disadvantages.

Chapter 4

Estimating Lyapunov exponents by global indirect methods

"C'est par la logique qu'on démontre,
c'est par l'intuition qu'on découvre".

— Jules Henri Poincaré

Abstract: This chapter presents a summary of the main concepts and definitions related to the global jacobian indirect method for estimating the Lyapunov exponents from time-series data. Section 4.1 describes some key features about the global neural net approach by approximating the unknown non-linear system through a feed-forward single hidden layer neural network. It also provide the analytical derivation, rather than numerical, of the jacobian needed for the estimation of the Lyapunov exponents. Section 4.2 illustrates the correct procedure in order to obtain a consistent estimator of the Lyapunov exponent from the partial derivates computed previously by two different procedures and four ways of subsampling by blocks. Section 4.3 provides a statistical framework for testing the hypothesis of chaos based on the theoretical asymptotic properties of the neural net estimator of the Lyapunov exponent and the consistent estimator of its variance. Section 4.4 reports the empirical results of this chapter and a discussion of them. Section 4.5 contains some concluding remarks.

Most of this chapter is contained in the paper written by Julio E. Sandubete and Lorenzo Escot titled 'DChaos: An R Package for Chaotic Time Series Analysis' (submitted to *R Journal*). It has been also presented at the 2nd Spanish Young Statisticians and Operational Researchers Meeting on June 5-8, 2019.

4.1 Global jacobian indirect method: a neural net approach

Methods and techniques related to test the hypothesis of chaos (eq.3.3.1) try to quantify the initial-value sensitive property estimating the so-called Lyapunov exponents, for a review see Section 2.3. If one knows the data-generating process behind the time series the *theoretical* Lyapunov exponent can be calculated directly using its own definition, see Section 2.4. However we have now assumed that the true dynamics of the system is unknown because in most real-world observed time series the data-generating process is rarely known a priori. The embedding procedure allows us to get all the relevant information (invariant properties) about the unknown underlying dynamical system that generates the time-series data as we have seen in the earlier chapter. In this sense, the Lyapunov exponents defined previously must have approximately the same value in both the true and the reconstructed state space. This fact allows us to test the hypothesis of chaos in the unknown original dynamic system.

There are two main methods in the literature that provide the *estimated* Lyapunov exponent from time-series data. The first one, the so-called *direct* approach which directly measures the growth rate of the divergence between two trajectories with an infinitesimal difference in their initial conditions, for a review see Section 3.3. The second one, the so-called *indirect* approach (or jacobian-based method) which try to estimate the jacobian of the underlying generating system and then those partial derivatives are used to compute the Lyapunov exponent applying their analytical definition. Having discussed the traditional direct methods in the previous chapter we will now focus on greater detail into the jacobian indirect methods which solve the disadvantages of direct methods. In this section we are going to discuss the main features of this kind of indirect methods for estimating the Lyapunov exponents from time-series data.

The idea behind the *jacobian indirect approach* can be summarising as follows. One has to estimate the jacobian of the underlying generating system from the delayed-coordinate embedding vectors and then those partial derivatives are used to compute the Lyapunov exponent. Keep in mind that any method for estimating the Lyapunov exponent from some observed time-series data are based previously on the state space reconstruction method. Some *advantages* of the indirect method are the following: (*i*) it allows the estimation of the full spectrum of Lyapunov exponents; (*ii*) it provides us consistent estimators and robustness to the presence of (small) measurement noise as we will show later; (*iii*) it has a satisfactory performance in detecting existing nonlinearities on time-series data of moderate sample sizes; (*iv*) the asymptotic distribution of the estimator can be derived allowing the building of formal tests.

Let us discuss briefly the *algorithms* proposed in this context. There are three R packages recently developed related to nonlinear time series analysis and chaos. The **tseriesChaos** package written by Narzo (2019), the **nonlinearTseries** package proposed by Garcia (2019) and the **fNonlinear** package provided by Wuertz et al. (2017). These R packages are based on ideas inspired by the time series analysis (TISEAN) project suggested by Hegger et al. (1999). All of them implement the algorithm written by Kantz (1994) using exclusively the *direct method* for estimating the Lyapunov exponent. In this sense, we have developed a novel R package called **DChaos** which contains the algorithms that we have implemented in this chapter, see Sandubete and Escot (2020). These algorithms are publicly available at the Comprehensive R Archive Network (CRAN) `www.CRAN.R-project.org/package=DChaos`. The **DChaos** package provides the following contributions.

First, as far as we know the **DChaos** package is the first R library that provides the *jacobian indirect methods* proposed by Eckmann and Ruelle (1985) and Dechert and Gençay (1992) for estimating the Lyapunov exponents. These jacobian indirect methods solve all drawbacks which belong to the direct methods. Particularly, this package has focused on the neural net approach following McCaffrey et al. (1992) and Nychka et al. (1992) by approximating the unknown non-linear system through a feed-forward single hidden layer neural network. Those neural net methods provide a well-fit of any unknown linear or non-linear model. They also have the advantage to allow the analytical derivation, rather than numerically, the jacobian needed for the estimation of the Lyapunov exponents.

Second, the **DChaos** package allows the use of full sample estimation, but it also provides three different blocking methods for estimating the Lyapunov exponents following McCaffrey et al. (1992) and Shintani and Linton (2004). One of them is a new proposal based on the bootstrap method. The results provided by the three blocking methods improve the case of the full sample being very similar between them. Although, on average, the bootstrap method gives better results as we will show later.

Third, the **DChaos** package provides new algorithms implementing the formal test proposed by Shintani and Linton (2004) who provided a statistical framework for testing the hypothesis of chaos based on the theoretical asymptotic properties from the neural net estimator and its variance. R users might make statistical inference about Lyapunov exponents significance and test the hypothesis of chaos (eq.3.3.1). Remember that the asymptotic distribution of the estimator obtained from direct methods do not exist, that is, there is not chance to make statistical inference about chaos.

Fourth, as we have mentioned earlier any method for estimating the Lyapunov exponent from time-series data are previously based on the state space reconstruction procedure. A key point to create a suitable reconstruction of the state-space is to fix a criteria in order to estimate the embedding parameters. Researchers usually estimate them using heuristic approaches based on prescriptions proposed by e.g., Abarbanel (1996) or Kantz and Schreiber (2004). The main drawbacks of these heuristic approaches are the following: they are not intrinsically statistical; their results are not robust; they lead to estimators whose properties are unknown or largely unexplored; they do not take into account the results of any model fit. Although the **Dchaos** package also allows the use of heuristic methods we have implemented some alternative and consistent statistical methods based on model selection criteria following Chan and Tong (2001) which solves those disadvantages, for a review see Section 3.2.

Function	Description
`embedding`	Provides the delayed-coordinate embedding vectors backwards
`gauss.sim`	Simulates time-series data from the Gauss map
`henon.sim`	Simulates time-series data from the Hénon system
`jacobian.net`	Computes the partial derivatives from the best-fitted neural net model
`logistic.sim`	Simulates time-series data from the Logistic map
`lyapunov`	Estimates the Lyapunov exponent through several methods
`lyapunov.max`	Estimates the largest Lyapunov exponent by the Norma-2 procedure
`lyapunov.spec`	Estimates the Lyapunov exponent spectrum by the QR decomposition
`netfit`	Fits any standard feed-forward neural net model from time-series data
`rossler.sim`	Simulates time-series data from the Rössler system
`summary.lyapunov`	Summary method for a lyapunov object
`w0.net`	Estimates the initial parameter vector of the neural net model

Table 4.1.1: A summary of the functions available in the **DChaos** package

To sum up the **DChaos** package provides a novel R interface for researchers interested in the field of chaotic dynamic systems and nonlinear time series analysis and professors (and students) who teach (learn) courses related to those topics. There are 12 functions available in the **DChaos** package so far, see table 4.1.1. We are going to illustrate all of them as we go on explaining the theoretical process step by step.

From now on we assume that the true data-generating process is unknown. Hence we do not have the advantage of observing directly the state of the system X_t let alone knowing the functional form f that generate the dynamic associated with it. Instead of that, there is an *observer function* $f : \mathbb{R}^2 \to \mathbb{R}$ that include an *additive measurement error* ε_t which generates observations as $x_t = f(X_t, \varepsilon_t)$ where ε_t is a sequence of independent and identically distributed random variables for $t = 1, 2, 3, \ldots, n$. Therefore it is assumed that all information available is the noise-contaminated sequence $\{x_t\}_{t=1}^{n}$ as an univariate time-series data, for a review see Section 3.1.

Then we form a sequence of delayed vectors by associating for each time period a vector in a reconstructed state space $\mathbb{R}^m$, whose coordinates satisfy the following eq.3.2.1 $\boldsymbol{x}_t^m = \left(x_t, x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-2)\tau}, x_{t-(m-1)\tau}\right)$ where m is the *embedding dimension* and τ is the *reconstruction time-delay* (or lag). Once we have got the delayed-coordinate embedding vectors from time-series data we are going to focus on the next step. The procedure behind the *global jacobian indirect* method can be described as follows.

Definition 4.1 (Global jacobian indirect method)**.** Let assume following Gençay and Dechert (1992) that there exist a function $G : \mathbb{R}^m \to \mathbb{R}^m$ such that $\boldsymbol{x}_t^m = G\left(\boldsymbol{x}_{t-\tau}^m\right)$ where $\boldsymbol{x}_t^m$ are the uniform delayed-coordinate embedding vectors taking $x_t - x_{t-1} = x_s - x_{s-1}\ \forall t \neq s$. Under the assumption that the embedding is a homeomorphism, the map G is *topologically conjugate* to the unknown dynamic system F. This implies that certain dynamic properties of F and G are the same. In our case, the Lyapunov exponents of F and G should be the same, so we can focus on estimating the exponents from the map G. The dynamic system G may be expressed as a matrix by

$$\begin{pmatrix} x_t \\ x_{t-\tau} \\ \vdots \\ x_{t-(m-2)\tau} \\ x_{t-(m-1)\tau} \end{pmatrix} = G \begin{pmatrix} x_{t-\tau} \\ x_{t-2\tau} \\ \vdots \\ x_{t-(m-1)\tau} \\ x_{t-m\tau} \end{pmatrix}$$

$$= \begin{pmatrix} \upsilon\left(x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-2)\tau}, x_{t-(m-1)\tau}, x_{t-m\tau}\right) \\ x_{t-\tau} \\ \vdots \\ x_{t-(m-2)\tau} \\ x_{t-(m-1)\tau} \end{pmatrix} \tag{4.1.1}$$

The global jacobian corresponding to the reconstructed dynamic system G will be obtained as follows

$$\partial G = \begin{pmatrix} \frac{\partial \upsilon}{\partial x_{t-\tau}} & \frac{\partial \upsilon}{\partial x_{t-2\tau}} & \frac{\partial \upsilon}{\partial x_{t-3\tau}} & \cdots & \frac{\partial \upsilon}{\partial x_{t-(m-1)\tau}} & \frac{\partial \upsilon}{\partial x_{t-m\tau}} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{4.1.2}$$

Hence the estimation of the Lyapunov exponent by the jacobian indirect method is reduced to the estimation of the unknown nonlinear function $\upsilon : \mathbb{R}^m \to \mathbb{R}$. The different approaches that compose the indirect method differ in the algorithm used for the estimation of this function υ in the jacobian (eq.4.1.2). The main contributions focus on two different approaches. Firstly, those who use some kind of local lineal regression, see Sano and Sawada (1985), Eckmann et al. (1986), Brown et al. (1991) or their extension in the form of local polynomial regression proposed by Lu and Smith (1997). Second, other approaches use nonlinear models based on neural networks, see McCaffrey et al. (1992), Nychka et al. (1992), Dechert and Gençay (1992), Whang and Linton (1999), Shintani and Linton (2003, 2004). In this sense there is no analogous functions in the three R packages already mentioned since they consider exclusively the direct method.

The **DChaos** package have focused on the neural net approach which is a global nonparametric method that try to estimate the underlying dynamic system without imposing the restriction of local linearity. This approach has the following advantages over all other methods: (*i*) its robustness to the presence of (small) noise as the measurement errors present in most real-world observed time-series data; (*ii*) their satisfactory performance in detecting existing nonlinearities on time-series data of moderate sample sizes; (*iii*) allows the estimation of the full spectrum of Lyapunov exponents; (*iv*) the asymptotic distribution of the estimator can be derived allowing the building of formal tests. Now let us discuss briefly the *neural net approach* proposed in this context.

The neural network (or neural net) estimator of the Lyapunov exponent was first proposed by McCaffrey et al. (1992) and Nychka et al. (1992), and then revisited by Gençay and Dechert (1992) and by Shintani and Linton (2003, 2004). Note that in our case the main reason for using neural network models is not to look for the best predictive model but to estimate a model that captures the non-linear time dependence well enough and, additionally, allows us to obtain in an analytical way (instead of numerical) the jacobian functional of the unknown data-generating process (eq.4.1.2). The estimation of this jacobian will allow us to contrast the hypothesis of chaos (eq.3.3.1) using the equation 2.3.6.

Hornik et al. (1989) showed that any standard feed-forward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another for any desired degree of accuracy, providing sufficiently many hidden units. In this sense, the feed-forward networks are a class of universal approximations. Theoretically, neural nets are expected to perform better than other approximation methods, especially with high-dimensional models, since the approximation form is not so sensitive to the increasing dimension.

The results proposed by the authors mentioned above have enabled us to consider a neural network with just one single hidden layer. The number of hidden units (or neurones) in the single hidden layer is determined by statistical methods based on model selection criteria as it appears in the results of this chapter. Now let us illustrate how we have obtained a consistent neural net estimator based on the robust estimation of the function υ in the jacobian (eq.4.1.2). We have developed our own algorithm as follows.

Algorithm 4.1 (Neural net approach)**.** Let consider the m-dimensional reconstruction vector $\boldsymbol{x}_t = \upsilon\left(x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-2)\tau}, x_{t-(m-1)\tau}, x_{t-m\tau}\right)$ as defined by eq.4.1.1. The *neural network estimator* can be obtained by approximating the unknown nonlinear function υ through a feed-forward single hidden layer network with a single output by

$$\upsilon \approx \hat{\upsilon} = \Phi_0 \left[\hat{\alpha}_0 + \sum_{q=1}^{h} \hat{\omega}_{qo} \Phi_q \left(\hat{\alpha}_q + \sum_{j=1}^{m} \hat{\omega}_{jq} x_{t-j\tau} \right) \right] \tag{4.1.3}$$

where $\Phi_0 \in I$, $\hat{\alpha}_0$ is the estimated network bias from input, h is the number of neurones (or nodes) in the single hidden layer, $\hat{\omega}_{q0}$ are the estimated layers connection weights from input to hidden layer, Φ_q is the transfer function which in our case is the logistic function, $\hat{\alpha}_q$ is the estimated network bias from hidden layer, m is the embedding dimension and $\hat{\omega}_{jq}$ are the estimated layers connection weights from hidden layer to output. The issue of parameter estimation is reduced to a least squares problem in which the quantity to be minimized is defined by

$$\sum_{t=1}^{n} \left(x_t - \left[\alpha_0 + \sum_{q=1}^{h} \omega_{qo} \Phi_q \left(\alpha_q + \sum_{j=1}^{m} \omega_{jq} x_{t-j\tau} \right) \right] \right)^2$$

Note that in our case the main reason for using neural network models is not to look for the best predictive model but to estimate a model that captures the non-linear time dependence well enough and, additionally, allows us to obtain in an analytical way (instead of numerical) the jacobian functional of the unknown underlying generator system (eq. 4.1.2). The estimation of this jacobian or partial derivatives will later allow us to contrast our hypothesis of chaos using equation 2.3.6.

We have obtained the partial derivates of the jacobian in eq.4.1.2 applying the chain rule to eq.4.1.3 as

$$\frac{\partial \hat{v}}{\partial x_{t-j\tau}} = \Phi'_0(z_0) \sum_{q=1}^{h} \hat{\omega}_{q0} \Phi'_0(z_q) \hat{\omega}_{jq} \tag{4.1.4}$$

where

$$z_0 = \hat{\alpha}_0 + \sum_{q=1}^{h} \hat{\omega}_{q0} \Phi_q(z_q), \quad z_q = \hat{\alpha}_q + \sum_{j=1}^{m} \hat{\omega}_{jq} x_{t-j\tau}$$

and the estimated partial derivates are given by

$$\hat{\partial G} = \begin{pmatrix} \frac{\partial \hat{v}}{\partial x_{t-\tau}} & \frac{\partial \hat{v}}{\partial x_{t-2\tau}} & \frac{\partial \hat{v}}{\partial x_{t-3\tau}} & \cdots & \frac{\partial \hat{v}}{\partial x_{t-(m-1)\tau}} & \frac{\partial \hat{v}}{\partial x_{t-m\tau}} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{4.1.5}$$

Now let us discuss the functions available in the **DChaos** package in order to calculate analytically the partial derivatives from the best-fitted neural net model. The **DChaos** package provides the function `netfit` which arguments are (`serie`, `m`, `lag`, `timelapse`, `h`, `w0maxit`, `wtsmaxit`, `pre.white`, `trace`, `seed.t`, `seed`) to fit the unknown nonlinear function v in eq.4.1.5 through a feed-forward single hidden layer network. See R documentation for the interpretation of this parameter set. We have considered the `nnet` function that belongs to the **nnet** package to fit the neural net models.

Note that the process of adjustment to a neural network often suffers from being trapped in local optima and different initialization strategies should be taken into account. For this reason we have implemented the function `w0.net` which arguments are (`x`, `y`, `m`, `h`, `rangx`, `w0maxit`, `seed.t`, `seed`). This function estimates previously the initial parameter vector of the neural network being able to set the maximum number of iterations that we want to obtain setting `w0maxit`. In addition, by default the neural network estimation is initialized with a fixed seed denoted by `seed.t=TRUE` with a value equal to `seed=56666459`. The R users can let the seed be fixed either randomly by `seed.t=FALSE` or even fix other value of the seed to be able to replicate the results obtained. Let us illustrate some examples considering time-series data simulated from the Logistic map with a chaotic behaviour ($\mu = 4$) contained in the **DChaos** package (`logistic.sim`). The command `set.seed(34)` will set the seed for reproducibility.

Example 4.1 (Best-fitted neural network models)**.** The `netfit` function returns several objects. The best-fitted feed-forward single hidden layer neural net model is saved. It also contains some useful information about the best set of weights found, the fitted values, the residuals obtained or the best embedding parameters set chosen. The best 10 models are displayed on the console as we show below for `m` $\in \{1:4\}$, `lag` $\in \{1:3\}$ and `h` $\in \{2:10\}$. The first column is the neural net number, the second column is the embedding dimension, the third column is the lag or reconstruction delay considered, the fourth column is the number of neurones (or nodes) in the single hidden layer and the fifth column is the Bayesian Information Criterion (BIC) value corresponding to that neural net. Notice that the neural net models are sorted from lowest to highest BIC values. In this case the best-fitted neural net model has the following parameter set values `m=1`, `lag=1` and `h=9`. The BIC criteria is defined following Shintani and Linton (2003) by

$$BIC = \log(RSS) + \frac{\log(n)}{n}\left[1 + h\,(m+2)\right]$$

where RSS is the residual sum of squares, n is the number of observations, m is the embedding dimension and h is the number of neurones (or nodes) used in the single hidden layer as noted above.

```
## Simulates time-series data from the Logistic map with chaos
ts    <- DChaos::logistic.sim(a=4, n=1000)
[1] 7.47e-01 7.57e-01 7.37e-01 7.76e-01 6.95e-01 8.48e-01

## Provides the best-fitted neural network models
model <- DChaos::netfit(ts, m=1:4, lag=1:3, timelapse="FIXED",
         h=2:10)

 Best models:
   m lag  h   BIC
85 1   1  9 -15.96086
73 1   1  8 -15.95343
97 1   1 10 -15.91173
37 1   1  5 -15.29339
61 1   1  7 -15.26654
63 3   1  7 -15.04988
49 1   1  6 -15.04090
3  3   1  2 -14.74957
98 2   1 10 -14.17451
```

```
## Summary method for a nnet object: provides the best set of
## weights found for the best-fitted neural net model (#85)
summary(model)

a 1-9-1 network with 28 weights
options were - linear output units
 b->h1 i1->h1
 -0.91   0.38
 b->h2 i1->h2
 -6.29   4.61
 b->h3 i1->h3
  3.06  -3.11
 b->h4 i1->h4
  1.49   6.10
 b->h5 i1->h5
  0.11   0.64
 b->h6 i1->h6
  0.70   3.02
 b->h7 i1->h7
 -0.19   0.57
 b->h8 i1->h8
 -1.37   0.84
 b->h9 i1->h9
 -0.03  -1.44
 b->o h1->o h2->o h3->o h4->o h5->o h6->o h7->o h8->o h9->o
-4.75 -0.16 -3.30  2.97  0.64 -0.39  5.07 -0.45 -2.02 -2.27
```

Example 4.2 (Partial derivatives are calculated analytically from the best–fitted neural net model)**.** The **DChaos** package provides as well the function `jacobian.net` which arguments are (`model`, `data`, `m`, `lag`, `timelapse`, `h`, `w0maxit`, `wtsmaxit`, `pre.white`, `trace`, `seed.t`, `seed`) to compute the partial derivatives of the jacobian in eq.4.1.5 from the best-fitted feed-forward single hidden layer network. See R documentation for the interpretation of this parameter set. This function returns several objects considering the parameter set selected by the user. Partial derivatives are calculated analytically from the best-fitted neural net model. It also contains some useful information about the best-fitted feed-forward single hidden layer neural net model saved, the best set of weights found, the fitted values, the residuals obtained or the best embedding parameters set chosen. This function allows the R user uses the data previously obtained from the best-fitted neural network estimated by the `netfit` function if `model` is not empty. Otherwise `data` has to be specified. We provide below the first seven partial derivatives values (`dx1`) corresponding to the best-fitted neural net model estimated above (neural net #85).

```
## Computes analytically the partial derivatives from the
## best-fitted neural net model showed in the netfit example(#85)
jacobian <- DChaos::jacobian.net(model=model)
show(head(jacobian$jacobian))
```

```
         dx1
1 -1.9701373
2 -2.0499734
3 -1.8893034
4 -2.2064735
5 -1.5568866
6 -2.7836348
7 -0.1313633
```

Example 4.3 (Partial derivatives are calculated analytically without setting previously any neural net model)**.** Now we are going to provide the results obtained from the function `jacobian.net` without setting the best-fitted neural net model estimated previously. We have considered the dataset simulated from the logistic map again. We have chosen the following parameter set values `m=3`, `lag=1`, `timelapse=FIXED`, `h=2:10`, `w0maxit=100`, `wtsmaxit=1e6`, `pre.white=FALSE`, `trace=1`, `seed.t=TRUE` and `seed=56666459`. We show below the first seven partial derivatives values (`dx1`, `dx2`, `dx3`) corresponding to compute the partial derivatives of the jacobian following eq.4.1.5 from the best-fitted feed-forward single hidden layer network (neural net #6).

```
## Partial derivatives are calculated analytically without
## setting previously any neural net model
jacobian <- DChaos::jacobian.net(data=ts, m=3:3, lag=1:1,
            timelapse="FIXED", h=2:10)
```

```
 Best models:
  m lag  h   BIC
6 3   1  7 -15.04988
1 3   1  2 -14.74957
7 3   1  8 -13.73532
4 3   1  5 -13.63287
3 3   1  4 -13.29422
8 3   1  9 -13.08082
2 3   1  3 -11.79369
9 3   1 10 -11.78703
5 3   1  6 -11.61098
```

```
show(head(jacobian$jacobian))
```

```
         dx1           dx2           dx3
1 -1.8880449 -0.0003353808  0.0020567805
2 -2.2071082 -0.0007068399  0.0014469903
3 -1.5552770  0.0001417462  0.0026879632
4 -2.7896525 -0.0011615596  0.0003214048
5 -0.1343617  0.0029934521  0.0053918764
6 -3.9235626 -0.0012869863 -0.0015386951
7  3.9269526  0.0116947029  0.0046071740
```

4.2 Lyapunov exponent estimator: full and blocking samples

As we have said at the beginning one of the main features of chaos is the well-known *initial-value sensitivity* property. The standard notion in deterministic dynamic systems theory which quantifies the initial-value sensitivity is the *Lyapunov exponent*. So quantifying chaos through this kind of quantitative measure is a key point for detecting chaos from real-world observed time-series data which data-generating process is unknown. Remember that the Lyapunov exponent gives the average exponential rate of divergence of infinitesimal nearby initial conditions on the attractor. Note that this interpretation coincides exactly with the definition of initial-value sensitivity, so we can use the Lyapunov exponent to quantify the sensitivity property of two neighbouring points, for a review see Section 2.3. In this section we are going to provide the right procedure to obtain a consistent estimator of the Lyapunov exponent (eq.2.3.6) from the partial derivates estimated following eq.4.1.5. In addition, we are going to illustrate how to test the hypothesis of chaos (eq.3.3.1) based on the theoretical asymptotic properties of the neural net estimator in subsequent section. The ith *Lyapunov exponent estimator* can be described as follows.

Definition 4.2 (Lyapunov exponent estimator)**.** Let us consider the partial derivatives $\partial\hat{G}$ calculated analytically from the best-fitted neural net model as defined by eq.4.1.5. Once we have obtained them we are going to estimate the ith Lyapunov exponent (eq.2.3.6) given by the *Lyapunov exponent estimator* as *(global jacobian approach)*:

$$\hat{\lambda}_i = \lim_{M\to\infty} \frac{1}{M} \log \mu_i \left(\left| \partial\hat{G}^M \right| \right) \tag{4.2.1}$$

where μ_i is the ith largest eigenvalue provided by $\partial\hat{G}^M = \partial\hat{G}(x_M)\cdot\partial\hat{G}(x_{M-1})\cdot \ldots \cdot \partial\hat{G}(x_1)$ for $i = 1, 2, 3, \ldots, m$ where m denote the embedding dimension and $\partial\hat{G}(\cdot)$ are the partial derivates estimated above following equation 4.1.5.

Remark 4.1 (Full and blocking samples)**.** Note that it is necessary to distinguish between the *sample size* n used for estimating the partial derivatives of the jacobian in eq.4.1.5 and the *block length* M, defined in eq.4.2.1, which is the number of evaluation points (number of products of the jacobian) used for estimating the ith Lyapunov exponent. Since the number of evaluation points is less than or equal to the sample size T, the block length M can be also understood as the *sample size* of a *subsample*.

Hence the choice of the block length M is an important issue in practice. McCaffrey et al. (1992) discussed the optimal choice of the block length suggesting that averaging the Lyapunov exponent estimators from the *non-overlapping blocking method* ($B = n/M$) might reduce the overall bias, for a review see also Ellner et al. (1991) and Nychka et al. (1992). However, it should be noted that such an estimate in the one-dimensional case is identical to the estimate based on the *full sample* ($M = n$).

Whang and Linton (1999) illustrated that the Lyapunov exponent estimator can be derived not only from the non-overlapping blocking method but also from any other subsampling method. They proposed the *equally spaced blocking method* ($B = n/M$). In fact, they showed that the optimal subsample for the Lyapunov exponent estimator depends on the data generating process. Therefore, we may use either the non-overlapping or the equally spaced blocking method as a choice of subsample, even some kind of *bootstrap blocking method* which takes random samples without replacement by each block B, see table 4.2.1.

Blocking method	Sample size	Block length	Block number	Block subset
Full	n			
Non-overlapping	Blocks	M	$B = n/M$	$\begin{cases} 1, 2, \ldots, M \\ M+1, M+2, \ldots, 2M \\ \vdots \\ (B-1)M+1, (B-1)M+2, \ldots, BM \end{cases}$
Equally spaced	Blocks	M	$B = n/M$	$\begin{cases} 1, 1+B, 1+2B, \ldots, 1+(M-1)B \\ 2, 2+B, 2+2B, \ldots, 2+(M-1)B \\ \vdots \\ B, 2B, 3B, \ldots, BM \end{cases}$
Bootstrap	Blocks	M	$B = 100$	Randomly

Table 4.2.1: Blocking methods for figuring out the neural network estimator of the ith Lyapunov exponent by jacobian indirect methods.

The **DChaos** package allows the use of full sample estimation, but it also provides the three different blocking methods mentioned above for estimating the Lyapunov exponents. One of them is a *new proposal* based on the bootstrap method, see table 4.2.1. The first column shows the blocking methods considered. The second, third and fourth column gives the sample size, the block length and the block number of each subsampling method, respectively. The fifth column provides the way in which the algorithm picks the position of each element inside the block where each block B corresponds to one row. The bootstrap blocking method takes random samples without replacement by each block.

Now let us discuss the functions available in the **DChaos** package which provide several ways to figure out robustly the Lyapunov exponent estimator defined as eq.4.2.1 by the jacobian indirect method using the neural net approach. On the one hand if the R users have previously obtained the partial derivatives from the `jacobian.net` function they can apply directly the function `lyapunov.spec` which estimates the Lyapunov exponent spectrum taking into account the QR decomposition procedure. They can also use the function `lyapunov.max` which estimates only the largest Lyapunov exponent considering the Norma-2 procedure. The arguments of both functions are the same (`data`, `blocking`, `B`, `doplot`). See R documentation for the interpretation of this parameter set. Hence the **DChaos** package allows the R users to choose between two different procedures to obtain the neural net estimator of the Lyapunov exponent and four ways of subsampling by blocks: full sample, non-overlapping sample, equally spaced sample and bootstrap sample, for a review see table 4.2.1.

Thus the **DChaos** package provides 8 internal functions (one for each procedure and blocking method). These functions return several objects considering the parameter set selected by the user. The largest Lyapunov exponent (`lyapunov.max`) or the Lyapunov exponent spectrum (`lyapunov.spec`) by each blocking method are estimated. They also contain some useful information about the estimated jacobian, the best-fitted feed-forward single hidden layer neural net model, the best set of weights found, the fitted values, the residuals obtained, the best embedding parameters set chosen, the sample size or the block length considered by each blocking method. The blocking methods split the time-series data into several blocks to estimate a Lyapunov exponent for each subsample. The R users can choose the non-overlapping sample (`blocking = NOVER`), the equally spaced sample (`blocking = EQS`) or the bootstrap sample (`blocking = BOOT`). The mean and median values of the Lyapunov exponent for each block or subsample are saved. By default we recommend using the median value as it is more robust to the presence of outliers. Notice that the parameter `B` (a non-negative integer denoting the number of bootstrap iterations) will only be considered if the R users choose the bootstrap blocking method. Let us illustrate some examples.

Example 4.4 (Lyapunov exponent estimator by jacobian indirect methods using the neural net approach)**.** We are going to compare the results obtained from the `lyapunov.max` and `lyapunov.spec` functions for estimating the largest Lyapunov exponent and the Lyapunov exponent spectrum respectively. We have considered the dataset simulated from the logistic map again, the best-fitted neural net model estimated previously in the `jacobian.net` example (neural net #6) which best embedding parameters set chosen is `m=3`, `lag=1`, `h=7` and the partial derivatives are `dx1`, `dx2`, `dx3`, for a review see Example 4.3.

We show below the estimation of the largest Lyapunov exponent provided by the Norma-2 procedure and the Lyapunov exponent spectrum taking into account the results obtained by the QR decomposition procedure (bootstrap blocking method `B=1000`). The results provided by both methods are very similar between them (0.6943782 and 0.6942063 respectively) but the `lyapunov.spec` function also allows the estimation of the full spectrum of the Lyapunov exponents. The estimated values by both methods are certainly close to the theoretical value 0.6931472 (see table 2.4.1) and they are statistically significant at the 99% confidence level. This fact evidences the consistency of the proposed algorithms. The **DChaos** package provides the `summary` method for class `"lyapunov"` called `summary.lyapunov` as well.

```
## Provides the largest Lyapunov exponent by the Norma-2 procedure
exponent <- DChaos::lyapunov.max(data=jacobian, blocking="BOOT",
            doplot=FALSE)

## Summary method for a lyapunov object: reports the main results
## given in an object of class lyapunov [lyapunov.max]
summary(exponent)

Call:
Largest Lyapunov exponent

Coefficients:
             Estimate Std. Error   z value Pr(>|z|)
Exponent    0.6943782 0.00443619  1783.442        1
---
Procedure: Norma-2 by bootstrap blocking method
Embedding dimension: 3, Time-delay: 1, No. hidden units: 7
Sample size: 997, Block length: 82, No. blocks: 1000
```

```
## Provides the Lyapunov exponent spectrum by the QR procedure
exponent <- DChaos::lyapunov.spec(data=jacobian, blocking="BOOT",
            doplot=FALSE)

## Summary method for a lyapunov object: reports the main results
## given in an object of class lyapunov [lyapunov.spec]
summary(exponent)

Call:
Lyapunov exponent spectrum

Coefficients:
             Estimate Std. Error   z value Pr(>|z|)
Exponent 1  0.6942063 0.00309979  1670.773        1
Exponent 2 -3.5818150 0.00915042 -3873.927        0
Exponent 3 -3.7257022 0.00898081 -4053.132        0
---
Procedure: QR decomposition by bootstrap blocking method
Embedding dimension: 3, Time-delay: 1, No. hidden units: 7
Sample size: 997, Block length: 82, No. blocks: 1000
```

On the other hand the **DChaos** package provides an *all-in-one* function called `lyapunov`. We have considered it appropriate to incorporate a function that unifies the whole process to make it easier and more intuitive for the R users. The `lyapunov()` function has the following *usage*:

```
lyapunov(data, m, lag, timelapse, h, w0maxit, wtsmaxit, pre.white,
         lyapmethod, blocking, B, trace, seed.t, seed, doplot)
```

This function has *fifteen* main arguments where only the first one is mandatory. These are:

> `data`: a `vector`, a time-series object `ts` or `xts`, a `data.frame`, a `data.table` or a `matrix` depending on the method selected in `timelapse`. If `FIXED` has been selected `data` must be a `vector` or a time-series object `ts` or `xts`. Otherwise `VARIABLE` has to be specified. In this case `data` must be a `data.frame`, a `data.table` or a `matrix`.
>
> `m`: a non-negative integer denoting a lower and upper bound for the embedding dimension (Default 1:4).
>
> `lag`: a non-negative integer denoting a lower and upper bound for the the reconstruction delay (Default 1:1).

`timelapse:` a character denoting if the time-series data are sampled at uniform time-frequency e.g., 1-month, 1-day, 1-hour, 30-min, 5-min, 1-min and so on `FIXED` or non- uniform time-frequency which are not equally spaced in time `VARIABLE` (Default `FIXED`).

`h:` a non-negative integer denoting a lower and upper bound for the number of neurones (or nodes) in the single hidden layer (Default 2:10).

`w0maxit:` a non-negative integer denoting the maximum iterations to estimate the initial parameter vector of the neural net models (Default 100).

`wtsmaxit:` a non-negative integer denoting the maximum iterations to estimate the weights parameter vector of the neural net models (Default 1e6).

`pre.white:` a logical value denoting if the user wants to use as points to evaluate the partial derivatives the delayed vectors filtered by the neural net model chosen `TRUE` or not `FALSE` (Default `TRUE`).

`lyapmethod:` a character denoting the procedure chosen to estimate the Lyapunov exponent. If `LLE` has been selected the function will estimate only the largest Lyapunov exponent through the Norma-2 method. If `SLE` has been selected the function will estimate the Lyapunov exponent spectrum through the QR decomposition. Otherwise `ALL` has to be specified. In this case the function will estimate the Lyapunov exponent considering both procedures (Default `SLE`).

`blocking:` a character denoting the blocking method chosen for figuring out the Lyapunov exponent. Available options are `FULL` if the user considers the full sample, `NOVER` if the user considers the non-overlapping sample, `EQS` if the user considers the equally spaced sample, `BOOT` if the user considers the bootstrap sample or `ALL` if the user considers all of them (Default `BOOT`).

`B:` a non-negative integer denoting the number of bootstrap iterations (Default 1000).

`trace:` a binary value denoting if the user wants to print the output on the console `1` or not `0` (Default `1`).

`seed.t:` a logical value denoting if the user wants to fix the seed `TRUE` or not `FALSE` (Default `TRUE`).

`seed:` a non-negative integer denoting the value of the seed selected if `seed.t = TRUE` (Default `56666459`).

`doplot:` a logical value denoting if the user wants to draw a plot `TRUE` or not `FALSE`. If it is `TRUE` the evolution of the Lyapunov exponent values are represented for the whole period considering the different procedures and blocking methods chosen by the user. (Default `TRUE`).

To sum up, the `lyapunov()` function provides at the same time the delayed-coordinate embedding vector from time-series data (eq.3.2.1), estimates the best-fitted neural net model from the delayed-coordinate embedding vectors (eq.4.1.3), calculates analytically the partial derivatives from the chosen neural nets model (eq.4.1.5). Finally, the Lyapunov exponent estimator (eq.4.2.1) is obtained from the partial derivatives computed previously. This function return several objects considering the parameter set selected by the user. The largest Lyapunov exponent (`lyapmethod = LLE`), the Lyapunov exponent spectrum (`lyapmethod = SLE`) or both (`lyapmethod = ALL`) by each blocking method are estimated. They also contain some useful information about the estimated jacobian, the best-fitted feed-forward single hidden layer neural net model, the best set of weights found, the fitted values, the residuals obtained, the best embedding parameters set chosen, the sample size or the block length considered by each blocking method. Let us show an example about this all-in-one function.

Example 4.5 (Lyapunov exponent estimator by jacobian indirect methods using the neural net approach)**.** Now we are going to provide the results obtained from the `lyapunov` function considering the dataset simulated from the logistic map again for the following parameter set values `m=3:3`, `lag=1:1`, `timelapse=FIXED`, `h=2:10`, `w0maxit=100`, `wtsmaxit=1e6`, `trace=1`, `seed.t=TRUE`, `seed=56666459` and `doplot=FALSE`. In this case we have estimated the Lyapunov exponent spectrum taking into account the results provided by the QR decomposition procedure (`lyapmethod=SLE`) and all blocking methods (`blocking=ALL`). Note that the delayed-coordinate embedding vector, the best-fitted neural net model (with a lower BIC value) and the partial derivatives are internally calculated according to the parameter set chosen (all-in-one way). We provide again the results obtained from the `summary` method for a `lyapunov` object.

```
## Provides the Lyapunov exponent spectrum by several blocking
## methods (all-in-one way)
exponent <- DChaos::lyapunov(ts, m=3:3, lag=1:1, timelapse="FIXED",
            h=2:10, w0maxit=100, wtsmaxit=1e6, pre.white=TRUE,
            lyapmethod="SLE", blocking="ALL", B=100, trace=1,
            seed.t=TRUE, seed=56666459, doplot=FALSE)
```

```
 Best models:
  m lag  h       BIC
8 3   1  9 -13.37732
9 3   1 10 -12.90353
7 3   1  8 -12.76361
4 3   1  5 -12.60372
2 3   1  3 -12.20489
5 3   1  6 -11.88894
6 3   1  7 -11.85880
1 3   1  2 -11.75219
3 3   1  4 -10.76974

## Summary method for a lyapunov object: reports the main results
## given in an object of class lyapunov [lyapunov]
summary(exponent)

Call:
Lyapunov exponent spectrum

Coefficients:
             Estimate Std. Error    z value Pr(>|z|)
Exponent 1  0.6922016 0.08452517   74.15722        1
Exponent 2 -2.7082326 0.09158355 -267.77832        0
Exponent 3 -3.2056094 0.10913227 -265.98941        0
---
Procedure: QR decomposition by bootstrap blocking method
Embedding dimension: 3, Time-delay: 1, No. hidden units: 9
Sample size: 997, Block length: 82, No. blocks: 100

... only the first method is shown (see lyapunov object)
```

Let us point out briefly some useful information to understand the results presented in this example. For the estimation based on blocking methods median values of all used blocks are given. For the block length M the **DChaos** package uses $M = int\left[c \times (n/\log n)^{1/6}\right]$ with $c = 36.2$ where $int\,[A]$ signifies the integer part of A following Shintani and Linton (2004). The number of blocks B depends on the sample size n of each time-series data. QS Kernel with optimal bandwidth has been used for the heteroskedasticity and autocorrelation consistent covariance matrix estimation following Andrews (1991). In this example the results provided by the three blocking methods improve the case of the full sample being very similar between them. This result is consistent with the recommendation proposed by Shintani and Linton (2004) on the use of blocking methods instead of full sampling. Now we are going to focus on the asymptotic properties of the estimator we just saw.

4.3 Lyapunov exponent inference and asymptotic distribution

The asymptotic properties of the Lyapunov exponent estimator defined as eq.4.2.1 was derived by Shintani and Linton (2004). They provided a statistical framework for testing the hypothesis of chaos (eq.3.3.1) based on the neural net estimator of the Lyapunov exponent and the consistent estimator of its variance. The asymptotic properties of the estimator can be described as follows.

Definition 4.3 (Asymptotic distribution)**.** Let λ_i be the theoretical Lyapunov exponent derived by eq.2.3.6, for a review see Section 2.3. Let $\hat{\lambda}_i$ be the non-parametric neural network estimator of the Lyapunov exponent (eq.4.2.1) for a block length M and $i = 1, \ldots, m$ where m denote the embedding dimension. Shintani and Linton (2004) proved the asymptotic normality of the Lyapunov exponent estimator under certain conditions by

$$\sqrt{M}\left(\hat{\lambda}_i - \lambda_i\right) \sim N\left(0, \varphi_i\right) \tag{4.3.1}$$

where φ_i is the variance of the ith Lyapunov exponent estimator (eq.4.2.1). They proved that $\hat{\varphi}_i$ is a consistent variance estimator of φ_i. It can be defined as follows

$$\hat{\varphi}_i \equiv Var\left(\hat{\lambda}_i\right) = \lim_{M\to\infty} Var\left(\frac{1}{\sqrt{M}}\sum_{t=1}^{M}\eta_{i,t}\right) \tag{4.3.2}$$

where M is the block length, that is equal to n for the full sample. The quadratic spectral kernel function $\eta_{i,t}$ is given by $\eta_{i,t} = \xi_{i,t} - \hat{\lambda}_i$ following Shintani and Linton (2004). The parameter $\xi_{i,t}$ is obtained by

$$\xi_{i,t} = \frac{1}{2}\log\mu_i\left(\left|\hat{\partial G}^t\right|\right) - \frac{1}{2}\log\mu_k\left(\left|\hat{\partial G}^{t-1}\right|\right) \tag{4.3.3}$$

where $\hat{\partial G}^t = \hat{\partial G}\left(x_t\right)\cdot\hat{\partial G}\left(x_{t-1}\right)\cdot\ldots\cdot\hat{\partial G}\left(x_1\right)$ and $\hat{\partial G}\left(\cdot\right)$ are the partial derivates estimated above following equation 4.1.5 for $t = 1, 2, \ldots, M$. Then a feasible test statistics were introduced and a one-sided test was proposed for the purpose of testing the hypothesis of chaos (eq.3.3.1) based on the theoretical asymptotic properties of the neural net estimator. That is, we would know if the estimated Lyapunov exponent values are or not statistically significative. Hence our interest will be to test the null hypothesis $H_0 : \hat{\lambda}_i > 0$ against the alternative $H_1 : \hat{\lambda}_i \leqslant 0$. The test statistics can be defined as follows

$$\hat{t}_i = \frac{\hat{\lambda}_i}{\sqrt{\hat{\varphi}_i/M}} \sim N\left(0, \hat{\varphi}_i\right) \tag{4.3.4}$$

Hence we will reject the null hypothesis if $\hat{t}_i \leqslant -z_\alpha$ where z_α is the critical value that satisfies $\Pr[Z \geqslant z_\alpha] = \alpha$ with Z being a standard normal random variable and α is the significance level. Under the null hypothesis H_0 that the data-generating process is chaotic, the neural net estimator $\hat{\lambda}_i$ leads to asymptotically valid inferences in that the associated p-value follows a normal distribution on $N(0, \hat{\varphi}_i)$. Reject the null hypothesis $H_0 : \hat{\lambda}_i > 0$ means lack of chaotic behaviour. Thus we have used these results to calculate the standard error of the Lyapunov exponent estimator and investigate the statistical significance of the sign of the exponents.

The **DChaos** package provides new algorithms implementing the formal test proposed by Shintani and Linton (2004) who provided a statistical framework for testing the hypothesis of chaos based on the theoretical asymptotic properties from the neural net estimator and its variance as we have just seen. The **DChaos** package contains the three functions discussed in the previous section (`lyapunov`, `lyapunov.spec` and `lyapunov.max`) which provide the standard error, the z test value and the p-value for testing the hypothesis of chaos (eq.3.3.1). As we said before, reject the null hypothesis H_0 means lack of chaotic behaviour. That is, the data-generating process does not have a chaotic attractor because of it does not show the initial-value sensitivity property, see e.g., Example 4.4 or Example 4.5 to know how it works.

Remember that the asymptotic distribution of the estimator obtained from direct methods do not exist, that is, there is not chance to make statistical inference about the direct Lyapunov exponent estimator and test if the estimated values are or not statistically significants. In this sense there is no analogous functions in the three R packages considered in the previous chapter since they consider exclusively the direct method, for a review see Section 3.3. We have to keep this fact in mind. We will now move on to the empirical results of this chapter.

4.4 Data analysis and simulations

In this section we will focus on studying the robustness of the algorithms available in the **DChaos** package for estimating the Lyapunov exponent. We have considered some noise-contaminated time-series data from four well-known chaotic dynamic systems. The Logistic map, the Gauss map, the Hénon map (in discrete-time) and the Rössler system (in continuous-time), for a review see table 2.4.1. We have added to each time-series data a normal multinomial error term denoted by $\varepsilon_t \sim N(0, s)$ with different variance values s. We have considered it appropriate to add a measurement noise term because most real-world observed time-series data are usually noise-contaminated signals. In this sense we want to know if as the measurement noise increases, the error committed in obtaining the estimator is amplified with the consequent inaccuracy and inconsistency or instead it is reduced.

For this purpose we have compared the results provided by the traditional direct methods considering the **tseriesChaos** package and the **nonlinearTseries** package (see Section 3.4) with those of the **DChaos** package. The algorithms considered in this empirical analysis for estimating the Lyapunov exponents are the following. The **tseriesChaos** package includes the function `lyap.k`, the **nonlinearTseries** package has a function called `maxLyapunov`, `lyapunov.max` and `lyapunov.spec` are the functions that belong to the **DChaos** package. Those R packages are publicly available at the Comprehensive R Archive Network (CRAN) `www.CRAN.R-project.org`.

The command `set.seed(34)` will set the seed for reproducibility. To save CPU time we have set the embedding dimension $1 \leqslant m \leqslant 7$, the time-delay $\tau = 1$, the number of nodes in the single hidden layer $2 \leqslant h \leqslant 10$ and the length of all time-series data is $n = 1000$. The R packages which implement the direct method proposed by Kantz (1994) exclusively use the heuristic approach for estimating the embedding parameters. The **DChaos** package uses by default the statistical approach based on model selection procedures. It also allows R users to choose the heuristic approach, for a review see Section 3.2.

In this empirical analysis 63 different neural nets models have been estimated from each of the 24000 simulated series in order to obtain the results shown in table 4.4.1. Then the neural net models are sorted from lowest to highest BIC values. After that the best-fitted neural net models with a lower BIC value and their associated embedding parameters are considered for estimating the 24000 largest Lyapunov exponents. Finally the mean square error (MSE) are calculated between the theoretical value (see Section 2.4) and the value estimated by several direct (see Section 3.3) and indirect methods (see Section 4.3).

To do that, we have used the Monte Carlo method. We have done 1000 repetitions by different initial conditions when simulating the time-series data from the four dynamic systems and six measurement noise levels considered. We have taken into account only the bootstrap blocking method and the number of bootstrap iterations is $B = 1000$. The MSE values based on the estimation of the largest Lyapunov exponent from the direct methods provided by the **tseriesChaos** and **nonlinearTseries** packages are denoted by D1 and D2 respectively. Those obtained by the jacobian indirect methods through the **DChaos** library are denoted by N2 (`lyapunov.max`) and QR (`lyapunov.spec`) regarding the Norma-2 and QR decomposition procedures respectively. The data shown in table 4.4.1 provide the following comments.

First, we can remark that our algorithms are robust to the presence of (small) measurement errors because the results obtained are comparable to those which are noise free. Although as the noise increases the error committed increases but it is not proportional in any case. Second, the indirect methods provide better estimates than direct methods in all the experiments we have conducted. The algorithms proposed by the **tseriesChaos** package behave better than those of the **nonlinearTseries** package. Between the two methods available in the **DChaos** package we do not observe significant differences. Third, the direct methods are surely less flexible and robust but much faster and still informative. The jacobian indirect methods based on the neural net approach seems to perform well for every noisy time-series data. The price we have to pay is a greater computational complexity from two points of view, the computing time and the tuning parameters (node weights). Fourth, we have only focused on the largest Lyapunov exponent as direct methods do not estimate the full spectrum. In addition those methods do not allow us to make inference about it. In our case, we will be able to do so. Hence we will focus on testing the reliability of the algorithms provided by the **DChaos** package.

Logistic map	$s = 0$	$s = 0.01$	$s = 0.02$	$s = 0.03$	$s = 0.04$	$s = 0.05$
D1 direct method	0.0001220	0.0056643	0.0030120	0.003006	0.0033485	0.0030913
D2 direct method	0.0802315	0.4765133	0.4814125	0.4815446	0.4790305	0.4830895
N2 indirect method	0.0000324	0.0000382	0.0000691	0.0000994	0.0001314	0.0001532
QR indirect method	0.0000331	0.0000348	0.0000672	0.0000986	0.0000997	0.0001124
Gauss map						
D1 direct method	0.0005270	0.0111180	0.0205349	0.0293853	0.0275621	0.0336681
D2 direct method	0.0474216	0.1480353	0.1477251	0.1464405	0.1481204	0.1476371
N2 indirect method	0.0000436	0.0000526	0.0000555	0.0000678	0.0000719	0.0000944
QR indirect method	0.0000618	0.0000656	0.0000672	0.0000782	0.0000817	0.0000924
Hénon system						
D1 direct method	0.0005650	0.0067221	0.0092761	0.0100339	0.0141379	0.0189926
D2 direct method	0.0121588	0.3133259	0.3145991	0.3115671	0.3226997	0.3178221
N2 indirect method	0.0000365	0.0000486	0.0000635	0.0000761	0.0000899	0.0000917
QR indirect method	0.0000318	0.0000451	0.0000589	0.0000601	0.0000866	0.0000932
Rössler system						
D1 direct method	0.0004471	0.0049521	0.0063189	0.0072719	0.0127326	0.0174911
D2 direct method	0.0398841	0.6412752	0.6388524	0.6396631	0.6451333	0.6499127
N2 indirect method	0.0002477	0.0003529	0.0005997	0.0006122	0.0009521	0.0019947
QR indirect method	0.0003168	0.0004891	0.0006070	0.0007155	0.0008190	0.0009268

Table 4.4.1: The mean square error (MSE) values based on the estimation of the largest Lyapunov exponent from direct methods provided by the **tseriesChaos** (D1) and **nonlinearTseries** (D2) packages are showed. Also those obtained by the jacobian indirect methods through the **DChaos** library are presented (N2 for `lyapunov.max` and QR for `lyapunov.spec`).

For this purpose we have calculated the so-called *size* and *power* of our hypothesis test. Due to the test is based on probabilities, there is always a chance of making an incorrect conclusion. When one does a hypothesis test, two types of errors are possible. As a reminder when the null hypothesis is true and we reject it, we make a *type I error*. The probability of making a type I error is denoted by α, which is the significance level that we set for our hypothesis test. An α of 0.05 indicates that we are willing to accept a 5% chance that we are wrong when we reject the null hypothesis. To lower this risk, we must use a lower value for α.

However, using a lower value for alpha means that we will be less likely to detect a true difference if one really exists. The probability of not rejecting the null hypothesis when it is true (not committing a type I error) is called the *size of the test*. When the null hypothesis is false and we fail to reject it, we make a *type II error*. The probability of making a type II error is called beta, and is often denoted by β. We can decrease our risk of committing a type II error by ensuring our test has enough power. We can do this by ensuring our sample size is large enough to detect a practical difference when one truly exists. The probability of rejecting the null hypothesis when it is false (not committing a type II error) is called the *power of the test*.

In order to understand the interrelationship between the type error I and II in this particular case we have to consider the following. As we pointed out before a feasible test statistics were introduced and a one-sided test was proposed for the purpose of testing the hypothesis of chaos (eq.3.3.1) based on the theoretical asymptotic properties of the neural net estimator. Under the null hypothesis H_0 that the data-generating process is chaotic, the neural net estimator $\hat{\lambda}_k$ leads to asymptotically valid inferences in that the associated p-value follows a normal distribution on $N\left(0, \hat{\varphi}_k\right)$.

Hence our interest will be to test the null hypothesis $H_0 : \hat{\lambda}_k > 0$ against the alternative $H_1 : \hat{\lambda}_k \leqslant 0$. We will reject the null hypothesis if $\hat{t}_k \leqslant -z_\alpha$ where z_α is the critical value that satisfies $\Pr\left[Z \geqslant z_\alpha\right] = \alpha$ with Z being a standard normal random variable and α is the significance level. Reject the null hypothesis $H_0 : \hat{\lambda}_k > 0$ means lack of chaotic behaviour. That is, the data-generating process does not have a chaotic attractor because of it does not show the initial-value sensitivity property. Thus we have used these results to calculate the standard error of the Lyapunov exponent estimator and investigate the statistical significance of the sign of the exponents in order to test the reliability of the algorithms available in the **DChaos** package for estimating the Lyapunov exponent.

$n = 50$	$s = 0$	$s = 0.01$	$s = 0.02$	$s = 0.03$	$s = 0.04$	$s = 0.05$
Logistic map	2.40	2.60	3.60	3.50	3.30	4.10
Gauss map	2.70	3.60	4.20	4.70	4.30	4.70
Hénon system	2.90	3.70	3.90	4.00	4.50	4.30
Rössler system	3.10	3.50	4.70	4.90	4.50	5.10
$n = 100$						
Logistic map	1.20	1.50	2.30	2.50	3.40	3.90
Gauss map	1.50	1.70	1.90	2.70	3.10	3.50
Hénon system	1.70	1.90	2.20	3.30	3.70	4.00
Rössler system	1.70	1.80	2.30	4.10	3.70	4.70
$n = 200$						
Logistic map	0.30	0.50	1.20	1.40	1.50	2.10
Gauss map	0.50	0.80	0.90	1.10	1.40	2.50
Hénon system	0.40	0.60	0.70	1.80	1.90	2.20
Rössler system	0.50	1.10	1.30	1.70	2.50	3.30

Table 4.4.2: Rejection percentages at the 5% significance level from the dynamic systems considered with a chaotic behaviour. These results provide the size of our hypothesis test.

The results reports in tables 4.4.2-4.4.3 are given in terms of rejection percentages of the tests at the 5% significance level over 1000 Monte Carlo replications. The rejection percentages provide a measure of the size (tab.4.4.2) and the power (tab.4.4.3) of the tests. In this case we have chosen the length $n = 50, 100, 200$ as we want to check the reliability of the algorithms in small sample sizes. Table 4.4.2 provides the results from *chaotic* dynamic systems which have the following parameter set values: $\mu = 4$ (logistic map); $\alpha = 6.2, \beta = -0.5$ (gauss map); $a = 1.4, b = 0.3$ (hénon system) and $a = 0.2, b = 0.2, c = 5.7$ (rössler system). Table 4.4.3 gives the results from *non-chaotic* dynamic systems which have the following parameter set values: $\mu = 3.2$ (logistic map); $\alpha = 4.9, \beta = -0.58$ (gauss map); $a = 1.2, b = 0.1$ (hénon system) and $a = 0.1, b = 0.1, c = 7$ (rössler system).

The data shown in both tables provides the following comments. First, we can point out that the reliability of the tests is solid at the 5% significance level. The rejection percentages in almost every situation, even for $n = 50$, are low when the null hypothesis H_0 is true and high when H_0 is false. Second, the results provide a satisfactory performance in moderate sample sizes. This fact is really important for those researchers who work with short time-series data. Third, the empirical size decrease and the empirical power increase as the sample size increase which means that our tests are consistent as well. Fourth, the noise-contaminated data are comparable to those which are noise free. Although as the noise increases the errors committed increases but not significantly. Fifth, the results shown are fairly robust with respect to the parameter values of the four dynamic systems considered.

$n = 50$	$s = 0$	$s = 0.01$	$s = 0.02$	$s = 0.03$	$s = 0.04$	$s = 0.05$
Logistic map	99.30	98.90	98.40	97.70	97.50	96.90
Gauss map	99.40	99.00	98.70	98.50	97.10	97.00
Hénon system	99.20	98.90	98.60	97.90	97.40	96.90
Rössler system	98.70	98.10	97.40	97.00	96.70	96.20
$n = 100$						
Logistic map	99.70	99.00	98.70	98.20	97.70	97.20
Gauss map	99.70	99.30	98.90	98.70	98.40	97.70
Hénon system	99.60	99.00	98.80	98.20	97.70	97.10
Rössler system	99.40	98.80	98.30	97.70	97.20	96.90
$n = 200$						
Logistic map	99.90	99.40	99.00	98.70	98.20	97.90
Gauss map	100.00	99.60	99.30	99.00	98.80	98.20
Hénon system	99.80	99.30	99.00	98.60	98.10	97.80
Rössler system	99.70	99.00	98.70	98.20	97.50	97.00

Table 4.4.3: Rejection percentages at the 5% significance level from the dynamic systems considered with a non-chaotic behaviour. These results provide the power of our hypothesis test.

4.5 Discussion

In this chapter we have provided a summary of the main concepts and definitions related to the jacobian indirect methods for estimating the Lyapunov exponents from time-series data. These jacobian indirect methods solve all drawbacks which belong to the direct methods. Particularly, we have focus on the global neural net approach by approximating the unknown nonlinear system through a feed-forward single hidden layer neural network. Those neural net methods provide a well-fit of any unknown linear or nonlinear model. They also have the advantage to allow the analytical derivation, rather than numerically, the jacobian needed for the estimation of the Lyapunov exponents. This process has been illustrated by examples.

We have also described the right procedure to obtain a consistent estimator of the Lyapunov exponent from the partial derivates computed previously by two different procedures and four ways of subsampling by blocks. At this point, we have provided a discussion about the optimal sample size used for estimating the partial derivatives of the jacobian and the optimal block length considered which is the number of evaluation points (number of products of the jacobian) used for estimating the Lyapunov exponents. Then a feasible test statistics were introduced and a one-sided test was proposed for the purpose of testing the hypothesis of chaos based on the theoretical asymptotic properties of the neural net estimator of the Lyapunov exponent and the consistent estimator of its variance.

At the same time we have presented a novel R package called **DChaos** which contains the algorithms that we have implemented in this chapter. These algorithms are publicly available at the Comprehensive R Archive Network (CRAN) `www.CRAN.R-project.org/package=DChaos`. The **DChaos** package provides a novel R interface for researchers interested in the field of chaotic dynamic systems and nonlinear time series analysis and professors (and students) who teach (learn) courses related to those topics.

To sum up, R users may computes the delayed-coordinate embedding vector from time-series data, estimates the best-fitted neural net model, calculates analytically the partial derivatives from the chosen neural nets model. They can also estimate both the largest Lyapunov exponent through the Norma-2 procedure and the Lyapunov exponent spectrum through the QR decomposition procedure taking into account the full sample and three different methods of subsampling by blocks. It also allows the R users to test robustly the hypothesis of chaos in order to know if the data-generating process behind time series behave chaotically or not. The package's functionality has been illustrated by examples.

Then we have focus on studying the robustness of the algorithms proposed in this chapter for estimating the Lyapunov exponent. For this purpose we have compared the results provided by the traditional direct methods with those of the jacobian indirect methods. In this sense we can remark that the indirect methods provide better estimates than direct methods in all the experiments we have conducted. We have also shown empirically that our algorithms are robust to the presence of (small) measurement errors. Although as the noise increases the error committed increases but it is not proportional in any case.

Note that we have only focused on the largest Lyapunov exponent as direct methods do not estimate the full spectrum. In addition those methods do not allow us to make inference about it. In our case, we will be able to do so. Hence we have tested the reliability of the algorithms provided by the **DChaos** package. The results have reported a satisfactory performance in moderate sample sizes. This fact is really important for those researchers who work with short time-series data. The empirical size decreased and the empirical power increased as the sample size increased which means that our tests are consistent as well.

Finally let us remark that the different contributions that compose the jacobian indirect methods differ in the algorithm used for the estimation of the jacobian. In this chapter we have focused on the global neural net approach which try to estimate the underlying dynamic system without take into account the condition of local nonlinearity. In the subsequent chapter we are going to introduce two novel jacobian indirect methods that try to estimate the function υ in the jacobian (eq.4.1.2) by local polynomial kernel regressions and local neural nets models.

Chapter 5

Estimating Lyapunov exponents by local indirect methods

"I learned that courage was not the absence of fear, but the triumph over it. Everyone can rise above their circumstances and achieve success if they are dedicated and passionate about what they do".

— Nelson Mandela

Abstract: This chapter presents a summary of the main concepts and definitions related to two new local jacobian indirect methods for estimating the Lyapunov exponents from time-series data. Section 5.1 provides a discussion about the right procedure to obtain the partial derivatives based on the local jacobian approach. The main differences with the global jacobian approach are also highlighted. In addition, it illustrates how to get a consistent estimator of the Lyapunov exponent based on the local jacobian approach. Section 5.2 describes the key features about the local polynomial kernel method by approximating the unknown nonlinear system through a local polynomial of order greater than one. Section 5.3 provides the main details about the local neural net method by approximating the unknown nonlinear system through a feed-forward single hidden layer local neural net. Section 5.4 reports the empirical results of this chapter and a discussion of them. Section 5.5 contains some concluding remarks.

Most of this chapter is contained in the working paper written by Julio E. Sandubete, Lorenzo Escot and Simone Giannerini titled 'Quantifying chaos by various jacobian indirect methods'. It also appears in the working paper written by Julio E. Sandubete, Lorenzo Escot and Simone Giannerini titled 'A novel algorithm for estimating Lyapunov exponents: Local Neural Nets'.

5.1 Local vs Global jacobian indirect methods

As we have seen in the previous chapters there are two main methods in the literature that provide the estimated Lyapunov exponent from time-series data. The first one, the so-called *direct* approach which directly measures the growth rate of the divergence between two trajectories with an infinitesimal difference in their initial conditions (Section 3.3). As we have illustrated empirically the direct methods do not provide us consistent estimators and robustness to the presence of measurement errors. In addition, the asymptotic distribution of the direct estimator does not exist which means that it does not allow the building of formal tests, see Section 3.4. The second one, the so-called *indirect* approach (or jacobian-based method) which try to estimate the jacobian of the underlying generating system and then those partial derivatives are used to compute the Lyapunov exponent applying their analytical definition. These jacobian indirect methods solve all drawbacks which belong to the direct methods (Section 4.1).

Particularly, these indirect methods have the following advantages over all other methods: (*i*) its robustness to the presence of (small) noise as the measurement errors present in most real-world observed time-series data; (*ii*) their satisfactory performance in detecting existing non-linearities on time-series data of moderate sample sizes; (*iii*) allows the estimation of the full spectrum of Lyapunov exponents; (*iv*) the asymptotic distribution of the estimator can be derived allowing the building of formal tests. Having discussed the *global* jacobian indirect approach in the previous chapter we will now focus on greater detail into the *local* context of the jacobian indirect methods. In this section we are going to describe the key features of the local jacobian procedure for estimating the Lyapunov exponents from time-series data remarking the main differences with the global approach.

Keep in mind that any method for estimating the Lyapunov exponent from some observed time-series data are based previously on the state space reconstruction method (Section 3.2). Let us remember that we have assumed that the true data-generating process is unknown. Hence we do not have the advantage of observing directly the state of the system X_t let alone knowing the functional form f that generate the dynamic associated with it. Instead of that, there is an *observer function* $f : \mathbb{R}^2 \to \mathbb{R}$ that include an *additive measurement error* ε_t which generates observations as $x_t = f(X_t, \varepsilon_t)$ where ε_t is a sequence of independent and identically distributed random variables for $t = 1, 2, 3, \ldots, n$. Therefore it is assumed that all information available is the noise-contaminated sequence $\{x_t\}_{t=1}^{n}$ as a scalar strictly stationary univariate time serie, for a review see Section 3.1.

Then we form a sequence of delayed vectors by associating for each time period a vector in a reconstructed state space $\mathbb{R}^m$, whose coordinates are defined by eq.3.2.1 satisfying $\boldsymbol{x}_t^m = \left(x_t, x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-2)\tau}, x_{t-(m-1)\tau}\right)$ where m is the *embedding dimension* and τ is the *reconstruction time-delay* (or lag). Once we have got the delayed-coordinate embedding vectors from the time-series data we are going to focus on the next step. The procedure behind the *local jacobian indirect* method can be described as below.

Definition 5.1 (Local jacobian indirect method)**.** We have followed the results proposed by Gençay and Dechert (1992) considering a local context over the jth point for $j = 1, 2, \ldots, n$. We assume that there exist jth functions $G_j : \mathbb{R}^m \to \mathbb{R}^m$ such that the reconstructed dynamic system can be defined locally as $\boldsymbol{x}_t^m = G_j\left(\boldsymbol{x}_{t-\tau}^m\right)$ where $\boldsymbol{x}_t^m$ are the uniform delayed-coordinate embedding vectors defined by eq.3.2.1. Under the assumption that the embedding is a homeomorphism, the jth maps G_j are *topologically conjugate* to the unknown dynamic system F. This implies that certain dynamic properties of F and G_j are the same. In our case, the Lyapunov exponents obtained from F and G_j should be the same, so we can focus on estimating the exponents from the maps G_j. The dynamic systems G_j may be expressed as a matrix by

$$\begin{pmatrix} x_t \\ x_{t-\tau} \\ \vdots \\ x_{t-(m-2)\tau} \\ x_{t-(m-1)\tau} \end{pmatrix} = G_j \begin{pmatrix} x_{t-\tau} \\ x_{t-2\tau} \\ \vdots \\ x_{t-(m-1)\tau} \\ x_{t-m\tau} \end{pmatrix}$$

$$= \begin{pmatrix} \upsilon_j\left(x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-2)\tau}, x_{t-(m-1)\tau}, x_{t-m\tau}\right) \\ x_{t-\tau} \\ \vdots \\ x_{t-(m-2)\tau} \\ x_{t-(m-1)\tau} \end{pmatrix} \tag{5.1.1}$$

The local partial derivatives based on the jacobians corresponding to the reconstructed dynamic systems G_j will be obtained as follows

$$\partial G_j = \begin{pmatrix} \frac{\partial \upsilon_j}{\partial x_{t-\tau}} & \frac{\partial \upsilon_j}{\partial x_{t-2\tau}} & \frac{\partial \upsilon_j}{\partial x_{t-3\tau}} & \cdots & \frac{\partial \upsilon_j}{\partial x_{t-(m-1)\tau}} & \frac{\partial \upsilon_j}{\partial x_{t-m\tau}} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{5.1.2}$$

In this local context we would have as many jacobians as points along the whole trajectory considered. Instead, there would be only one in the global jacobian (eq.4.1.2). That is, it will have the same jacobian regardless of the point at which it is valued. We have to keep this fact in mind. Once we have seen the idea behind the local jacobian we have to focus on the last step. The ith *Lyapunov exponent estimator* can be obtained as follows.

Definition 5.2 (Lyapunov exponent estimator)**.** Let us consider the estimation of the local jacobians ∂G_j defined by eq.5.1.2. Once we have obtained them we are going to estimate the ith Lyapunov exponent (eq.2.3.6) given by the *Lyapunov exponent estimator* as *(local jacobian approach)*:

$$\hat{\lambda}_i = \lim_{M\to\infty} \frac{1}{M} \log \mu_i \left(\left| \partial \hat{G}_j^M \right| \right) \tag{5.1.3}$$

where μ_i is the ith largest eigenvalue provided by the local jacobians $\partial \hat{G}_j^M = \partial \hat{G}_M (x_M) \cdot \partial \hat{G}_{M-1} (x_{M-1}) \cdot \ldots \cdot \partial \hat{G}_1 (x_1)$ for $j = 1, 2, \ldots, M$; $i = 1, 2, 3, \ldots, m$ where M denote the block length and m is the embedding dimension.

Note that it is necessary to distinguish between the sample size n used for estimating the partial derivatives of the local jacobian in eq.5.1.2 and the *block length* M, defined in eq.5.1.3, which is the number of evaluation points (number of products of each jth jacobian) used for estimating the ith Lyapunov exponent, for a review see Section 4.2. The **DChaos** package allows the use of full sample estimation, but it also provides three different blocking methods for estimating the Lyapunov exponents, see table 4.2.1. Remember that we have developed the **DChaos** package which allows R users to choose in both context, global and local, between two different procedures to obtain the Lyapunov exponent estimator and four ways of subsampling: full sample, non-overlapping sample, equally spaced sample and bootstrap sample.

The asymptotic properties of the Lyapunov exponent estimator was derived by Shintani and Linton (2004). They provided a statistical framework for testing the hypothesis of chaos (eq.3.3.1) based on the Lyapunov exponent and the consistent estimator of its variance, for a review see Section 4.3. The asymptotic properties of the Lyapunov exponent estimator are *common* to all jacobian indirect methods irrespective of the choice of the approach used for estimating the function υ in the global jacobian (eq.4.1.2) or the functions υ_j in the local jacobian (eq.5.1.2). This is due to the fact that the Lyapunov exponent, which does not depend on the estimation method, will be dominant in the limit distribution see e.g., Whang and Linton (1999), Shintani and Linton (2003).

In this sense under the null hypothesis H_0 that the data-generating process is chaotic, the Lyapunov exponent estimator leads to asymptotically valid inferences in that the associated p-value follows a normal distribution on $N(0, \hat{\varphi}_i)$. Reject the null hypothesis $H_0 : \hat{\lambda}_i > 0$ means lack of chaotic behaviour. That is, the data-generating process does not have a chaotic attractor because of it does not show the initial-value sensitivity property. The **DChaos** package contains three functions discussed in the previous chapter `lyapunov`, `lyapunov.spec` and `lyapunov.max` which provide the standard error, the z test value and the p-value for testing the hypothesis of chaos (eq.3.3.1), see e.g., Example 4.4 or Example 4.5 to remember how it works.

Remark 5.1. We have illustrated how to obtain a consistent Lyapunov exponent estimator based on the robust estimation of the function υ in the global jacobian (eq.4.1.2) by neural net models, for a review see Section 4.1. In this case, we have to estimate the functions $\upsilon_j : \mathbb{R}^m \to \mathbb{R}$ provided by the local jacobian (eq.5.1.2). Traditionally the main contributions proposed by the scientific community regarding the local jacobian indirect methods have focused on local *linear* approaches. In this sense we are going to extend those results by proposing two novel alternatives that differ in the algorithm used for the estimation of those functions υ_j in the local jacobian. Particularly, we are going to introduce the following two local *nonlinear* approaches: (i) polynomial kernel method; (ii) neural net kernel method.

We will also provide the suitable algorithms in order to computes the local jacobian from the best-fitted polynomial kernel model and neural net kernel model respectively. In this local context there is no analogous functions in the three R packages considered in the previous chapters since they consider exclusively the direct method, see Section 3.3. We have developed our own algorithms for estimating the partial derivatives ∂G_j provided by the local jacobian defined in eq.5.1.2 considering both alternatives in order to test the hypothesis of chaos (eq.3.3.1) based on the Lyapunov exponent estimator (eq.5.1.3). Particularly, we provide five more functions which will be publicly available in a future extended version of the **DChaos** package, see table 5.1.1. We are going to illustrate all of them as we go on explaining the theoretical process step by step in the subsequent sections. Now let us focus first on the local nonlinear approach based on polynomial kernel models.

Function	Description
`locpol`	Fits any polynomial kernel model from time-series data
`locnet`	Fits any feedforward neural net kernel model from time-series data
`jacobian.locpol`	Computes the local jacobian from the best-fitted polynomial kernel model
`jacobian.locnet`	Computes the local jacobian from the best-fitted neural net kernel model
`w0.locnet`	Estimates the initial parameter vector of the neural net kernel model

Table 5.1.1: Functions based on two nonlinear local jacobian approaches.

5.2 Local jacobian indirect method: a polynomial kernel approach

Traditionally the main contributions proposed by the scientific community regarding the local jacobian indirect methods have focused on local linear approaches. The local linear estimator of the Lyapunov exponent was first proposed by Lu (1994), Lu and Smith (1997), and then revisited by Whang and Linton (1999) and Shintani and Linton (2003). Fan (1992), Fan and Gijbels (1992, 1996) remarked some significant advantages of this local linear estimator over the global estimator seen in the previous chapter based on neural net models: (*i*) it improves the fit over the global neural net estimator; (*ii*) it adapts automatically to the boundary of design points and therefore no boundary modification is needed; (*iii*) it is superior to the global neural net estimator in the evaluation of the system's derivatives at each point along the whole trajectory; (*iv*) it estimates the functions υ_j considering the condition of local linearity. The idea behind the local linear regression procedure can be summarising as follows, for a review see e.g., DiNardo and Tobias (2001).

The local linear approach for the estimation of the functions υ_j in the local jacobian (eq.5.1.2) is based on essentially the linear weighted least squares regression technique, where the data points farther away from an arbitrary point x_0 receive less weight than data points closer to x_0. As with density estimation, the weighting of the points is done through a kernel function K, and, as a consequence, this procedure is often called *kernel regression*. The choice of which kernel K to use for a given bandwidth $\hbar$, which amounts to the choice of how to diminish the weight of more distant points, has a relatively *small effect* on the result of a kernel regression. However, the choice of bandwidth $\hbar$ that controls the distance (or neighbourhood size) around the point x_0 has a *larger effect* on results. It also remains true that bandwidths $\hbar$ that are too large produce over-smoothed estimates while if it is too small the estimates appear excessively bumpy or under-smoothed, for a review see e.g., Fan and Gijbels (1996), Loader (2006).

In this sense there is a vast set of publications about methods for bandwidth choice in kernel regression estimation. For a review see e.g., Sheather and Jones (1991), Chiu (1991), Stoker (1993), Jones et al. (1996), Fan and Yao (2008) Heidenreich et al. (2013). As far as these local jacobian indirect methods are concerned the bandwidth $\hbar$ (together with the embedding parameters) will be determined by the statistical approach based on model selection procedures taking into account the Bayesian information criterion and leave one out cross-validation technique, for a review see Section 3.2. Now let us illustrate how we have obtained the Lyapunov exponent estimator (eq.5.1.3) based on the *pointwise estimates* of the functions υ_j in the local jacobian through a linear regression model as follows. Let us first begin considering an embedding dimension $m = 1$.

Algorithm 5.1 (Linear kernel approach)**.** Let $x_t = \upsilon_j(x_{t-\tau})$ be a sequence of delayed values obtained from the reconstructed dynamic system defined locally by eq.5.1.1 for $m = 1$. The *linear kernel estimator* can be obtained by approximating *linearly* the functions υ_j locally at each point j through the result provided by jth linear weighted least squares regressions as

$$\begin{aligned} x_t &= \upsilon_j(x_{t-\tau}) \\ &\approx \hat{\upsilon}_j = \hat{\beta}_{0,j} + \hat{\beta}_{1,j}(x_{t-\tau}) \end{aligned} \tag{5.2.1}$$

where $\hat{\beta}_{0,j}$ and $\hat{\beta}_{1,j}$ are the estimated coefficients for $j = 1, 2, \ldots, n$. Note that the issue of parameter estimation $(\hat{\beta}_{0,j}, \hat{\beta}_{1,j})$ is reduced to a linear weighted least squares problem in which the quantity to be minimized for each point j following DiNardo and Tobias (2001) is given by

$$\min_{\beta_{0,j},\beta_{1,j}} \sum_{t=1}^{n} \left[(x_t - (\beta_{0,j} + \beta_{1,j}(x_{t-\tau} - x_{j-\tau})))^2 \cdot K\left(\frac{x_{t-\tau} - x_{j-\tau}}{\hbar}\right)\right]$$

where $K(\cdot)$ is the kernel function and $\hbar$ denotes the bandwidth. As far as this doctoral dissertation is concerned we have considered the well-known Gaussian kernel expressed in this context by

$$K\left(\frac{x_{t-\tau} - x_{j-\tau}}{\hbar}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_{t-\tau} - x_{j-\tau}}{\hbar}\right)^2}$$

The bandwidth $\hbar$ will be determined by the statistical approach based on model selection procedures taking into account the Bayesian information criterion and leave one out cross-validation technique as it appears in the results of this chapter. Then we have obtained the partial derivatives provided by the local jacobian defined in eq.5.1.2 considering the estimated functions $\hat{\upsilon}_j$ defined locally by eq.5.2.1 at each point j as follows

$$\frac{\partial \hat{\upsilon}_j}{\partial x_{t-\tau}} = \hat{\beta}_{1,j} \tag{5.2.2}$$

where the estimated partial derivates $\partial \hat{G}_j$ based on the local jacobian defined in eq.5.1.2 for $m = 1$ are given by

$$\partial \hat{G}_j = \left(\frac{\partial \hat{\upsilon}_j}{\partial x_{t-\tau}}\right) = \hat{\beta}_{1,j} \tag{5.2.3}$$

Once we have obtained the estimated partial derivates $\partial\hat{G}_j$ the *local linear estimator* of the Lyapunov exponent (eq.2.3.6) can be defined as follows

$$\hat{\lambda} = \lim_{n\to\infty} \frac{1}{n} \log \mu \left(\left| \partial \hat{G}_j \right| \right) = \lim_{n\to\infty} \frac{1}{n} \log \mu \left(\left| \hat{\beta}_{1,j} \right| \right) = \lim_{n\to\infty} \frac{1}{n} \log \hat{\beta}_{1,j} \quad (5.2.4)$$

where μ is the largest eigenvalue provided by the estimated partial derivates $\partial\hat{G}_j$ defined by eq.5.2.3 for $j = 1, 2, \ldots, n$ (full sample size). There will only be just one Lyapunov exponent because the embedding dimension $m = 1$. Note that the local linear estimator of the Lyapunov exponent is reduced to the mean-log (mean of the logarithm) of the estimated coefficients $\hat{\beta}_{1,j}$ due to the eigenvalue of a number is the number itself. The local linear regressions estimate offer several improvements over local constant regressions including performance at the boundaries and it fits better as argued by Fan and Gijbels (1996). The result provided by the algorithm 5.1 is exactly the same as the local linear estimator called *m-step Lyapunov-like index* proposed by Yao and Tong (1994b). In this sense we have extended the results obtained by this local linear estimator considering an embedding dimension m greater than one. It can be described as following.

Algorithm 5.2 (Linear kernel approach extended)**.** Let consider a sequence of delayed values $x_t = \upsilon_j \left(x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-2)\tau}, x_{t-(m-1)\tau}, x_{t-m\tau} \right) = \upsilon_j \left(\boldsymbol{x}_{t-\tau}^m \right)$ obtained from the reconstructed dynamic system defined locally by eq.5.1.1 for $m \geqslant 1$. The *linear kernel estimator extended* can be obtained by approximating *linearly* the functions υ_j locally at each point j through the result provided by jth linear weighted least squares regressions as

$$\begin{aligned} x_t &= \upsilon_j \left(\boldsymbol{x}_{t-\tau}^m \right) \\ &\approx \hat{\upsilon}_j = \hat{\beta}_{0,j} + \hat{\boldsymbol{\beta}}_{1,j}^i \left(\boldsymbol{x}_{t-\tau}^m \right) \end{aligned} \quad (5.2.5)$$

where $\hat{\beta}_{0,j}$ and $\hat{\boldsymbol{\beta}}_{1,j}^i = \left(\hat{\beta}_{1,j}^1, \hat{\beta}_{1,j}^2, \ldots, \hat{\beta}_{1,j}^{m-2}, \hat{\beta}_{1,j}^{m-1}, \hat{\beta}_{1,j}^m \right)$ are the estimated coefficients for $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n-(m-1)\tau$. Note that the issue of parameter estimation $(\hat{\beta}_{0,j}, \hat{\boldsymbol{\beta}}_{1,j})$ is reduced again to a linear weighted least squares problem in which the quantity to be minimized for each point j is given by

$$\min_{\beta_{0,j}, \hat{\boldsymbol{\beta}}_{1,j}^i} \sum_{t=1}^{n-(m-1)\tau} \left[\left(x_t - \left(\beta_{0,j} + \hat{\boldsymbol{\beta}}_{1,j}^i \left(\boldsymbol{x}_{t-\tau}^m - \boldsymbol{x}_{j-\tau}^m \right) \right) \right)^2 \cdot K \left(\frac{\boldsymbol{x}_{t-\tau}^m - \boldsymbol{x}_{j-\tau}^m}{\hbar} \right) \right]$$

where $K(\cdot)$ is the kernel function and $\hbar$ denotes the bandwidth. We have considered again the Gaussian kernel expressed in this context by

$$K \left(\frac{\boldsymbol{x}_{t-\tau}^m - \boldsymbol{x}_{j-\tau}^m}{\hbar} \right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\boldsymbol{x}_{t-\tau}^m - \boldsymbol{x}_{j-\tau}^m}{\hbar} \right)^2}$$

The bandwidth $\hbar$ will be determined by the statistical approach based on model selection procedures taking into account the Bayesian information criterion and leave one out cross-validation technique as it appears in the results of this chapter. Then we have obtained the partial derivatives provided by the local jacobian defined in eq.5.1.2 considering the estimated functions $\hat{\upsilon}_j$ defined locally by eq.5.2.5 at each point j as follows

$$\frac{\partial \hat{\upsilon}_j}{\partial x_{t-i\tau}} = \hat{\boldsymbol{\beta}}^i_{1,j} \tag{5.2.6}$$

where $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n - (m-1)\tau$ and the estimated partial derivates $\partial \hat{G}_j$ based on the local jacobian defined in eq.5.1.2 for $m \geqslant 1$ are given by

$$\partial \hat{G}_j = \begin{pmatrix} \hat{\beta}^1_{1,j} & \hat{\beta}^2_{1,j} & \hat{\beta}^3_{1,j} & \cdots & \hat{\beta}^{m-1}_{1,j} & \hat{\beta}^m_{1,j} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{5.2.7}$$

Once we have obtained the estimated partial derivates $\partial \hat{G}_j$ the *local linear estimator* (extended) of the Lyapunov exponent (eq.2.3.6) can be defined as

$$\hat{\lambda}_i = \lim_{n\to\infty} \frac{1}{n} \log \mu_i \left(\left| \partial \hat{G}_j \right| \right) \tag{5.2.8}$$

where μ_i is the ith largest eigenvalue provided by the estimated partial derivates $\partial \hat{G}_j$ defined by eq.5.2.7. In this case, there will be as many Lyapunov exponents as embedding dimensions we consider. The local linear estimator (extended) of the Lyapunov exponent is not reduced directly to the mean-log of the estimated coefficients because in this case is not the eigenvalue of just one number but from a matrix of dimensions $m \times m$. The main drawbacks of these local linear estimators of the Lyapunov exponent are the following: (i) the estimated partial derivatives will be a constant for each embedding dimension considered. That is, it will have the same value regardless of the point at which it is valued since it will only depend on the values taken by the estimated coefficients; (ii) approximate linearly the functions υ_j locally at each point j through the result provided by linear weighted least squares regressions may not be the best approach since it should not be forgotten that chaotic behaviour is a phenomenon in nonlinear dynamic systems. It does not exist in linear systems. For these reasons we are going to extend those results by proposing two local *nonlinear* approaches which solve the disadvantages of the local linear estimators. Particularly, we are going to introduce the following alternatives: (i) polynomial kernel method; (ii) neural net kernel method. Let us focus first on the local nonlinear approach based on polynomial kernel models.

Algorithm 5.3 (Polynomial kernel approach)**.** We have considered a local polynomial of order p greater than one generalizing the *m-step Lyapunov-like index* estimator proposed by Yao and Tong (1994b) when the embedding dimension m is equal to one. Let $x_t = \upsilon_j(x_{t-\tau})$ be a sequence of delayed values obtained from the reconstructed dynamic system defined locally by eq.5.1.1 for $m = 1$. The *polynomial kernel estimator* can be obtained by approximating *nonlinearly* the functions υ_j locally at each point j through the result provided by jth nonlinear weighted least squares regressions as

$$\begin{aligned} x_t &= \upsilon_j(x_{t-\tau}) \\ &\approx \hat{\upsilon}_j = \hat{\beta}_{0,j} + \hat{\beta}_{1,j}(x_{t-\tau}) + \cdots + \hat{\beta}_{p,j}(x_{t-\tau})^p \end{aligned} \tag{5.2.9}$$

where $\hat{\beta}_{0,j}, \hat{\beta}_{1,j}, \ldots, \hat{\beta}_{p,j}$ are the estimated coefficients for $j = 1, 2, \ldots, n$. Note that the issue of parameter estimation $(\hat{\beta}_{0,j}, \hat{\beta}_{1,j}, \ldots, \hat{\beta}_{p,j})$ is reduced to a nonlinear weighted least squares problem in which the quantity to be minimized for each point j is given by

$$\min_{\beta_{0,j},\ldots,\beta_{p,j}} \sum_{t=1}^{n} \left[\left(x_t - (\beta_{0,j} + \cdots + \beta_{p,j}(x_{t-\tau} - x_{j-\tau})^p)\right)^2 \cdot K\left(\frac{x_{t-\tau} - x_{j-\tau}}{\hbar}\right)\right]$$

where $K(\cdot)$ is the kernel function and $\hbar$ denotes the bandwidth. We have considered again the Gaussian kernel expressed in this context by

$$K\left(\frac{x_{t-\tau} - x_{j-\tau}}{\hbar}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_{t-\tau} - x_{j-\tau}}{\hbar}\right)^2}$$

The bandwidth $\hbar$ will be determined by the statistical approach based on model selection procedures taking into account the Bayesian information criterion and leave one out cross-validation technique as it appears in the results of this chapter. Then we have obtained the partial derivatives provided by the local jacobian defined in eq.5.1.2 considering the estimated nonlinear functions $\hat{\upsilon}_j$ defined locally by eq.5.2.9 at each point j as follows

$$\frac{\partial \hat{\upsilon}_j}{\partial x_{t-\tau}} = \hat{\beta}_{1,j} + 2\hat{\beta}_{2,j}(x_{t-\tau}) + \cdots + p\hat{\beta}_{p,j}(x_{t-\tau})^{p-1} \tag{5.2.10}$$

where the estimated partial derivates $\partial \hat{G}_j$ based on the local jacobian defined in eq.5.1.2 for $m = 1$ are given by

$$\partial \hat{G}_j = \left(\frac{\partial \hat{\upsilon}_j}{\partial x_{t-\tau}}\right) = \hat{\beta}_{1,j} + 2\hat{\beta}_{2,j}(x_{t-\tau}) + \cdots + p\hat{\beta}_{p,j}(x_{t-\tau})^{p-1} \tag{5.2.11}$$

Once we have obtained the estimated partial derivates $\partial\hat{G}_j$ the *local polynomial estimator* of the Lyapunov exponent (eq.2.3.6) can be defined as

$$\hat{\lambda} = \lim_{n\to\infty} \frac{1}{n} \log\mu\left(\left|\partial\hat{G}_j\right|\right) = \lim_{n\to\infty} \frac{1}{n} \log\mu\left(\left|\hat{\beta}_{1,j} + \cdots + p\hat{\beta}_{p,j}(x_{t-\tau})^{p-1}\right|\right) \tag{5.2.12}$$

where μ is the largest eigenvalue provided by the estimated partial derivates $\partial\hat{G}_j$ defined by eq.5.2.11 for $j = 1, 2, \ldots, n$ (full sample size). There will only be just one Lyapunov exponent because the embedding dimension $m = 1$. In this particular case the local polynomial estimator of the Lyapunov exponent is reduced to the mean-log of the estimated coefficients multiplied by the sequence of delayed values $x_{t-\tau}$ as defined by eq.5.2.10 evaluated along the whole trajectory. Note that the estimated partial derivatives will have a different value depending on the point at which it is valued, since in the polynomial case it will depend on the values taken by the estimated coefficients and the sequence of delayed values evaluated along the whole trajectory for each point. This result is relevant in order to provide better fits to pointwise estimates in this context because by definition when estimating the Lyapunov exponents through the indirect methods we have to make an evaluation of the jacobian at each point along the whole trajectory. Now let us extend the results obtained by this local polynomial kernel estimator considering an embedding dimension m greater than one.

Algorithm 5.4 (Polynomial kernel approach extended)**.** Let consider $x_t = \upsilon_j\left(x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-2)\tau}, x_{t-(m-1)\tau}, x_{t-m\tau}\right) = \upsilon_j\left(\boldsymbol{x}^m_{t-\tau}\right)$ as a sequence of delayed values obtained from the reconstructed dynamic system defined locally by eq.5.1.1 for $m \geqslant 1$. The *polynomial kernel estimator extended* can be obtained by approximating *nonlinearly* the functions υ_j locally at each point j through the result provided by jth nonlinear weighted least squares regressions as follows

$$\begin{aligned} x_t &= \upsilon_j\left(\boldsymbol{x}^m_{t-\tau}\right) \\ &\approx \hat{\upsilon}_j = \hat{\beta}_{0,j} + \hat{\boldsymbol{\beta}}^i_{1,j}\left(\boldsymbol{x}^m_{t-\tau}\right) + \cdots + \hat{\boldsymbol{\beta}}^i_{p,j}\left(\boldsymbol{x}^m_{t-\tau}\right)^p \end{aligned} \tag{5.2.13}$$

where $\hat{\beta}_{0,j}, \hat{\boldsymbol{\beta}}^i_{1,j}, \ldots, \hat{\boldsymbol{\beta}}^i_{p,j} = \left(\hat{\beta}^1_{p,j}, \hat{\beta}^2_{p,j}, \ldots, \hat{\beta}^{m-2}_{p,j}, \hat{\beta}^{m-1}_{p,j}, \hat{\beta}^m_{p,j}\right)$ are the estimated coefficients for $p = 1, 2, \ldots$; $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n-(m-1)\tau$. Note that the issue of parameter estimation $(\hat{\beta}_{0,j}, \hat{\boldsymbol{\beta}}^i_{1,j}, \ldots, \hat{\boldsymbol{\beta}}^i_{p,j})$ is reduced again to a nonlinear weighted least squares problem in which the quantity to be minimized for each point j is given by

$$\min_{\beta_{0,j},\ldots,\boldsymbol{\beta}^i_{p,j}} \sum_{t=1}^{n-(m-1)\tau} \left[\left(x_t - \left(\beta_{0,j} + \cdots + \boldsymbol{\beta}^i_{p,j}\left(\boldsymbol{x}^m_{t-\tau} - \boldsymbol{x}^m_{j-\tau}\right)^p\right)\right)^2 \cdot K\left(\frac{\boldsymbol{x}^m_{t-\tau} - \boldsymbol{x}^m_{j-\tau}}{\hbar}\right)\right]$$

where $K(\cdot)$ is the kernel function and $\hbar$ denotes the bandwidth. We have considered again the Gaussian kernel expressed in this context by

$$K\left(\frac{\boldsymbol{x}_{t-\tau}^{m}-\boldsymbol{x}_{j-\tau}^{m}}{\hbar}\right)=\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{\boldsymbol{x}_{t-\tau}^{m}-\boldsymbol{x}_{j-\tau}^{m}}{\hbar}\right)^{2}}$$

The bandwidth $\hbar$ will be determined by the statistical approach based on model selection procedures taking into account the Bayesian information criterion and leave one out cross-validation technique as it appears in the results of this chapter. Then we have obtained the partial derivatives provided by the local jacobian defined in eq.5.1.2 applying the chain rule to the estimated nonlinear functions $\hat{\upsilon}_j$ defined locally by eq.5.2.13 at each point j considering a polynomial of order p, its derivatives and cross products. For instance, the quadratic case $p=2$ with an embedding dimension $m=2$ will be as follows

$$\frac{\partial\hat{\upsilon}_j}{\partial x_{t-i\tau}}=\ \hat{\boldsymbol{\beta}}_{1,j}^{i}+\sum_{k=2}^{p}k\cdot\hat{\boldsymbol{\beta}}_{k,j}^{i}(x_{t-i\tau})^{k-1}+\hat{\beta}_{12,j}\left(\boldsymbol{x}_{t-\tau}^{m}-x_{t-i\tau}\right)\qquad(5.2.14)$$

where $i=1,2;\ j=1,2,\ldots,n-(m-1)\tau$ and the estimated partial derivates $\partial\hat{G}_j$ based on the local jacobian defined in eq.5.1.2 for $m\geqslant 1$ are given by

$$\partial\hat{G}_j=\begin{pmatrix}\frac{\partial\hat{\upsilon}_j}{\partial x_{t-\tau}} & \frac{\partial\hat{\upsilon}_j}{\partial x_{t-2\tau}} & \frac{\partial\hat{\upsilon}_j}{\partial x_{t-3\tau}} & \cdots & \frac{\partial\hat{\upsilon}_j}{\partial x_{t-(m-1)\tau}} & \frac{\partial\hat{\upsilon}_j}{\partial x_{t-m\tau}}\\ 1 & 0 & 0 & \cdots & 0 & 0\\ 0 & 1 & 0 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & 1 & 0\end{pmatrix}\qquad(5.2.15)$$

Once we have obtained the estimated partial derivates $\partial\hat{G}_j$ the *local polynomial estimator* (extended) of the Lyapunov exponent is defined as

$$\hat{\lambda}_i=\lim_{n\to\infty}\frac{1}{n}\log\mu_i\left(\left|\partial\hat{G}_j\right|\right)\qquad(5.2.16)$$

where μ_i is the ith largest eigenvalue provided by the estimated partial derivates $\partial\hat{G}_j$ defined by eq.5.2.15. In this case, there will be as many Lyapunov exponents as embedding dimensions we consider. The local polynomial estimator (extended) of the Lyapunov exponent is not reduced directly to the mean-log of the estimated coefficients multiplied by the sequence of delayed values evaluated along the whole trajectory because in this case is not the eigenvalue of just one number but from a matrix of dimensions $m\times m$. Note that there is usually a preference for odd-order polynomial fits see e.g., Ruppert and Wand (1994), Fan and Gijbels (1996).

As far as this doctoral dissertation is concerned the main reason for using local polynomial kernel regressions is not to look for the best predictive model but to estimate a model that captures the nonlinear time dependence well enough and, additionally, allows us to obtain in an analytical way (instead of numerical) the local jacobians from the unknown data-generating process (eq.5.1.2). The robust estimation of these local jacobians will allow us to contrast consistently the hypothesis of chaos (eq.3.3.1). Now let us discuss the functions that will be available in a future extended version of the **DChaos** package in order to calculate analytically the partial derivatives from the best-fitted local polynomial kernel model.

The **DChaos** package will provide the function `locpol` which arguments are (`serie`, `m`, `lag`, `timelapse`, `order`, `trace`) for approximating *nonlinearly* the functions υ_j defined locally by eq.5.2.13 at each point j through the result provided by nonlinear weighted least squares regressions. We have considered the `lm` function that belongs to the **stats** package considering the parameter `weights` to fit any polynomial kernel model from time-series data. Note that the process of adjustment to a polynomial kernel model need to estimate the optimal bandwidth value. For this reason we have considered the `kedd` function together with the leave one out cross-validation technique and the well-known Gaussian kernel. Let us illustrate some examples considering time-series data simulated from the Logistic map with a chaotic behaviour ($\mu = 4$) contained in the **DChaos** package (`logistic.sim`). The command `set.seed(34)` will set the seed for reproducibility.

Example 5.1 (Best-fitted polynomial kernel models)**.** The `locpol` function returns several objects. The best-fitted polynomial kernel model is saved. It also contains some useful information about the best estimated coefficients found, the fitted values, the residuals obtained or the best embedding parameters set chosen. The best models are displayed on the console as we show below for `m` $\in \{1:2\}$, `lag` $\in \{1:1\}$ and `order` $\in \{1:4\}$. The first column is the model number, the second column is the embedding dimension, the third column is the lag or reconstruction time-delay considered, the fourth column is the order (or degree) of the polynomial kernel model, the fifth column is the optimal bandwidth value and the sixth column is the Bayesian Information Criterion (BIC) value corresponding to that parameter set. Notice that the polynomial kernel model are sorted from lowest to highest BIC values. In this case the best-fitted polynomial kernel model has the following parameter set values `m=1`, `lag=1` and `order=3`. The BIC criteria is given by

$$BIC = \log(RSS) + \frac{\log(n)}{n}\left[1 + coef\,(m+2)\right]$$

where RSS is the residual sum of squares, n is the number of observations, m is the embedding dimension and $coef$ is the number of coefficients considered in each weighted least squares regression.

```
## Simulates time-series data from the Logistic map with chaos
ts    <- DChaos::logistic.sim(a=4, n=1000)
[1] 7.47e-01 7.57e-01 7.37e-01 7.76e-01 6.95e-01 8.48e-01

## Provides the best-fitted polynomial kernel models
model <- DChaos::locpol(ts, m=1:2, lag=1:1, timelapse="FIXED",
         order=1:4)

  Best models:
  m lag order         bw       BIC
5 1   1     3 0.07272004 -7129.716
7 1   1     4 0.08550298 -7103.145
3 1   1     2 0.03213206 -7091.049
4 2   1     2 0.03661950 -7083.641
6 2   1     3 0.07960813 -7041.907
8 2   1     4 0.09160846 -7015.703
2 2   1     1 0.02711087 -1770.700
1 1   1     1 0.02189892 -1261.733
```

Example 5.2 (Partial derivatives are calculated analytically from the best–fitted polynomial kernel model)**.** The **DChaos** package provides the function `jacobian.locpol` which arguments are (`model`, `data`, `m`, `lag`, `timelapse`, `order`, `trace`) to compute the partial derivatives of the local jacobian in eq.5.2.15 from the best-fitted polynomial kernel model. This function returns some useful information about the partial derivatives calculated analytically, the best-fitted polynomial kernel model saved, the best estimated coefficients found, the fitted values, the residuals obtained or the best embedding parameters set and bandwidth chosen. It also allows the R user uses the data previously obtained from the best-fitted polynomial kernel model estimated by the `locpol` function if `model` is not empty. Otherwise `data` has to be specified. We provide below the first five partial derivatives values (`dx1`) corresponding to the best-fitted polynomial kernel model estimated above (model #5).

```
## Computes analytically the partial derivatives from the best
## fitted polynomial kernel model showed in the locpol example
jacobian <- DChaos::jacobian.locpol(model=model)
show(head(jacobian$jacobian))

          dx1
[1,]  3.889823
[2,]  3.565362
[3,]  2.355905
[4,] -1.224857
[5,] -3.249863
```

Example 5.3 (Partial derivatives are calculated analytically without setting previously any polynomial kernel model)**.** Now we are going to provide the results obtained from the function `jacobian.locpol` without setting the best-fitted polynomial kernel model estimated previously. We have considered the dataset simulated from the logistic map again. We have chosen the following parameter set values `m=3`, `lag=1`, `timelapse=FIXED`, `order=1:7`, `trace=1`. We show below the first seven partial derivatives values (`dx1`, `dx2`, `dx3`) corresponding to compute the partial derivatives of the jacobian following eq.5.2.15 from the best-fitted polynomial kernel (model #4).

```
## Partial derivatives are calculated analytically without
## setting previously any polynomial kernel model
jacobian <- DChaos::jacobian.locpol(data=ts, m=3:3, lag=1:1,
            timelapse="FIXED", order=1:7)
```

```
  Best models:
  m lag order         bw       BIC
4 3   1     4 0.03666687 -7222.558
5 3   1     5 0.07209342 -7029.127
7 3   1     7 0.08657706 -7000.285
6 3   1     6 0.07969632 -6986.525
3 3   1     3 0.03205978 -6504.087
2 3   1     2 0.02715546 -4998.156
1 3   1     1 0.04446699 -2866.692
```

```
show(head(jacobian$jacobian))
```

```
           dx1           dx2           dx3
[1,]  2.355912 -1.108206e-04  3.243993e-04
[2,] -1.224857 -1.479029e-09  4.302044e-09
[3,] -3.249863  1.121081e-08 -2.897358e-08
[4,]  1.280805 -3.232026e-09 -7.401307e-11
[5,] -3.179769  7.883640e-08  2.433850e-07
[6,]  1.055467  1.653099e-09 -2.207235e-10
[7,] -3.442995 -1.138271e-08 -3.872880e-08
```

Note that the local linear model (`order = 1`) reports the worst fit of the 7 polynomial kernel models estimated. It is exactly the same in the Example 5.1. These result would support the arguments mentioned above. The algorithms provide in this section for computing the local jacobian from the best-fitted polynomial kernel model will be publicly available in a future extended version of the **DChaos** package. Now let us move on to the second local nonlinear approach proposed in this chapter which is a class of kernel-type neural network estimator based on the local jacobian indirect methods.

5.3 Local jacobian indirect method: a neural net kernel approach

In this section we are going to propose another novel local nonlinear approach based on neural net kernel models. As far as we know the local neural net kernel estimator of the Lyapunov exponent has not been proposed by anyone so far. This approach has some significant advantages over the global neural net case seen in the previous chapter: (*i*) it improves the fit over the global neural net estimator; (*ii*) it adapts automatically to the boundary of design points and therefore no boundary modification is needed; (*iii*) it is superior to the global neural net estimator in the evaluation of the system?s derivatives at each point along the whole trajectory; (*iv*) it estimates the functions υ_i considering the condition of local nonlinearity. The idea behind the local neural net procedure can be summarising as follows. Note that it is similar to the case described in the previous section because both are based on nonlinear kernel regressions.

The neural net kernel approach for the estimation of the functions υ_j in the local jacobian (eq.5.1.2) is based essentially on pointwise estimates through a local standard feed-forward neural network with as few as one hidden layer instead of a local polynomial, where the data points farther away from an arbitrary point x_0 receive less weight than data points closer to x_0. The weighting of the points is done through a kernel function K, and, as a consequence, this procedure is often called kernel regression. The choice of which kernel K to use for a given bandwidth $\hbar$, which amounts to the choice of how to diminish the weight of more distant points, has a relatively *small effect* on the result of a kernel regression. However, the choice of bandwidth $\hbar$ that controls the distance (or neighbourhood size) around the point x_0 has a *larger effect* on results as we mentioned before.

As far as these local jacobian indirect methods are concerned the bandwidth $\hbar$ (together with the embedding parameters and the number of hidden units in the single hidden layer) will be determined by the statistical approach based on model selection procedures taking into account the Bayesian information criterion and leave one out cross-validation technique, for a review see Section 3.2. In this sense, the feed-forward networks with just one single hidden layer are a class of universal approximations following Hornik et al. (1989). Theoretically, neural nets are expected to perform better than other approximation methods, especially with high-dimensional models, since the approximation form is not so sensitive to the increasing dimension. Now let us illustrate how we have obtained the Lyapunov exponent estimator (eq.5.1.3) based on the pointwise estimates of the functions υ_j in the local jacobian through a local feed-forward single hidden layer network. We have developed our own algorithm as follows.

Algorithm 5.5 (Neural net kernel approach)**.** Let consider a sequence $x_t = \upsilon_j\left(x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-2)\tau}, x_{t-(m-1)\tau}, x_{t-m\tau}\right) = \upsilon_j\left(\boldsymbol{x}_{t-\tau}^m\right)$ of delayed values obtained from the reconstructed dynamic system defined locally by equation 5.1.1 for $m \geqslant 1$. The *neural net kernel estimator* can be obtained by approximating *nonlinearly* the functions υ_j locally at each point j through the result provided by jth feed-forward single hidden layer neural networks with a single output by

$$\begin{aligned} x_t &= \upsilon_j\left(\boldsymbol{x}_{t-\tau}^m\right) \\ &\approx \hat{\upsilon}_j = \Phi_0\left[\hat{\alpha}_{0,j} + \sum_{q=1}^{h} \hat{\omega}_{qo,j}\Phi_q\left(\hat{\alpha}_{q,j} + \sum_{i=1}^{m} \hat{\omega}_{iq,j} x_{t-i\tau}\right)\right] \end{aligned} \tag{5.3.1}$$

where $\Phi_0 \in I$, $\hat{\alpha}_{0,j}$ is the estimated network coefficients from input at each point j, the number of neurones (or nodes) in the single hidden layer is denoted by h, $\hat{\omega}_{q0,j}$ are the estimated layers connection weights from input to hidden layer at each point j, Φ_q is the transfer function which in our case is the logistic function, $\hat{\alpha}_{q,j}$ is the estimated network coefficients from hidden layer at each point j, m is the embedding dimension and $\hat{\omega}_{iq,j}$ are the estimated layers connection weights from hidden layer to output at each point j. The issue of parameter estimation is reduced to a nonlinear least squares problem in which the quantity to be minimized for each point j is given by

$$\sum_{t=1}^{n-(m-1)\tau}\left[\left(x_t - \left(\alpha_{0,j} + \sum_{q=1}^{h}\omega_{qo,j}\Phi_q\left(\alpha_{q,j} + \sum_{i=1}^{m}\omega_{iq,j}\left(x_{t-i\tau} - x_{j-i\tau}\right)\right)\right)\right)^2 \cdot K\left(\frac{\boldsymbol{x}_{t-\tau}^m - \boldsymbol{x}_{j-\tau}^m}{\hbar}\right)\right]$$

where $K(\cdot)$ is the kernel function and $\hbar$ denotes the bandwidth. We have considered again the Gaussian kernel expressed in this context by

$$K\left(\frac{\boldsymbol{x}_{t-\tau}^m - \boldsymbol{x}_{j-\tau}^m}{\hbar}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\boldsymbol{x}_{t-\tau}^m - \boldsymbol{x}_{j-\tau}^m}{\hbar}\right)^2}$$

The bandwidth $\hbar$ will be determined by the statistical approach based on model selection procedures taking into account the Bayesian information criterion and leave one out cross-validation technique as it appears in the results of this chapter. Then we have obtained the partial derivatives provided by the local jacobian defined in eq.5.1.2 applying the chain rule to eq.5.3.1 as follows

$$\frac{\partial \hat{\upsilon}_j}{\partial x_{t-i\tau}} = \Phi'_0\left(z_{0,j}\right)\sum_{q=1}^{h}\hat{\omega}_{q0,j}\Phi'_0\left(z_{q,j}\right)\hat{\omega}_{iq,j} \tag{5.3.2}$$

where

$$z_{0,j} = \hat{\alpha}_{0,j} + \sum_{q=1}^{h} \hat{\omega}_{q0,j} \Phi_q \left(z_{q,j} \right), \quad z_{q,j} = \hat{\alpha}_{q,j} + \sum_{i=1}^{m} \hat{\omega}_{iq,j} x_{t-i\tau}$$

where $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n - (m-1)\tau$ and the estimated partial derivates $\partial \hat{G}_j$ based on the local jacobian defined in eq.5.1.2 for $m \geqslant 1$ are given by

$$\partial \hat{G}_j = \begin{pmatrix} \frac{\partial \hat{v}_j}{\partial x_{t-\tau}} & \frac{\partial \hat{v}_j}{\partial x_{t-2\tau}} & \frac{\partial \hat{v}_j}{\partial x_{t-3\tau}} & \cdots & \frac{\partial \hat{v}_j}{\partial x_{t-(m-1)\tau}} & \frac{\partial \hat{v}_j}{\partial x_{t-m\tau}} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{5.3.3}$$

Once we have obtained the estimated partial derivates $\partial \hat{G}_j$ the *local neural net estimator* of the Lyapunov exponent is defined as

$$\hat{\lambda}_i = \lim_{n \to \infty} \frac{1}{n} \log \mu_i \left(\left| \partial \hat{G}_j \right| \right) \tag{5.3.4}$$

where μ_i is the ith largest eigenvalue provided by the estimated partial derivates $\partial \hat{G}_j$ defined by eq.5.3.3. In this case, there will be as many Lyapunov exponents as embedding dimensions we consider. The local neural net estimator of the Lyapunov exponent is not reduced directly to the mean-log of the estimated coefficients multiplied by the sequence of delayed values evaluated along the whole trajectory because in this case is not the eigenvalue of just one number but from a matrix of dimensions $m \times m$. We recommend reviewing section 4.1 for a comparison with the neural network approach in the global jacobian context.

The main difference from the global procedure is that now pointwise estimates will have to be made. That is, there will be as many neural nets (and jacobians) as points, having to evaluate them along the whole trajectory. In this case, the kernel function and an optimal bandwidth will have to be taken into account when obtaining estimates of the coefficients. Note that again as with polynomial kernel models the main reason for using local neural net kernel regressions is not to look for the best predictive model but to estimate a model that captures the nonlinear time dependence well enough and, additionally, allows us to obtain in an analytical way (instead of numerical) the local jacobians from the unknown data-generating process (eq.5.1.2). The robust estimation of these local jacobians will allow us to contrast consistently the hypothesis of chaos (eq.3.3.1).

Now let us discuss the functions that will be available in a future extended version of the **DChaos** package in order to calculate analytically the partial derivatives from the best-fitted local neural net kernel model. The **DChaos** package will provide the function `locnet` which arguments are (`serie`, `m`, `lag`, `timelapse`, `h`, `w0maxit`, `wtsmaxit`, `pre.white`, `trace`, `seed.t`, `seed`) for approximating *nonlinearly* the functions v_j defined locally by eq.5.3.1 at each point j through the result provided by nonlinear neural network regressions. We have considered the `nnet` function that belongs to the **nnet** package to fit the neural net models. Note that the process of adjustment to a neural network often suffers from being trapped in local optima and different initialization strategies should be taken into account.

For this reason we have implemented the function `w0.locnet` which arguments are (`x`, `y`, `m`, `h`, `rangx`, `kwts`, `w0maxit`, `seed.t`, `seed`). This function estimates the initial parameter vector of the neural net kernel models being able to set the maximum number of iterations that we want to obtain setting `w0maxit`. In addition, by default the neural net estimation is initialized with a fixed seed denoted by `seed.t=TRUE` with a value equal to `seed=56666459`. The R users can let the seed be fixed either randomly by `seed.t=FALSE` or even fix other value of the seed to be able to replicate the results obtained. Let us illustrate some examples considering time-series data simulated from the Logistic map with a chaotic behaviour ($\mu = 4$) contained in the **DChaos** package. The command `set.seed(34)` will set the seed for reproducibility.

Example 5.4 (Best-fitted neural net kernel models)**.** The `locnet` function returns several objects. The best-fitted local feed-forward single hidden layer neural net model is saved. It also contains some useful information about the best set of weights found, the fitted values, the residuals obtained or the best embedding parameters and bandwidth chosen. The best models are displayed on the console as we show below for `m` $\in \{1:2\}$, `lag` $\in \{1:1\}$ and `h` $\in \{2:5\}$. The first column is the neural net number, the second column is the embedding dimension, the third column is the lag or reconstruction time-delay considered, the fourth column is the number of neurones (or nodes) in the single hidden layer, the fifth column is the optimal bandwidth value and the sixth column is the Bayesian Information Criterion (BIC) value corresponding to that parameter set. Notice that the neural net kernel models are sorted from lowest to highest BIC values. In this case the best-fitted neural net kernel model has the following parameter set values `m=1`, `lag=1` and `h=4`. The BIC criteria is given by

$$BIC = \log(RSS) + \frac{\log(n)}{n}\left[1 + h\,(m+2)\right]$$

where RSS is the residual sum of squares, n is the number of observations, m is the embedding dimension and h is the number of neurones (or nodes) used in the single hidden layer as noted above.

```
## Simulates time-series data from the Logistic map with chaos
ts    <- DChaos::logistic.sim(a=4, n=1000)
[1] 7.47e-01 7.57e-01 7.37e-01 7.76e-01 6.95e-01 8.48e-01

## Provides the best-fitted neural net kernel models
model <- DChaos::locnet(ts, m=1:2, lag=1:1, timelapse="FIXED",
         h=2:5)

 Best models:
  m lag h         bw       BIC
5 1   1 4 0.07272004 -1399.491
2 2   1 2 0.02711087 -1346.217
1 1   1 2 0.02189892 -1313.243
7 1   1 5 0.08550298 -1312.417
8 2   1 5 0.09160846 -1289.765
4 2   1 3 0.03661950 -1264.586
3 1   1 3 0.03213206 -1255.138
6 2   1 4 0.07960813 -1247.401
```

Example 5.5 (Partial derivatives are calculated analytically from the best–fitted neural net kernel model)**.** The **DChaos** package provides the function `jacobian.locnet` which arguments are (`model`, `data`, `m`, `lag`, `timelapse`, `h`, `w0maxit`, `wtsmaxit`, `pre.white`, `trace`, `seed.t`, `seed`) to compute the partial derivatives of the local jacobian in eq.5.3.3 from the best-fitted neural net kernel model. This function returns some useful information about the partial derivatives calculated analytically, the best-fitted neural kernel model saved, the best estimated coefficients found, the fitted values, the residuals obtained or the best embedding parameters set and bandwidth chosen. It also allows the R user uses the data previously obtained from the best-fitted neural kernel model estimated by the `locnet` function if `model` is not empty. Otherwise `data` has to be specified. We provide below the first five partial derivatives values (`dx1`) corresponding to the best-fitted neural net kernel model estimated above (model #5).

```
## Computes analytically the partial derivatives from the best
## fitted neural net kernel model showed in the locnet example
jacobian <- DChaos::jacobian.locnet(model=model)
show(head(jacobian$jacobian))

           dx1
[1,]  3.861380
[2,]  3.553705
[3,]  2.341330
[4,] -1.166301
[5,] -3.230116
```

Example 5.6 (Partial derivatives are calculated analytically without setting previously any neural net kernel model)**.** Now we are going to provide the results obtained from the function `jacobian.locnet` without setting the best-fitted neural net kernel model estimated previously. We have considered the dataset simulated from the logistic map again. We have chosen the following parameter set values `m=3`, `lag=1`, `timelapse=FIXED`, `h=2:8`. We show below the first seven partial derivatives values (`dx1`, `dx2`, `dx3`) corresponding to compute the partial derivatives of the jacobian following eq.5.3.3 from the best-fitted neural net kernel (model #2).

```
## Partial derivatives are calculated analytically without
## setting previously any neural net kernel model
jacobian <- DChaos::jacobian.locnet(data=ts, m=3:3, lag=1:1,
            timelapse="FIXED", h=2:8)

 Best models:
  m lag h         bw       BIC
2 3   1 3 0.02715546 -1359.728
5 3   1 6 0.07209342 -1307.186
3 3   1 4 0.03205978 -1306.432
1 3   1 2 0.04446699 -1295.486
4 3   1 5 0.03666687 -1290.409
6 3   1 7 0.07969632 -1260.583

show(head(jacobian$jacobian))

            dx1         dx2        dx3
[1,]  2.0803901  0.99704217  0.1955900
[2,] -0.5371118 -0.84208032 -1.2894434
[3,] -1.4126468  1.75168139  1.0928212
[4,]  1.2388249  0.09050501  0.3848790
[5,] -1.2902753 -2.16803985  0.7238846
[6,]  1.2777432  0.29183047 -0.4119059
[7,] -1.9627564 -1.27507940  0.8574443
```

Note that in Example 5.5 the partial derivatives obtained are very similar to those provided by the polynomial kernel model (Example 5.2). It makes sense since the embedding dimension and the lag for the best-fitted model are the same in both cases. This result shows that given the same embedding parameters we would get a similar approximation whether we choose a cubic polynomial kernel model or a local feed-forward single hidden layer neural net model with 7 nodes in the hidden layer. The algorithms provided in this section will be publicly available in a future extended version of the **DChaos** package. Now let us focus on the main empirical results of this chapter.

5.4 Data analysis and simulations

In this section we will focus on studying the robustness of the algorithms available in the **DChaos** package for estimating the Lyapunov exponent extending it to the local jacobian indirect methods proposed in this chapter. We have considered some noise-contaminated time-series data from four well-known chaotic dynamic systems. The Logistic map, the Gauss map, the Hénon map (in discrete-time) and the Rössler system (in continuous-time), for a review see table 2.4.1. We have added to each time-series data a normal multinomial error term denoted by $\varepsilon_t \sim N(0, s)$ with different variance values s. We think it is appropriate to add a measurement noise term because most real-world time-series data are usually noise-contaminated signals.

We are interested to know if local jacobian indirect methods report better estimates than direct methods and global jacobian indirect methods and if so, we want to contrast if the local jacobian indirect methods provide us (or not) consistent Lyapunov exponent estimators and robustness to the presence of measurement errors. In this sense we want to know if as the measurement noise increases, the error committed in obtaining the estimator is amplified with the consequent inaccuracy and inconsistency or instead it is reduced.

For this purpose we have compared the results provided by the traditional direct methods considering the **tseriesChaos** package and the **nonlinearTseries** package (see Section 3.3) with those of the **DChaos** package. In this case, we have considered three indirect methods: one based on the global jacobian (neural net approach) and two based on the local jacobian (polynomial kernel approach and neural net kernel approach). The algorithms considered in this empirical analysis for estimating the Lyapunov exponents are the following. The **tseriesChaos** package includes the function `lyap.k`, the **nonlinearTseries** package has a function called `maxLyapunov`; `lyapunov.max` and `lyapunov.spec` are the functions that belong to the **DChaos** package.

The command `set.seed(34)` will set the seed for reproducibility. To save CPU time we have set the embedding dimension $1 \leqslant m \leqslant 7$, the time-delay $\tau = 1$, the number of neurones (or nodes) in the single hidden layer $2 \leqslant h \leqslant 10$, the order (or degree) in the polynomial kernel model $1 \leqslant p \leqslant 4$ and the length of all time-series data is $n = 1000$. The R packages which implement exclusively the direct method proposed by Kantz (1994) use heuristic approaches and prescriptions for estimating the embedding parameters that mostly rely on physical or geometrical arguments. The jacobian indirect methods provided by the **DChaos** package uses by default the statistical approach based on model selection procedures, for a review see Section 3.2. In this case, we are going to use the statistical approach taking into account the Bayesian information criterion together with the leave one out cross-validation technique.

Logistic map	$s = 0$	$s = 0.01$	$s = 0.02$	$s = 0.03$	$s = 0.04$	$s = 0.05$
D1 direct method	0.0001220	0.0056643	0.0030120	0.003006	0.0033485	0.0030913
D2 direct method	0.0802315	0.4765133	0.4814125	0.4815446	0.4790305	0.4830895
N2 indirect method (GNN)	0.0000324	0.0000382	0.0000691	0.0000994	0.0001314	0.0001532
N2 indirect method (LPO)	0.0000134	0.0000150	0.0000223	0.0000376	0.0000488	0.0000564
N2 indirect method (LNN)	0.0000124	0.0000157	0.0000204	0.0000369	0.0000433	0.0000505
QR indirect method (GNN)	0.0000331	0.0000348	0.0000672	0.0000986	0.0000997	0.0001124
QR indirect method (LPO)	0.0000152	0.0000197	0.0000264	0.0000359	0.0000422	0.0000516
QR indirect method (LNN)	0.0000140	0.0000178	0.0000239	0.0000330	0.0000411	0.0000487
Gauss map						
D1 direct method	0.0005270	0.0111180	0.0205349	0.0293853	0.0275621	0.0336681
D2 direct method	0.0474216	0.1480353	0.1477251	0.1464405	0.1481204	0.1476371
N2 indirect method (GNN)	0.0000436	0.0000526	0.0000555	0.0000678	0.0000719	0.0000944
N2 indirect method (LPO)	0.0000170	0.0000199	0.0000247	0.0000388	0.0000460	0.0000507
N2 indirect method (LNN)	0.0000137	0.0000168	0.0000190	0.0000246	0.0000312	0.0000438
QR indirect method (GNN)	0.0000618	0.0000656	0.0000672	0.0000782	0.0000817	0.0000924
QR indirect method (LPO)	0.0000176	0.0000197	0.0000271	0.0000365	0.0000479	0.0000502
QR indirect method (LNN)	0.0000153	0.0000185	0.0000199	0.0000244	0.0000312	0.0000443

Table 5.4.1: One-dimensional chaotic dynamic systems. The mean square error (MSE) values based on the estimation of the largest Lyapunov exponent from direct methods **tseriesChaos** (D1) and **nonlinearTseries** (D2) are showed. Those obtained by the jacobian indirect methods through the **DChaos** library are provided: global neural net models (GNN), local polynomial kernel models (LPO) and local neural net kernel models (LNN).

In this empirical analysis we have estimated the following models in order to obtain the results shown in tables 5.4.1-5.4.2: (*i*) 63 different global neural net models have been estimated from each 24000 simulated series; (*ii*) 28000 different local polynomial kernel models have been estimated from each 24000 simulated series; (*iii*) 63000 different local neural net kernel models have been estimated from each 24000 simulated series. Then the neural net models, the polynomial kernel models and the neural net kernel models are sorted from lowest to highest BIC values. After that the best-fitted models with a lower BIC value and their associated embedding parameters are considered for estimating the 24000 largest Lyapunov exponents. Finally the mean square error (MSE) are calculated between the theoretical value (see Section 2.4) and the value estimated. To do that, we have used the Monte Carlo method. We have done 1000 repetitions by different initial conditions when simulating the time-series data from the four dynamic systems and six measurement noise levels considered. We have taken into account only the bootstrap blocking method and the number of bootstrap iterations is $B = 1000$. The MSE values are shown based on the estimation of the largest Lyapunov exponent from the direct methods **tseriesChaos** (D1), **nonlinearTseries** (D2) and the jacobian indirect methods: global neural net models (GNN), local polynomial kernel models (LPO) and local neural net kernel models (LNN). Those obtained by the jacobian indirect methods through the **DChaos** library are denoted by N2 (`lyapunov.max`) and QR (`lyapunov.spec`) regarding the Norma-2 and QR decomposition procedures respectively.

The data shown in tables 5.4.1-5.4.2 provide the following comments. First, we can remark that our algorithms are robust to the presence of (small) measurement errors because the results obtained are comparable to those which are noise free ($s = 0$). Although as the noise increases the error committed increases but it is not proportional in any case. Second, the indirect methods provide better estimates than direct methods in all the experiments we have conducted. The algorithms proposed by the **tseriesChaos** package behave better than those of the **nonlinearTseries** package. Between the two methods available in the **DChaos** package we do not observe significant differences. Third, local jacobian indirect methods report better estimates than direct methods and global jacobian indirect methods. Both local nonlinear approaches behave similarly although local neural net kernel models are somewhat better than local polynomial kernel models. Fourth, we have only focused on the largest Lyapunov exponent as direct methods do not estimate the full spectrum. Fifth, the direct methods are surely less flexible and robust but much faster and still informative. The jacobian indirect methods seems to perform well for every noisy time-series data. The price we have to pay is a greater computational complexity from two points of view, the computing time and the tuning parameters. This fact has a greater effect on local jacobian indirect methods than might be expected.

Hénon system	$s = 0$	$s = 0.01$	$s = 0.02$	$s = 0.03$	$s = 0.04$	$s = 0.05$
D1 direct method	0.0005650	0.0067221	0.0092761	0.0100339	0.0141379	0.0189926
D2 direct method	0.0121588	0.3133259	0.3145991	0.3115671	0.3226997	0.3178221
N2 indirect method (GNN)	0.0000365	0.0000486	0.0000635	0.0000761	0.0000899	0.0000917
N2 indirect method (LPO)	0.0000122	0.0000254	0.0000288	0.0000339	0.0000426	0.0000522
N2 indirect method (LNN)	0.0000104	0.0000215	0.0000249	0.0000314	0.0000408	0.0000513
QR indirect method (GNN)	0.0000318	0.0000451	0.0000589	0.0000601	0.0000866	0.0000932
QR indirect method (LPO)	0.0000156	0.0000237	0.0000296	0.0000349	0.0000452	0.0000567
QR indirect method (LNN)	0.0000130	0.0000213	0.0000284	0.0000332	0.0000428	0.0000541
Rössler system						
D1 direct method	0.0004471	0.0049521	0.0063189	0.0072719	0.0127326	0.0174911
D2 direct method	0.0398841	0.6412752	0.6388524	0.6396631	0.6451333	0.6499127
N2 indirect method (GNN)	0.0002477	0.0003529	0.0005997	0.0006122	0.0009521	0.0019947
N2 indirect method (LPO)	0.0001722	0.0001921	0.0002390	0.0002789	0.0003141	0.0003667
N2 indirect method (LNN)	0.0001433	0.0001581	0.0002209	0.0002644	0.0003056	0.0003371
QR indirect method (GNN)	0.0003168	0.0004891	0.0006070	0.0007155	0.0008190	0.0009268
QR indirect method (LPO)	0.0001631	0.0001993	0.0002106	0.0002558	0.0003219	0.0003867
QR indirect method (LNN)	0.0001542	0.0001766	0.0002004	0.0002275	0.0003102	0.0003347

Table 5.4.2: Multidimensional chaotic dynamic systems. The mean square error (MSE) values based on the estimation of the largest Lyapunov exponent from direct methods **tseriesChaos** (D1) and **nonlinearTseries** (D2) are showed. Those obtained by the jacobian indirect methods through the **DChaos** library are provided: global neural net models (GNN), local polynomial kernel models (LPO) and local neural net kernel models (LNN).

5.5 Discussion

In this chapter we have provided a summary of the main concepts and definitions related to two new local jacobian indirect methods for estimating the Lyapunov exponents from time-series data. These jacobian indirect methods solve all drawbacks which belong to the direct methods. We have provided a discussion about the right procedure to obtain the partial derivatives based on the local jacobian approach. In this sense, we have highlighted the main differences with the global jacobian approach. We have also illustrated how to get a consistent estimator of the Lyapunov exponent based on the local jacobian approach. Note that the asymptotic properties of the Lyapunov exponent estimator are common to all ijacobian indirect methods irrespective of the choice of the approach used for estimating the function υ in the global jacobian (eq.4.1.2) or the functions υ_j in the local jacobian (eq.5.1.2). This is due to the fact that the Lyapunov exponent, which does not depend on the estimation method, will be dominant in the limit distribution. Thus we have used these results to calculate the standard error of the Lyapunov exponent estimator and investigate the statistical significance of the sign of the exponents considering the local jacobian context.

Traditionally the main contributions proposed by the scientific community regarding the local jacobian indirect methods have focused on local linear approaches. The main drawbacks of these local linear estimators of the Lyapunov exponent are the following: (i) the estimated partial derivatives will be a constant for each embedding dimension considered. That is, it will have the same value regardless of the point at which it is valued since it will only depend on the values taken by the estimated coefficients; (ii) approximate linearly the functions υ_j locally at each point j through the result provided by linear weighted least squares regressions may not be the best approach since it should not be forgotten that chaotic behaviour is a phenomenon in nonlinear dynamic systems. It does not exist in linear systems. For these reasons we have extended those results by proposing two novel alternatives which solve the disadvantages of the local linear estimators.

Particularly, we have proposed the following two local nonlinear approaches: (i) polynomial kernel method; (ii) neural net kernel method. We have also provided the suitable algorithms in order to computes the local jacobian from the best-fitted polynomial kernel model and neural net kernel model respectively. These procedures has been illustrated by several examples. Remember that in this local context there is no analogous functions in the R packages considered in the previous chapters since they consider exclusively the direct method. We have developed our own algorithms which will be publicly available in a future extended version of the **DChaos** package considering both alternatives in order to test the hypothesis of chaos based on the Lyapunov exponent estimator in a local context.

Then we have focused on studying the robustness of the algorithms available in the **DChaos** package for estimating the Lyapunov exponent extending it to the local jacobian indirect methods proposed in this chapter. For this purpose we have compared the results provided by the traditional direct methods with those of the jacobian indirect methods based on three nonlinear approaches. In this sense we can remark that the indirect methods provide better estimates than direct methods in all the experiments we have conducted. In particular, the local jacobian indirect methods report better estimates than direct methods and global jacobian indirect methods. Both local nonlinear approaches behave similarly although local neural net kernel models are somewhat better than local polynomial kernel models.

Finally we have shown empirically that our algorithms are robust to the presence of (small) measurement errors because the results obtained are comparable to those which are noise free. Although as the noise increases the error committed increases but it is not proportional in any case. The jacobian indirect methods seems to perform well for every noisy time-series data. The price we have to pay is a greater computational complexity from two points of view, the computing time and the tuning parameters. This fact has a greater effect on local jacobian indirect methods than might be expected. Now let us move on to an applied data analysis based on different real-world financial time-series data which are usually noise-contaminated signals.

Chapter 6

Estimating Lyapunov exponents from non-uniform time frequency data-generating process

"Si mi voz muriera en tierra, llevadla al nivel del mar.
Sobre el corazón un ancla, sobre el ancla una estrella,
sobre la estrella el viento y sobre el viento la vela!".

— Rafael Alberti

Abstract: This chapter discusses an application to certain real-world financial time-series data which are characterised by its almost continuity on trade showing a variable time-frequency between each quote. Section 6.1 gives a possible explanation to the chaos model-data paradox in finance. Section 6.2 illustrates the correct procedure for the reconstruction of the state space from time-series data with non-uniform time-frequency. It also present an algorithm to get the non-uniform delayed-coordinate embedding vectors. Section 6.3 provides an overview about the theoretical framework for estimating the Lyapunov exponents from tick-by-tick time-serie data. Section 6.4 reports the empirical results of this chapter and a discussion of them. Section 6.5 contains some concluding remarks.

Most of this chapter is contained in the paper written by Julio E. Sandubete and Lorenzo Escot titled 'Chaotic signals inside some tick-by-tick financial time series' (published in *Chaos, Solitons & Fractals*). It has been also presented at the 11th International Conference on Chaotic Modeling and Simulation on June 5-8, 2018.

6.1 Financial tick-by-tick time-series data and chaos paradox

In this chapter we will discuss an application to real-world financial time-series data. As we have remarked at the beginning of this PhD thesis *chaos* is an ubiquitous phenomena which is produced everywhere and can be observed from quantum to cosmological scale see e.g., D'Alessio et al. (2016), Dittrich et al. (2017), Hanada et al. (2018). Particularly, in economics, it has been more than three decades since ideas from chaos began appearing in the literature showing that it is possible to design economic models in regime of chaotic behaviour from a theoretical point of view. Let us recommend some books and articles (and references therein) for anyone who wants to start getting involved in the *chaotic dynamic economics* theory; Brock et al. (1991), Medio (1992), Fernández-Díaz (1994), Grau-Carles (1996), Escot (2000), Shone (2002), Zhang (2006), Gandolfo (2009), Puu (2013), Faggini (2014), Bruno et al. (2016), Poitras (2018), Fernández-Díaz (2019).

However there is no clear evidence that economic time series behave chaotically. So far researchers have found substantial evidence for nonlinearity but relatively weak evidence for chaos see e.g., LeBaron (1994), Bask (1996), Barnett et al. (1997), Grau-Carles (2000), Fernández-Dıaz et al. (2002), Shintani and Linton (2003), Serletis and Shintani (2006), Bask et al. (2007), Olmedo (2011), Bensaïda and Litimi (2013), Faggini and Parziale (2016), Lahmiri (2017), Lahmiri and Bekiros (2018). Hence in this chapter our interest would be to check why it is usually difficult to detect chaos in economic time series whereas it is apparently easy to show a chaotic behaviour in many theoretical models. This phenomena was called the *chaos model-data paradox* by Brock et al. (1997).

The greatest difficulty in applying chaos theory in empirical studies in economics is related to the quantity and quality of dataset. Unlike experimental sciences where one can carry out experiments and simulations in the laboratory obtaining a large and clean dataset, most economic time series consist of daily, weekly, monthly, quarterly or annual noise contaminated data. The main reason why economic data traditionally were low frequency and discrete was the cost of collection and analysis. Particularly, in financial markets, the emergence of new technologies have had important implications for both the quantity and quality of existing data in finance. Nowadays trades realized by traders are replaced by algorithmic trades executed in an automated way providing a huge amount of data over many intra-daily disaggregated time intervals.

While it is natural to think that with respect to data more is always better, this statement needs to be considered carefully in certain environments. High-frequency financial dataset are still expensive to collect, maintain, and manipulate. These observations are also subject to a wide range of idiosyncratic factors such as non-synchronous trading, intra-day seasonal effects, measurements errors due to bid-ask spreads and even conceptual problems, such as defining a return during an interval in which no trading occur. However, the knowledge that we might have about the market micro-structure depends strongly on access to those high-frequency data and the suitable technological support to store and process that vast amount of data.

In order to explain why chaos is generic in theoretical economics models but elusive in data we will discuss a possible answer to it, at least when studying financial time series. Our main *motivation* is that chaos is elusive in financial empirical studies because of loss of information that occurs when daily quotes are used. This could hinder the detection of chaos in those time series. Chaotic systems are sensitive to initial conditions, so temporal dependence is lost as the chaotic time series are sampled at too long-time intervals, appearing as independent even though they come from a (chaotic) dynamic system. In the case of financial time series, which quotes are almost continuously changing upwards or backwards on markets, the daily sampling may be too long. In order to avoid this problem high-frequency data can be used to detect chaos in financial time series.

Recently some papers have been published looking for nonlinearities and chaos using intra-daily financial time series. For instance, Lo and Lee (2006), Aslan and Sensoy (2019) and Leone and Kwabi (2019) focused on testing the hypothesis of efficient markets using high-frequency data; Bensaïda (2014) analysed the stock performance of the 500 large companies in the United States, S&P 500 index returns, over four different frequencies: weekly, daily, 30-min and 5-min; Anagnostidis and Emmanouilides (2015) studied the Athens Exchange Composite Share Price Index (ACSPI) returns over five different frequencies: 10-min, 20-min, 40-min, 50-min and 80-min; Vamvakaris et al. (2018) evaluated the S&P 500 index returns as well considering exclusively 5-min data.

All of them found little evidence to support the presence of chaotic behaviours in their datasets but they found strong evidences of nonlinearities. Note that most of papers published in financial literature so far deal with data equally spaced in time, both in low-frequency as in high-frequency. However financial markets are characterised by its almost continuity on trade, showing a variable time-lapse between each quote. Thus these *tick-by-tick* time series are not separated by a fixed time period. In this sense we have focused on time-series data with non-uniform time-frequency instead of considering time-series data with uniform time-frequency e.g., 1-month, 1-day, 1-hour, 30-min, 5-min or 1-min as papers mentioned above.

Unfortunately one rarely has the opportunity to access primary databases as most financial markets provide only access to participants, and it is not clear to what extent those data are retained and therefore potentially usable for time series studies. This issue has taken on additional importance with the growth of algorithmic trading. Hence our ability to analyze the dynamic behind financial markets at micro-level is limited mainly by the availability of relevant data. The chance to obtain easily and free high-frequency data on the Foreign Exchange Market over a sufficiently long time period has made this particular financial market the preferred habitat for studies of nonlinearities and chaos in economics and finance.

The information generated by the interactions between traders who buys (bid orders) and sells (ask orders) those currencies is apparently encoded in frequencies which are not usually equally spaced in time. In our case the frequency of those quotations is tick-by-tick. Each *tick* will appear when there is a change, upward or downward, in the quotations. That is, in the highest bid or lowest ask orders in the top-of-book. In this chapter we have considered 14 tick-by-tick time-series data about some top currency pairs from the Foreign Exchange Market during November 2018. This month has been selected arbitrarily without any economic criteria. The dataset has been obtained from `https://truefx.com`. This financial market is characterised by being open 24 hours, so it is a market with continuous operations throughout the day.

In this sense we have focused on the bid rates top-of-book, tick-by-tick historical market data, with fractional pip spreads in miliseconds detail. In particular we have collected the following currency pairs: the Australian Dollar against Japanese Yen (2.781.632 ticks), the Australian Dollar against US Dollar (2.138.904 ticks), the Canadian Dollar against Japanese Yen (2.115.263 ticks), the Swiss Franc against Japanese Yen (2.579.371 ticks), the Euro against Swiss Franc (2.110.966 ticks), the Euro against British Pound (2.663.370 ticks), the Euro against Japanese Yen (5.909.146 ticks), the Euro against US Dollar (2.963.370 ticks), the British Pound against Japanese Yen (5.023.443 ticks), the British Pound against US Dollar (2.967.520 ticks), the New Zealand Dollar against US Dollar (1.398.569 ticks), the US Dollar against Canadian Dollar (1.682.270 ticks), the US Dollar against Swiss Franc (1.497.969 ticks) and the US Dollar against Japanese Yen (2.066.618 ticks).

As we can see the volume of tick-by-tick data is huge when one considers the tick frequency, even taking into account only a month. Thus due to difficult for processing that massive amount of information we have decided to focus on the analysis of FX quotations occurred just during one day. Particularly we have considered two days for the analysis of each FX rate, the highest and the lowest day by volume of tick-by-tick data during November 2018, see table 6.1.1.

	AUDJPY	AUDUSD	CADJPY	CHFJPY	EURCHF	EURGBP	EURJPY	EURUSD	GBPJPY	GBPUSD	NZDUSD	USDCAD	USDCHF	USDJPY
Highest	Nov 15	Nov 15	Nov 15	Nov 15	Nov 15	Nov 15	Nov 15	Nov 15	Nov 15	Nov 15	Nov 01	Nov 23	Nov 01	Nov 15
Sample	39.946	69.714	46.900	98.018	69.329	132.588	96.363	88.095	62.188	106.578	25.189	43.694	60.998	50.984
	AUDJPY	AUDUSD	CADJPY	CHFJPY	EURCHF	EURGBP	EURJPY	EURUSD	GBPJPY	GBPUSD	NZDUSD	USDCAD	USDCHF	USDJPY
Lowest	Nov 30	Nov 19	Nov 05	Nov 05	Nov 05	Nov 30	Nov 05	Nov 19	Nov 30	Nov 05	Nov 05	Nov 05	Nov 06	Nov 30
Sample	12.309	40.047	20.635	42.040	41.131	51.610	43.918	53.123	14.923	50.927	14.210	19.774	31.124	21.061

Table 6.1.1: Samples used in the analysis pertaining to the highest and lowest day by volume of tick-by-tick data are shown.

As we have pointed out before chaotic systems are sensitive to initial conditions, so temporal dependence in chaotic time series is lost as the series are sampled at too long time intervals, appearing as independent and not chaotic, even though they will in fact come from a chaotic dynamic system. Normally when studying the chaotic behaviour of financial time series, the log returns data is used. In the case of daily time series, the daily returns is usually considered in the literature but if we take into account high-frequency time series we can consider series of higher frequency log returns e.g., 1-hour, 30-min, 10-min, 1-min, 5-seconds or even tick-by-tick. Hence to carry out our analysis we need to differentiate the log quotes series in order to get the *lagged returns* time-series data defined as follows.

Definition 6.1 (Lagged return)**.** A scalar discrete-time stationary time series $\{x_t\}_{t=1}^{n}$ defined as eq.3.2.1 is said to be a lagged return time series, denoted by $\{r_t\}_{t=1}^{n}$, if satisfied the following (*lagged return equation*)

$$r_t = \log(p_t) - \log(p_{t-lag}) \tag{6.1.1}$$

where $p_t \in \mathbb{R}$ is the tick-by-tick bid price of the financial assets, $t \in \mathbb{Z}^+$ (discrete-time), $lag \in \mathbb{N}$ denotes the order of differentiation.

Remark 6.1. If we would be considering the classical daily returns we should use a lag equal to 1 day ($lag = 86400$ seconds) but this does not have to be the best option. The optimum lag or order of differentiation should not be too large or too short if we want to preserve the time dependency. On the one hand, a too short lag provides essentially the same information between two consecutive observations and therefore does not allow dynamic dependency to be detected. On the other hand, as a consequence of the initial-value sensitivity property, a too long lag can lead to the loss of information on time dependency. We have followed the heuristic criterion proposed by Fraser and Swinney (1986) in order to fix the optimum lag to capture the possible time dependency, chaotic or not, in our tick-by-tick time series. Thus the optimum lag to get the lagged returns time series (eq.6.1.1) will be equal to the *first minimum* of the mutual information function.

lag	AUDJPY	AUDUSD	CADJPY	CHFJPY	EURCHF	EURGBP	EURJPY	EURUSD	GBPJPY	GBPUSD	NZDUSD	USDCAD	USDCHF	USDJPY
3600	1.42459	1.34625	1.50337	1.50159	1.71316	1.52873	1.21170	1.42573	1.54625	1.45168	1.73233	1.59206	1.53898	1.51053
7200	1.23655	1.11691	1.29593	1.30134	1.50935	1.35328	0.96424	1.19954	1.37024	1.25494	1.55559	1.38015	1.34643	1.29251
10800	1.12022	1.00191	1.16691	1.18435	1.40717	1.26871	0.84823	1.08479	1.30135	1.18291	1.43811	1.29774	1.26241	1.17464
14400	1.05144	0.93061	1.09273	1.11882	1.34925	1.21614	0.78184	1.01184	1.22157	1.12881	1.36683	1.22439	1.17135	1.11558
18000	0.98691	0.89853	1.05321	1.04852	1.32610	1.13441	0.72233	0.94331	1.17541	1.07617	1.32829	1.19751	1.09473	1.06644
21600	0.92871	0.83331	1.01367	0.99607	1.29193	1.10482	0.67257	0.89064	1.12633	1.02134	1.26301	1.14892	1.06831	1.01625
25200	0.90082	0.78429	0.99744	0.96286	1.25098	1.10193	0.66096	0.85904	1.10616	0.97135	1.20728	1.11489	1.02161	1.00179
28800	0.85602	0.75359	0.96548	0.93156	1.24326	1.07213	0.62382	0.82514	1.05579	0.92491	1.16313	1.07581	0.99364	1.00019
32400	0.83258	0.69828	0.93776	0.89471	1.22558	**1.04701**	0.59347	0.80185	1.03172	0.91025	1.11786	1.07084	**0.94479**	0.9484617
36000	0.80833	0.65877	0.93379	0.87185	1.18801	1.06716	0.59054	0.78896	0.98381	0.89134	1.08925	1.05505	0.95069	0.91559
39600	0.77306	**0.64091**	0.93086	**0.86755**	**1.16595**	1.04931	0.57587	**0.75947**	0.97374	**0.86827**	1.03633	1.03991	0.92188	0.89781
43200	0.73044	0.64211	0.89921	0.86902	1.16637	1.03255	0.56802	0.76269	0.96435	0.87463	0.99281	1.02901	0.91607	0.88747
46800	0.71662	0.60232	0.87415	0.83121	1.14554	1.02448	0.55146	0.74642	0.94519	0.84249	0.96866	1.02671	0.94222	0.85004
50400	0.67247	0.57976	0.84612	0.79956	1.13452	1.01389	0.53349	0.72851	0.94274	0.80738	**0.95131**	0.98866	0.92501	**0.83288**
54000	0.64832	0.55559	**0.83548**	0.77186	1.09961	0.97952	**0.53083**	0.70752	**0.93339**	0.80629	0.95533	**0.95153**	0.90336	0.84541
57600	**0.62981**	0.52376	0.85609	0.75879	1.07345	0.97612	0.53716	0.71585	0.95414	0.80065	0.94142	0.95926	0.91749	0.85708
61200	0.63268	0.52171	0.83295	0.73011	1.05239	0.96801	0.53737	0.70701	0.92948	0.77113	0.90274	0.93278	0.90368	0.83642
64800	0.60392	0.50274	0.82439	0.72112	1.04108	0.96718	0.53542	0.71083	0.93002	0.76172	0.88257	0.92187	0.87971	0.81798
68400	0.59598	0.48944	0.83611	0.71528	1.02521	0.94166	0.53504	0.67611	0.95333	0.75567	0.86402	0.92123	0.87076	0.80361
72000	0.59702	0.48123	0.83193	0.72357	1.00721	0.92892	0.54619	0.67727	0.95373	0.73781	0.84931	0.88651	0.82664	0.77131
75600	0.61451	0.50671	0.81867	0.72218	1.00651	0.93324	0.53669	0.67572	0.93955	0.74966	0.83763	0.88818	0.82161	0.78261
79200	0.59798	0.49557	0.78218	0.70111	1.01662	0.90463	0.53189	0.67004	0.91406	0.75626	0.84391	0.87592	0.80419	0.75957
82800	0.58447	0.46626	0.77105	0.69074	1.01375	0.89941	0.53071	0.68439	0.89332	0.74407	0.82951	0.85972	0.82128	0.72281
86400	0.56083	0.45092	0.72931	0.66752	0.96267	0.91170	0.50383	0.67808	0.87884	0.73124	0.81355	0.84002	0.81609	0.72271

Table 6.1.2: Mutual information values between $\ln(p_t)$ and $\ln(p_{t-lag})$ of each tick-by-tick time-series data for some top currency pairs.

The results shown in table 6.1.2 (above) and figure 6.1.1 (below) provide the following comments. First, we have checked if the first minimum of the mutual information function between $\ln(p_t)$ and $\ln(p_{t-lag})$ is lower than 24 hours (86400 seconds), where p_t is the tick-by-tick bid rates. Second, the first minimum of the mutual information function is lower than 86400 seconds (24 hours) for all currency pairs studied, between $32400 \leqslant lag \leqslant 57600$ seconds (9-16 hours), when one takes into account the full sample (tick-by-tick data). This fact would confirm our hypothesis. That is, we would be losing information if we employ daily frequency returns. Third, the mutual information values between $\ln(p_t)$ and $\ln(p_{t-lag})$ of each tick-by-tick time-series data considered is decreasing as the lag increases.

Hence this preliminary analysis establishes that we have to analyse time series data of financial returns lagged between 9 and 16 hours to optimise the information on time dependency. We just showed a possible explanation to the chaos model-data paradox in the field of economics. We have proposed the use of high-frequency data instead of daily data. Indeed, we have used bid rates, tick-by-tick time-series data considering all the information available in the Foreign Exchange Market. So, could it be this fact one of the reasons why it is so difficult to find chaos inside real-world financial time-series data?. To answer this question we should detect chaos in the tick-by-tick lagged returns time series in order to justify empirically our hypothesis. In this sense the first step will be get the delayed-coordinate embedding vectors because any method for estimating the Lyapunov exponent from some observed time-series data are based previously on the state space reconstruction procedure.

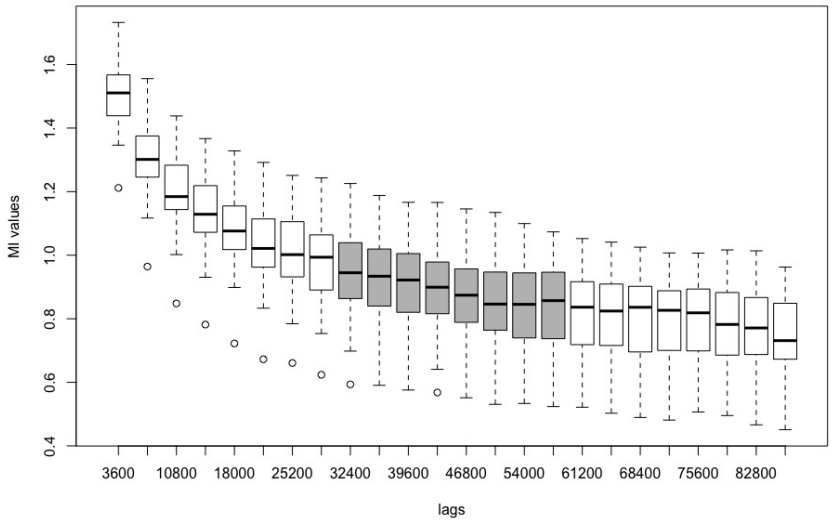


Figure 6.1.1: Boxplot of the Mutual Information values picked in table 6.1.1.

6.2 Non-uniform delayed-coordinate embedding procedure

The embedding (or reconstruction) procedure allows us to get all the relevant information about the unknown underlying dynamic system that generates the time-series data e.g., the Lyapunov exponents defined previously (see Section 2.3) must have approximately the same value in both the true and the reconstructed state space. This fact allows us to test the hypothesis of chaos (eq.3.3.1) in the unknown original dynamic system. The *embedding theorem* proposed by Takens (1981) provides a framework to reconstruct any unknown dynamic system which gave rise to a given observed scalar time series simply by reconstructing a new state space out of successive values of the time series, for a review see Section 3.2.

The reconstruction theorem 3.1 assumes that the dynamic system is sampled with *uniform* time-frequency e.g., 1-month, 1-day, 1-hour, 30-min, 5-min or 1-min. There has, however, been increasing interest in situations where observations are not uniform in time. That is, the data does not come from a series of values measured periodically in time. For instance, a motivating example would be the case of financial markets. The information generated by the interactions between traders who buys (bid orders) and sells (ask orders) financial instruments such as stocks, bonds, commodities, currencies, derivatives and so on is apparently encoded in frequencies which are not usually equally spaced in time e.g., tick-by-tick frequency.

Thus if we observe a particular financial asset i.e., a currency pair on the Foreign Exchange Market, the quotes or rates of this financial asset are sampling by tick-by-tick intervals which do not follow a constant rhythm. Each tick will appear when there is a change, upward or downward, in the trade price of each transaction. Can we use this kind of information to build a dynamic system model, and if so, how is it related to the true data-generating process?. Huke and Broomhead (2007) showed how to extend the reconstruction theorem under certain conditions when the dynamic system is sampled non-uniformly in time. So we can use their results to get the non-uniform delayed-coordinate embedding vectors from time-series data with non-uniform time-frequency. We have considered as in the uniform case the method of delayed-coordinates proposed by Ruelle and Takens (1971) to get the *non-uniform delayed-coordinate embedding* vectors from the tick-by-tick lagged returns time series as follows.

Definition 6.2 (Non-uniform delayed-coordinate embedding vector)**.** Let $\{r_{t_i}\}_{i=1}^{n}$ be the lagged returns time series given by definition 6.1 taking $t_i - t_{i-1} \neq t_s - t_{s-1}\ \forall i \neq s$. We form a sequence of *non-uniform delayed-coordinate embedding vectors* $\boldsymbol{r}_{t_i}^m$ by associating for each time period t_i a vector in a reconstructed state space $\mathbb{R}^m$ whose coordinates satisfy

$$\boldsymbol{r}_{t_i}^m = \left(r_{t_i}, r_{t_i-\tau}, r_{t_i-2\tau}, \ldots, r_{t_i-(m-2)\tau}, r_{t_i-(m-1)\tau}\right) \tag{6.2.1}$$

where m is the *embedding dimension*, τ is the *reconstruction time-delay* for $t_i - t_{i-1} \neq t_s - t_{s-1}\ \forall i \neq s$. Despite the fact that we have measured just a single observable, we can construct as in the uniform case a vector space whose axes represent all the relevant variables given by

$$\begin{aligned}
\boldsymbol{r}_{t_1}^m &= \left(r_{t_1}, r_{t_1-\tau}, r_{t_1-2\tau} \ldots, r_{t_1-(m-2)\tau}, r_{t_1-(m-1)\tau}\right) \\
\boldsymbol{r}_{t_2}^m &= \left(r_{t_2}, r_{t_2-\tau}, r_{t_2-2\tau} \ldots, r_{t_2-(m-2)\tau}, r_{t_2-(m-1)\tau}\right) \\
&\vdots \\
\boldsymbol{r}_{t_n}^m &= \left(r_{t_n}, r_{t_n-\tau}, r_{t_n-2\tau}, \ldots, r_{t_n-(m-2)\tau}, r_{t_n-(m-1)\tau}\right)
\end{aligned}$$

where the number of embedding vectors will be equal to $n - (m-1)\,\tau$ and its length equal to the embedding dimension m. The underlying idea is to make copies of the single observable signal with *non-uniform* time-frequency and consider those delayed values as coordinates of a reconstructed state space retrieved from the time-series data. Let us illustrate the state space reconstruction procedure in both cases for any embedding dimension m where the reconstruction time-delay $\tau = 3$, see figure 6.2.1

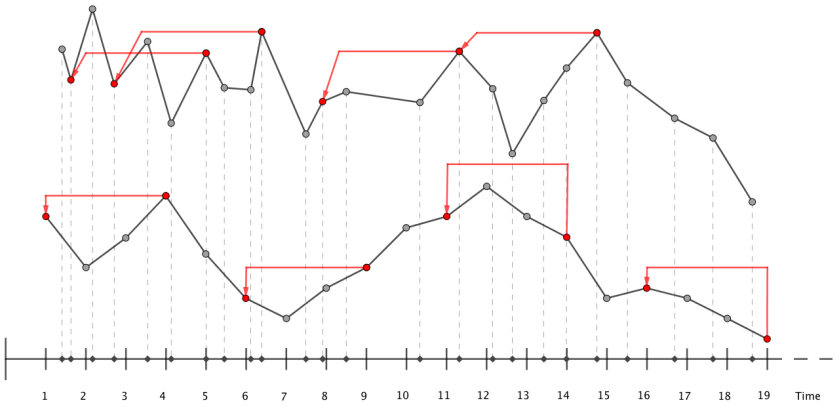


Figure 6.2.1: Uniform (below) and non-uniform (above) time-frequency for delayed-coordinate embedding vectors for any embedding dimension m where the time-delay $\tau = 3$.

Note that according to the embedding theorem just a single univariate time series should be needed to reconstruct properly the state space. However, for certain real-world time-series data there may be some advantages in using all attributes or variables available. Cao et al. (1998) can be considered pioneers at the early stage of developments about how we could embed a multivariate time series showing some numerical examples. A few years later Garcia and Almeida (2005), Vlachos and Kugiumtzis (2009) showed that multivariate attractor reconstruction preserves better the underlying dynamic of the unknown system. Deyle and Sugihara (2011) proved theoretically that the reconstruction theorem can be generalized suitable from single time series to multiple time series under certain conditions.

As far as this chapter is concerned we will consider only the non-uniform delayed-coordinate embedding vectors from a single univariate time-series data postponing for future work the multivariate case. Now we are going to discuss briefly the *algorithms* proposed in this context. The three R packages mentioned earlier have implemented their algorithms considering only the state space reconstruction from time-series data sampled with uniform time-frequency. The **fNonlinear** package has a function called `embeddPSR`, `buildTakens` is the function that belongs to the **nonlinearTseries** package and the **tseriesChaos** package includes the function `embedd`. Note that such packages consider the embedding vectors *forwards* while the **DChaos** package developed by us consider them *backwards* using the `embedding` function. Both ways are correct but we think it more convenient to define the delayed-coordinate embedding vectors backwards since in time series analysis we explain the initial and future instants from what has happened in the past. Let us illustrate the reconstruction procedure by the following examples.

Example 6.1 (Uniform delayed-coordinate embedding vectors forwards)**.** The first five values are showed using the `embedd` function which belongs to **tseriesChaos** package from the Logistic map contained in the **DChaos** package. In this example we choose arbitrarily the following embedding parameters $m = 5$, $\tau = 2$ and $n = 1000$. The number of embedding vectors will be $n - (m-1)\tau = 1000 - (5-1) \cdot 2 = 992$. Thus we have 992 points in the 5-dimensional reconstructed state space given by

```
## The time-series data from the Logistic map with chaos
[1] 7.47e-01 7.57e-01 7.37e-01 7.76e-01 6.95e-01 8.48e-01

## The uniform delayed-coordinate embedding vectors forwards
data  <- tseriesChaos::embedd(ts, m=5, d=2)
show(head(data, 5))

        V1/0      V1/2      V1/4       V1/6       V1/8
[1,] 0.7466701 0.7365940 0.69509054 0.51625546 0.00422337
[2,] 0.7566155 0.7760930 0.84775873 0.99894304 0.01682213
[3,] 0.7365940 0.6950905 0.51625546 0.00422337 0.06615659
[4,] 0.7760930 0.8477587 0.99894304 0.01682213 0.24711960
[5,] 0.6950905 0.5162555 0.00422337 0.06615659 0.74420600
```

Example 6.2 (Uniform delayed-coordinate embedding vectors backwards)**.** The first five values are showed using the `embedding` function which belongs to **DChaos** package considering the time serie previously simulated from the Logistic map. In this example we have chosen arbitrarily the same embedding parameters $m = 5$, $\tau = 2$ and $n = 1000$. The number of embedding vectors will be $n - (m-1)\tau = 1000 - (5-1) \cdot 2 = 992$. Thus we have 992 points in the 5-dimensional reconstructed state space again given by

```
## The uniform delayed-coordinate embedding vectors backwards
data  <- DChaos::embedding(ts, m=5, lag=2, timelapse="FIXED")
show(head(data, 5))

         y         x1         x2        x3        x4
1 0.00422337 0.51625546 0.69509054 0.7365940 0.7466701
2 0.01682213 0.99894304 0.84775873 0.7760930 0.7566155
3 0.06615659 0.00422337 0.51625546 0.6950905 0.7365940
4 0.24711960 0.01682213 0.99894304 0.8477587 0.7760930
5 0.74420600 0.06615659 0.00422337 0.5162555 0.6950905
```

Hence we can write the reconstructed trajectory joining the points from $x_1, \ldots, x_{992}$ where $\boldsymbol{x}_t^m = (x_t, x_{t-\tau}, x_{t-2\tau}, x_{t-3\tau}, x_{t-4\tau})$ for $t = 1, \ldots, 992$, for a review see definition 3.3. Now let us focus on the non-uniform case showing the *algorithm* proposed in this context.

Algorithm 6.1 (Non-uniform delayed-coordinate embedding vectors)**.** Let $\{x_{t_i}\}_{i=1}^{n}$ be a time-series data with non-uniform time-frequency given by eq.6.2.1 taking $t_i - t_{i-1} \neq t_s - t_{s-1}\ \forall i \neq s$. Let $m \geqslant 2$ and $\tau \geqslant 1$ be the embedding parameters. Let $f_{t_i} = t_i - t_1$ be the cumulative frequency. The *non-uniform embedding* procedure can be described as follows.

Data: A time-series data with non-uniform time-frequency
Result: The non-uniform delayed-coordinate embedding vectors
Begin ;
for $k = 1 : m$ **do**
  $ii \leftarrow 1$;
  for $i = 1 : n$ **do**
    $F_{t_i} \leftarrow f_{t_i} - (k * \tau)$;
    if $F_{t_i} < 0$ **then**
      $x_{t_i} \leftarrow 0$;
    else
      while $F_{t_i} > f_{t_{ii}}$ **do**
        $ii \leftarrow ii + 1$;
        if $F_{t_i} = 0$ **then**
          $x_{t_i} \leftarrow x_{t_{ii}}$;
        else
          $x_{t_i} \leftarrow x_{t_{ii-1}}$;
        end
      end
    end
  end
  $\boldsymbol{x}_{t_i}^{k} \leftarrow \{x_{t_i}\}_{i=1}^{n}$ by columns ;
end

In this sense there is no analogous functions in the three R packages already mentioned since they consider exclusively time-series data with uniform time-frequency. The **DChaos** package also takes into account time-series data with non-uniform time-frequency. As far as we know the **DChaos** package is the first R library which provides the state space reconstruction from time-series data with non-uniform time-frequency by the `embedding` function which arguments are (`data`, `m`, `lag`, `timelapse`). The parameter `timelapse` is a character denoting if the time-series data are sampled at uniform time-frequency e.g., 1-month, 1-day, 1-hour, 30-min, 5-min, 1-min and so on then `timelapse=FIXED` must be selected see e.g., Example 6.2. Otherwise `timelapse=VARIABLE` has to be specified e.g., tick-by-tick frequency. See R documentation for the interpretation of this parameter set. Let us illustrate the non-uniform embedding procedure by the following example.

Example 6.3 (Non-uniform delayed-coordinate embedding vectors)**.** The function `embedding` returns the uniform or non-uniform delayed-coordinate embedding vectors backwards by columns from an univariate time serie considering the parameter set selected by the user. If `FIXED` has been selected `data` must be a `vector` or a time-series object `ts` or `xts`. Otherwise `VARIABLE` has to be specified. In this case `data` must be a `data.frame`, a `data.table` or a `matrix` with two columns, the date and the univariate time series as a sequence of numerical values, in that order. The date can have the following three classes: `POSIXt`, `Date` or `Factor`. In the latter case the date should come in the following format `YMD H:M:OS3` considering milliseconds e.g., `20190407 00:00:03.347`. If the user does not consider them must put `.000` after seconds.

Let us focus on a brief example about the non-uniform embedding case. We have considered a tick-by-tick dataset called `sbux` which contains the bid quote data with non-uniform frequency for Starbucks company. It is available in the **highfrequency** package. This R package provides several useful tools for high-frequency time-series data analysis. In this case we have obtained the sequence of non-uniform delayed-coordinate embedding vectors $\boldsymbol{x}_{t_i}^m$ taking $t_i - t_{i-1} \neq t_s - t_{s-1}\ \forall i \neq s$ whose coordinates satisfy $\boldsymbol{x}_{t_i}^m = \left(x_{t_i}, x_{t_i-\tau}, x_{t_i-2\tau}, \ldots, x_{t_i-(m-2)\tau}, x_{t_i-(m-1)\tau}\right)$ following algorithm 6.1. The first five values corresponding to the non-uniform embedding for `m = 3`, `lag = 4` and `timelapse=VARIABLE` are showed below.

```
## The tick-by-tick data from the bid quote for Starbucks company
ts    <- highfrequency::sbux

## The non-uniform delayed-coordinate embedding vectors backwards
data  <- DChaos::embedding(ts, m=3, lag=4, timelapse="VARIABLE")
show(head(data, 5))
                           y           x1           x2
2010-07-01 15:30:09 -0.0012327924 -0.0010255359  0.0008179960
2010-07-01 15:30:11  0.0000000000 -0.0002053177 -0.0004090816
2010-07-01 15:30:12 -0.0004112688 -0.0002053177 -0.0020479221
2010-07-01 15:30:15  0.0004112688  0.0000000000 -0.0002053177
2010-07-01 15:30:16 -0.0004112688 -0.0004112688 -0.0002053177
```

Now once we have shown how to deal with the state space reconstruction let us move on to the next step. There are mainly two kind of methods in order to estimate the Lyapunov exponents from time-series data as we have seen previously. In the next section we are going to focus on the jacobian indirect method based on the global neural net approach exclusively postponing for future work the application on these tick-by-tick financial time series considering the local jacobian indirect methods seen in Chapter 5.

6.3 Lyapunov exponents from tick-by-tick time-serie data

Methods and techniques related to test the hypothesis of chaos (eq.3.3.1) try to quantify the initial-value sensitive property estimating the so-called Lyapunov exponents, for a review see Section 2.3. If one knows the data-generating process behind the time series the theoretical Lyapunov exponent can be calculated directly using its own definition, see Section 2.4. However we have now assumed that the true dynamics of the system is unknown because in the tick-by-tick financial time series the data-generating process is rarely known a priori. The embedding procedure allows us to get all the relevant information (invariant properties) about the unknown underlying dynamic system that generates the time-series data as we have seen in the earlier section. In this sense, the Lyapunov exponents must have approximately the same value in both the true and the reconstructed state space. Remember that this fact allows us to test the hypothesis of chaos in the unknown original dynamic system.

There are two main methods in the literature that provide the estimated Lyapunov exponent from time-series data. The first one, the so-called *direct* approach which directly measures the growth rate of the divergence between two trajectories with an infinitesimal difference in their initial conditions, for a review see Section 3.3. The second one, the so-called *indirect* approach (or jacobian-based method) which try to estimate the jacobian of the underlying generating system and then those partial derivatives are used to compute the Lyapunov exponent applying their analytical definition, for a review see Section 4.1. As far as this chapter is concerned we have focused into the jacobian indirect methods using the global neural net approach. Keeping in mind that tick-by-tick financial time-series data are usually noise-contaminated signals and characterised by an erratic and persistent volatility in certain periods, this procedure has the following advantages over all other methods: (*i*) its robustness to the presence of (small) noise; (*ii*) their satisfactory performance in detecting existing nonlinearities on time-series data of moderate sample sizes; (*iii*) allows the estimation of the full spectrum of Lyapunov exponents; (*iv*) the asymptotic distribution of the estimator can be derived allowing the building of formal tests.

Now let us focus on knowing how we can implement the global neural net method (algorithm 4.1) in the non-uniform case postponing for future work the application to local methods in this context. Once we have got the non-uniform delayed-coordinate embedding vectors from the tick-by-tick financial time-series data (eq.6.2.1) we have to move on to the next step. In this case, the procedure behind the jacobian indirect method can be described as follows.

Definition 6.3 (Jacobian indirect method (non-uniform case))**.** Let assume following Gençay and Dechert (1992) that there exist a function $H : \mathbb{R}^m \to \mathbb{R}^m$ such that $\boldsymbol{r}_{t_i}^m = H\left(\boldsymbol{r}_{t_i-\tau}^m\right)$ where $\boldsymbol{r}_{t_i}^m$ are the non-uniform delayed-coordinate embedding vectors from the tick-by-tick lagged returns time series which contains the uniform case taking $t_i - t_{i-1} = t_s - t_{s-1}\ \forall i \neq s$. Under the assumption that the embedding is a homeomorphism, the map H is *topologically conjugate* to the unknown dynamic system F. This implies that certain dynamic properties of F and H are the same. In our case, the Lyapunov exponents of F and H should be the same, so we can focus on estimating the exponents from the map H. The dynamic system H may be expressed as a matrix by

$$\begin{pmatrix} r_{t_i} \\ r_{t_i-\tau} \\ \vdots \\ r_{t_i-(m-2)\tau} \\ r_{t_i-(m-1)\tau} \end{pmatrix} = H \begin{pmatrix} r_{t_i-\tau} \\ r_{t_i-2\tau} \\ \vdots \\ r_{t_i-(m-1)\tau} \\ r_{t_i-m\tau} \end{pmatrix}$$

$$= \begin{pmatrix} \nu\left(r_{t_i-\tau}, r_{t_i-2\tau}, \ldots, r_{t_i-(m-2)\tau}, r_{t_i-(m-1)\tau}, r_{t_i-m\tau}\right) \\ r_{t_i-\tau} \\ \vdots \\ r_{t_i-(m-2)\tau} \\ r_{t_i-(m-1)\tau} \end{pmatrix} \tag{6.3.1}$$

The jacobian corresponding to the reconstructed dynamic system H will be obtained as follows

$$\partial H = \begin{pmatrix} \frac{\partial \nu}{\partial r_{t_i-\tau}} & \frac{\partial \nu}{\partial r_{t_i-2\tau}} & \frac{\partial \nu}{\partial r_{t_i-3\tau}} & \cdots & \frac{\partial \nu}{\partial r_{t_i-(m-1)\tau}} & \frac{\partial \nu}{\partial r_{t_i-m\tau}} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{6.3.2}$$

Hence the estimation of the Lyapunov exponent by the jacobian indirect method is reduced to the estimation of the unknown nonlinear function $\nu : \mathbb{R}^m \to \mathbb{R}$ as in the uniform case. In this sense we have focused on the global neural net approach as well. The neural network (or neural net) estimator of the Lyapunov exponent was first proposed by McCaffrey et al. (1992) and Nychka et al. (1992), and then revisited by Gençay and Dechert (1992) and by Shintani and Linton (2003, 2004).

Note that in our case the main reason for using neural network models is not to look for the best predictive model but to estimate a model that captures the nonlinear time dependence well enough and, additionally, allows us to obtain in an analytical way (instead of numerical) the jacobian functional of the unknown data-generating process (eq.6.3.1). The estimation of this jacobian will allow us to contrast the hypothesis of chaos (eq.3.3.1) using the equation 2.3.6. The results proposed by the authors mentioned above have enabled us to consider a neural network with just one single hidden layer. The number of hidden units (or neurones) in the single hidden layer is determined by statistical methods based on model selection criteria as it appears in the results of this chapter. Now let us illustrate how we have obtained a consistent neural net estimator based on the robust estimation of the function ν in the jacobian (eq.6.3.1). We have developed our own algorithm in the non-uniform case as follows.

Algorithm 6.2 (Neural net approach (non-uniform case))**.** Let consider the embedding vector $\boldsymbol{r}_{t_i} = \nu\left(r_{t_i-\tau}, r_{t_i-2\tau}, \ldots, r_{t_i-(m-2)\tau}, r_{t_i-(m-1)\tau}, r_{t_i-m\tau}\right)$ as defined by eq.6.3.1. The *neural network estimator* can be obtained by approximating the unknown nonlinear function ν through a feed-forward single hidden layer network with a single output by

$$\nu \approx \hat{\nu} = \Phi_0 \left[\hat{\alpha}_0 + \sum_{q=1}^{h} \hat{\omega}_{qo} \Phi_q \left(\hat{\alpha}_q + \sum_{j=1}^{m} \hat{\omega}_{jq} r_{t_i-j\tau} \right) \right] \tag{6.3.3}$$

where $\Phi_0 \in I$, $\hat{\alpha}_0$ is the estimated network bias from input, h is the number of neurones (or nodes) in the single hidden layer, $\hat{\omega}_{q0}$ are the estimated layers connection weights from input to hidden layer, Φ_q is the transfer function which in our case is the logistic function, $\hat{\alpha}_q$ is the estimated network bias from hidden layer, m is the embedding dimension and $\hat{\omega}_{jq}$ are the estimated layers connection weights from hidden layer to output. The issue of parameter estimation is reduced to a least squares problem in which the quantity to be minimized is defined by

$$\sum_{i=1}^{n} \left(r_{t_i} - \left[\alpha_0 + \sum_{q=1}^{h} \omega_{qo} \Phi_q \left(\alpha_q + \sum_{j=1}^{m} \omega_{jq} r_{t_i-j\tau} \right) \right] \right)^2$$

We have obtained the partial derivates of the jacobian in eq.6.3.2 applying the chain rule to eq.6.3.3 as

$$\frac{\partial \hat{\nu}}{\partial r_{t_i-j\tau}} = \Phi'_0\left(z_0\right) \sum_{q=1}^{h} \hat{\omega}_{q0} \Phi'_0\left(z_q\right) \hat{\omega}_{jq} \tag{6.3.4}$$

where

$$z_0 = \hat{\alpha}_0 + \sum_{q=1}^{h} \hat{\omega}_{q0} \Phi_q (z_q), \quad z_q = \hat{\alpha}_q + \sum_{j=1}^{m} \hat{\omega}_{jq} r_{t_i - j\tau}$$

and the estimated partial derivates are given by

$$\partial \hat{H} = \begin{pmatrix} \frac{\partial \hat{\nu}}{\partial r_{t_i-\tau}} & \frac{\partial \hat{\nu}}{\partial r_{t_i-2\tau}} & \frac{\partial \hat{\nu}}{\partial r_{t_i-3\tau}} & \cdots & \frac{\partial \hat{\nu}}{\partial r_{t_i-(m-1)\tau}} & \frac{\partial \hat{\nu}}{\partial r_{t_i-m\tau}} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{6.3.5}$$

Once we have obtained them we are going to estimate the kth Lyapunov exponent (eq.2.3.6) given by the *Lyapunov exponent estimator* as *(global neural net approach)*:

$$\hat{\lambda}_k = \lim_{M \to \infty} \frac{1}{M} \log \mu_k \left(\left| \partial \hat{H}^M \right| \right) \tag{6.3.6}$$

where μ_k is the kth largest eigenvalue provided by $\partial \hat{H}^M = \partial \hat{H}(x_M) \cdot \partial \hat{H}(x_{M-1}) \cdot \ldots \cdot \partial \hat{H}(x_1)$ for $k = 1, 2, 3, \ldots, m$ where m denote the embedding dimension and $\partial \hat{H}(\cdot)$ are the partial derivates estimated above following equation 6.3.5.

Remark 6.2. Note that it is necessary to distinguish between the sample size n used for estimating the partial derivatives of the jacobian in eq.6.3.5 and the *block length* M, defined in eq.6.3.6, which is the number of evaluation points (number of products of the jacobian) used for estimating the kth Lyapunov exponent, for a review see Section 4.2. The asymptotic properties of the Lyapunov exponent estimator defined by eq.6.3.6 was derived by Shintani and Linton (2004). They provided a statistical framework for testing the hypothesis of chaos (eq.3.3.1) based on the neural net estimator of the Lyapunov exponent and the consistent estimator of its variance, for a review see Section 4.3. The asymptotic properties of the Lyapunov exponent estimator are exactly the same in the uniform case as in the non-uniform case. Hence we have used these results to calculate the standard error of the Lyapunov exponent estimator and investigate the statistical significance of the sign of the exponents from the tick-by-tick lagged returns time-series data. Now, let us focus on the statistical analysis of the tick-by-tick data considered in this chapter.

6.4 Data analysis and simulations

In this section we provide the main empirical results of this chapter. We are going to compare the estimation of the largest Lyapunov exponent provided by several blocking methods. We have considered the full sample denoted by $\hat{\lambda}_F$ and three different blocking methods: non-overlapping subsampling $\hat{\lambda}_N$, equally spaced subsampling $\hat{\lambda}_E$ and bootstrap subsampling $\hat{\lambda}_B$. The results report by this section have been obtained using different algorithms from the **DChaos** package. These algorithms are publicly available at `www.CRAN.R-project.org/package=DChaos`. To save CPU time we have set the embedding dimension $3 \leqslant m \leqslant 10$, the time-delay $1 \leqslant \tau \leqslant 10$, the number of nodes in the single hidden layer $2 \leqslant h \leqslant 10$ and the length of all time-series data is showed in table 6.4.1. In particular 720 different neural nets models have been estimated by each of the 28 tick-by-tick time-series data in order to obtain the results shown in figure 6.4.1 and table 6.4.1.

We have illustrated in figure 6.4.1 the box-plot diagrams of the largest Lyapunov exponents derived from all the neural network regressions for each Fx rate based on the bootstrap blocking method. Results pertaining to the highest day by volume of tick-by-tick data are shown above. Those of the lowest day are shown below. As one can see the estimated median value (bold line) of the Lyapunov exponent based on 720 different neural nets models estimated are positives in twelve FX rates in both days, see figure 6.4.1, only the AUDJPY and USDCAD rates among the 14 currency pairs considered have a negative value.

The results provided by table 6.4.1 show the estimation of the largest Lyapunov exponent corresponding to the best-fitted neural net model (with a lower BIC value) for each currency pair and blocking method. Let us point out some useful information to understand the data presented in this table. Numbers in parentheses are standard errors. Into brackets denote $[m, \tau, h]$. We have indicated in bold those Lyapunov exponent with positive values. We have also remarked with an asterisk $*$ those Lyapunov exponent that are statistically significant at the 99% confidence level. For the estimation based on blocking methods median values of all used blocks are presented. For the block length M we use $M = int\left[c \times (n/\log n)^{1/6}\right]$ with $c = 36.2$ where $int\,[A]$ signifies the integer part of A. The number of blocks B depends on the sample size n of each currency pair, see table 4.2.1. QS Kernel with optimal bandwidth has been used for the heteroskedasticity and autocorrelation consistent covariance matrix estimation following Andrews (1991).

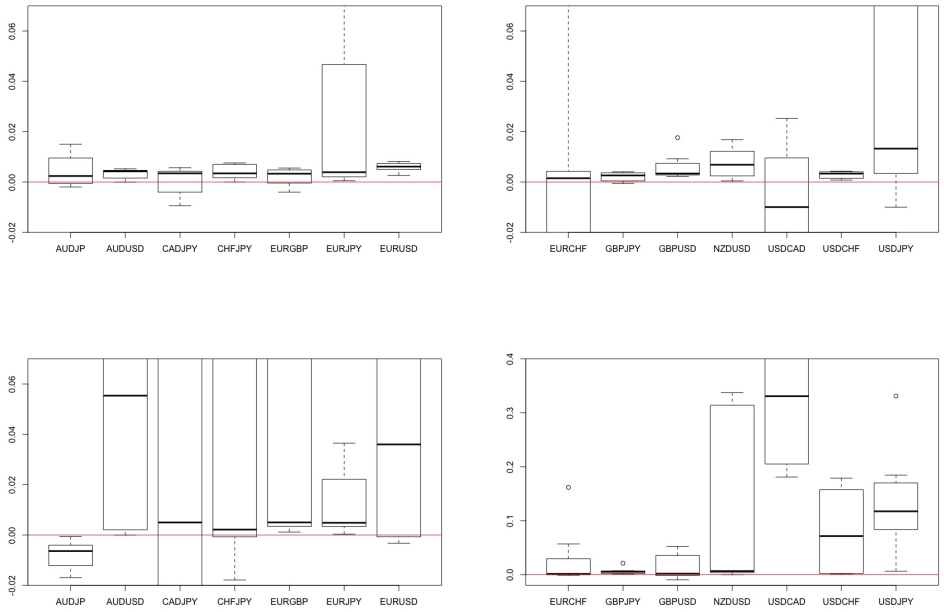


Figure 6.4.1: Box-plot from the Lyapunov exponent based on the bootstrap blocking method. We have considered 720 different neural nets models estimated by each currency pair for $3 \leqslant m \leqslant 10$, $1 \leqslant \tau \leqslant 10$ and $2 \leqslant h \leqslant 10$. Results pertaining to the highest day by volume of tick-by-tick data are shown above. Those of the lowest day are shown below.

The data shown in table 6.4.1 provide the following comments. First, if we take into account a sufficiently high embedding dimension m (above 7) the estimated values of the largest expected Lyapunov exponent are positives and significants in both sampled days for the following currency pairs: EURCHF, EURGBP, EURJPY, NZDUSD, GBPUSD, USDJPY. The currency pairs AUDUSD, CADJPY, EURUSD, GBPJPY, USDCHF were positives but only significants on the lowest day by volume of tick-by-tick data, while the CHFJPY rate was positive but not significant in either day. The AUDJPY and USDCAD rates are the only ones that show negative values on one of the two days. Thus these results would support the hypothesis of chaotic behaviour in almost all the FX rates considered. Second, we have found in all cases that the optimum time-delay $\tau = 1$ as Takens (1981) suggested. Third, the results provided by the three blocking methods improve the case of the full sample being very similar between them. Although, on average, the bootstrap method gives better results. These facts are consistent with the suggestion proposed by Shintani and Linton (2004) on the use of blocking methods instead of full sampling when estimating the Lyapunov exponents.

	$[m,\tau,h]$	$\hat{\lambda}_F$	$\hat{\lambda}_N$	$\hat{\lambda}_E$	$\hat{\lambda}_B$	$[m,\tau,h]$	$\hat{\lambda}_F$	$\hat{\lambda}_N$	$\hat{\lambda}_E$	$\hat{\lambda}_B$
AUDJPY	[7,1,10]	**0.19177** (8.1e-04)*	**0.18127** (7.2e-03)*	**0.18194** (8.4e-03)*	**0.18199** (9.0e-03)*	[7,1,5]	-0.02013 (8.7e-05)	-0.01061 (6.9e-03)	-0.01057 (6.7e-03)	-0.01043 (6.6e-03)
AUDUSD	[8,1,9]	-0.00208 (1.4e-05)	**0.00447** (4.3e-03)	**0.00445** (4.3e-03)	**0.00440** (4.2e-03)	[9,1,8]	**0.16010** (1.0e-03)*	**0.15787** (3.2e-03)*	**0.15488** (5.0e-03)*	**0.15413** (4.6e-03)*
CADJPY	[9,1,9]	-0.00094 (2.0e-05)	**0.00572** (4.2e-03)	**0.00564** (4.1e-03)	**0.00557** (4.0e-03)	[10,1,2]	**0.17453** (9.0e-04)*	**0.17184** (2.7e-03)*	**0.17182** (6.4e-03)*	**0.17253** (6.1e-03)*
CHFJPY	[8,1,10]	-0.00054 (8.3e-04)	**0.00615** (5.4e-03)	**0.00673** (5.3e-03)	**0.00672** (5.2e-03)	[7,1,10]	-6.9e-06 (9.4e-05)	**0.00529** (4.5e-03)	**0.00419** (4.3e-03)	**0.00423** (4.6e-03)
EURCHF	[7,1,10]	**0.37030** (2.2e-04)*	**0.36021** (7.9e-03)*	**0.35997** (7.7e-03)*	**0.36013** (7.5e-03)*	[8,1,10]	**0.16506** (3.0e-03)*	**0.16120** (3.2e-03)*	**0.16179** (3.6e-03)*	**0.16177** (3.9e-03)*
EURGBP	[7,1,8]	**0.00172** (7.2e-05)*	**0.00564** (2.9e-04)*	**0.00559** (2.9e-04)*	**0.00554** (2.9e-04)*	[10,1,7]	**0.10923** (9.5e-05)*	**0.10013** (7.1e-03)*	**0.09987** (6.2e-03)*	**0.09998** (6.3e-03)*
EURJPY	[7,1,10]	**0.31488** (1.5e-04)*	**0.30518** (5.0e-03)*	**0.30540** (4.9e-03)*	**0.30588** (4.9e-03)*	[9,1,5]	**0.04076** (2.1e-05)*	**0.03649** (4.0e-03)*	**0.03624** (4.2e-03)*	**0.03687** (4.1e-03)*
EURUSD	[9,1,8]	-0.00080 (1.4e-05)	**0.00759** (5.0e-03)	**0.00774** (4.8e-03)	**0.00797** (4.8e-03)	[9,1,9]	**0.14388** (7.5e-04)*	**0.14191** (1.2e-02)*	**0.14017** (1.2e-02)*	**0.14013** (1.2e-02)*
NZDUSD	[7,1,9]	**0.08501** (6.4e-04)*	**0.08241** (4.3e-03)*	**0.08308** (7.4e-03)*	**0.08281** (7.9e-03)*	[8,1,5]	**0.29497** (8.2e-04)*	**0.29438** (3.4e-03)*	**0.29734** (7.6e-03)*	**0.29703** (7.7e-03)*
GBPJPY	[9,1,9]	-0.00236 (7.5e-05)	**0.00372** (3.7e-03)	**0.00389** (3.6e-03)	**0.00411** (3.6e-03)	[10,1,6]	**0.02425** (4.8e-05)*	**0.02116** (1.5e-03)*	**0.02098** (1.0e-03)*	**0.02114** (1.1e-03)*
GBPUSD	[10,1,7]	**0.01143** (7.5e-04)*	**0.01664** (3.7e-03)*	**0.01785** (3.6e-03)*	**0.01797** (3.6e-03)*	[9,1,9]	**0.02665** (4.5e-03)*	**0.03286** (5.5e-03)*	**0.02690** (1.2e-03)*	**0.02686** (1.2e-03)*
USDCAD	[9,1,8]	-0.01118 (8.1e-05)	-0.00549 (4.5e-03)	-0.00530 (4.4e-03)	-0.00504 (4.4e-03)	[7,1,10]	**0.61118** (2.6e-02)*	**0.61610** (8.6e-03)*	**0.61349** (6.5e-03)*	**0.61437** (6.7e-03)*
USDCHF	[8,1,5]	-0.00058 (2.5e-05)	**0.00427** (4.8e-03)	**0.00454** (4.7e-03)	**0.00473** (4.7e-03)	[10,1,2]	**0.14118** (1.1e-03)*	**0.13634** (2.1e-03)*	**0.13615** (5.0e-03)*	**0.13596** (5.1e-03)*
USDJPY	[7,1,10]	**0.36246** (5.1e-03)*	**0.36575** (4.8e-03)*	**0.36713** (4.5e-03)*	**0.36957** (4.4e-03)*	[8,1,10]	**0.3365** (2.6e-04)*	**0.33253** (3.9e-03)*	**0.33091** (4.5e-03)*	**0.33095** (4.4e-03)*

Table 6.4.1: The Lyapunov exponent values based on several blocking methods by each currency pair (F=Full, N=Non-overlapping, E=Equally spaced, B=Bootstrap). Results from the highest day by volume of tick-by-tick data are shown on the left. Those of the lowest day are shown on the right.

To sum up, we have been able to detect chaotic signals inside real-world financial time-series data, at least in the 14 currency pairs and days chosen. Hence the results obtained in this chapter would support our proposal that chaos is elusive in most financial empirical studies because of loss of information that occurs when daily quotes are used. This could hinder the detection of chaos in those time series. We have proposed take into account all the information available in the financial markets instead of daily data. Chaotic systems are sensitive to initial conditions, so temporal dependence is lost as the chaotic time series are sampled at too long-time intervals, appearing as independent even though they come from a (chaotic) dynamic system. Thus the daily sampling may be too long as we have seen. In order to avoid this problem high-frequency data can be used. Indeed, we have focused on tick-by-tick financial time-series data which are characterised by its almost continuity on trade showing a variable time-frequency between each quote. In this sense we have obtained a possible explanation to the chaos model-data paradox, at least in the 14 tick-by-tick rates and days considered. The preliminary results obtained from this empirical analysis are provided below.

AUDJPY	[3,1,4]	[4,1,9]	[5,1,7]	[6,1,10]	[7,1,10]	[8,1,10]	[9,1,9]	[10,1,5]
$\hat{\lambda}_F$	-0.00479	-0.00335	-0.00481	-0.03757	**0.19177**	**0.00965**	-0.00565	-0.00563
	(5.1e-05)	(1.4e-04)	(2.4e-05)	(1.3e-02)	(8.1e-04)*	(1.0e-02)	(3.2e-05)	(3.4e-05)
$\hat{\lambda}_N$	-0.00198	**0.00115**	**0.00083**	-0.02811	**0.18127**	**0.01652**	**0.00369**	**0.00409**
	(2.7e-03)	(4.2e-03)	(4.6e-03)	(6.3e-03)	(7.2e-03)*	(5.7e-03)*	(5.8e-03)	(5.7e-03)
$\hat{\lambda}_E$	-0.00198	**0.00112**	**0.00084**	-0.02465	**0.18194**	**0.01509**	**0.00369**	**0.00410**
	(2.7e-03)	(4.2e-03)	(4.6e-03)	(1.7e-02)	(8.4e-03)*	(8.5e-03)*	(5.8e-03)	(5.7e-03)
$\hat{\lambda}_B$	-0.00196	**0.00111**	**0.00084**	-0.02455	**0.18199**	**0.01500**	**0.00370**	**0.00410**
	(2.7e-03)	(4.2e-03)	(4.6e-03)	(1.7e-02)	(9.0e-03)*	(8.7e-03)*	(5.8e-03)	(5.7e-03)
AUDUSD	[3,1,9]	[4,1,6]	[5,1,10]	[6,1,10]	[7,1,10]	[8,1,9]	[9,1,10]	[10,1,7]
$\hat{\lambda}_F$	-0.00225	-0.00338	-0.00131	-0.00112	-0.00162	-0.00208	-0.00207	-0.00194
	(1.6e-04)	(1.1e-05)	(1.9e-05)	(2.3e-05)	(1.4e-05)	(1.4e-05)	(1.4e-05)	(1.6e-05)
$\hat{\lambda}_N$	**0.00010**	**4.1e-08**	**0.00292**	**0.00310**	**0.00413**	**0.00447**	**0.00489**	**0.00529**
	(2.3e-03)	(3.3e-03)	(3.7e-03)	(3.8e-03)	(3.8e-03)	(4.3e-03)	(4.3e-03)	(4.4e-03)
$\hat{\lambda}_E$	**0.00008**	-9.8e-06	**0.00308**	**0.00448**	**0.00410**	**0.00445**	**0.00487**	**0.00528**
	(2.4e-03)	(3.3e-03)	(4.0e-03)	(4.1e-03)	(3.8e-03)	(4.3e-03)	(4.3e-03)	(4.4e-03)
$\hat{\lambda}_B$	**0.00007**	-0.00001	**0.00307**	**0.00442**	**0.00411**	**0.00440**	**0.00487**	**0.00528**
	(2.4e-03)	(3.3e-03)	(4.0e-03)	(4.1e-03)	(3.8e-03)	(4.2e-03)	(4.3e-03)	(4.4e-03)
CADJPY	[3,1,9]	[4,1,10]	[5,1,7]	[6,1,9]	[7,1,8]	[8,1,8]	[9,1,9]	[10,1,9]
$\hat{\lambda}_F$	-0.00107	**0.00010**	-0.00088	-0.00118	-0.00068	-0.01182	-0.00094	-0.40771
	(7.7e-06)	(2.5e-05)	(1.6e-05)	(2.1e-05)	(1.8e-05)	(3.6e-03)	(2.0e-05)	(6.5e-05)
$\hat{\lambda}_N$	**0.00136**	**0.00350**	**0.00343**	**0.00386**	**0.00476**	-0.00556	**0.00572**	-0.39337
	(2.4e-03)	(3.4e-03)	(4.0e-03)	(4.2e-03)	(4.2e-03)	(5.4e-03)	(4.2e-03)	(3.1e-02)
$\hat{\lambda}_E$	**0.00120**	**0.00363**	**0.00358**	**0.00398**	**0.00495**	-0.00918	**0.00564**	-0.39335
	(2.2e-03)	(3.6e-03)	(4.0e-03)	(4.2e-03)	(4.1e-03)	(5.2e-03)	(4.1e-03)	(3.1e-02)
$\hat{\lambda}_B$	**0.00107**	**0.00384**	**0.00377**	**0.00426**	**0.00524**	-0.00940	**0.00557**	-0.39341
	(2.1e-03)	(3.1e-03)	(4.0e-03)	(4.2e-03)	(4.0e-03)	(5.1e-03)	(4.0e-03)	(3.0e-02)
CHFJPY	[3,1,4]	[4,1,9]	[5,1,6]	[6,1,7]	[7,1,10]	[8,1,10]	[9,1,8]	[10,1,7]
$\hat{\lambda}_F$	-0.00070	-0.00036	-0.00056	-0.00053	-0.00061	-0.00054	**0.00251**	**0.00457**
	(5.3e-05)	(1.8e-04)	(6.1e-06)	(7.0e-06)	(7.9e-03)	(8.3e-04)	(4.8e-03)	(2.7e-04)*
$\hat{\lambda}_N$	**0.00093**	**0.00247**	**0.00315**	**0.00377**	**0.00593**	**0.00615**	**0.00773**	**0.00533**
	(1.7e-03)	(2.8e-03)	(3.3e-03)	(3.4e-03)	(4.7e-03)	(5.4e-03)	(8.1e-03)	(3.4e-03)
$\hat{\lambda}_E$	**0.00099**	**0.00251**	**0.00340**	**0.00390**	**0.00731**	**0.00673**	**0.00771**	**0.00554**
	(1.6e-03)	(2.8e-03)	(3.1e-03)	(3.2e-03)	(4.4e-03)	(5.3e-03)	(7.9e-03)	(3.2e-03)
$\hat{\lambda}_B$	**0.00131**	**0.00260**	**0.00365**	**0.00412**	**0.00729**	**0.00672**	**0.00759**	**0.00576**
	(1.6e-03)	(2.7e-03)	(3.1e-03)	(3.0e-03)	(4.3e-03)	(5.2e-03)	(7.9e-03)	(3.2e-03)

Table 6.4.2: The Lyapunov exponent values based on several blocking methods by each currency pair. Intermediate results pertaining to the highest day by volume of tick-by-tick data are shown. Numbers in parentheses are standard errors. Into brackets denote $[m, \tau, h]$. We have indicated in bold those Lyapunov exponent with positive values. We have also remarked with an asterisk $*$ those Lyapunov exponent that are statistically significant at the 99% confidence level. For the estimation based on blocking methods median values are presented. For the block length M we use $M = int\left[c \times (n/\log n)^{1/6}\right]$ with $c = 36.2$ where $int\,[A]$ signifies the integer part of A. The number of blocks B depends on the sample size n of each currency pair.

EURCHF	[3,1,4]	[4,1,9]	[5,1,6]	[6,1,10]	[7,1,10]	[8,1,10]	[9,1,4]	[10,1,4]
$\hat{\lambda}_F$	-0.00159	-0.00003	-0.00019	-1.07765	**0.37030**	-0.00032	-4.14129	-3.80575
	(9.5e-04)	(2.7e-05)	(7.1e-06)	(4.0e-05)	(2.2e-04)*	(1.7e-05)	(2.3e-04)	(2.7e-04)
$\hat{\lambda}_N$	**0.00025**	**0.00309**	**0.00286**	-1.05855	**0.36021**	**0.00549**	-4.01148	-3.68775
	(1.9e-03)	(3.0e-03)	(3.4e-03)	(3.3e-02)	(7.9e-03)*	(4.6e-03)	(6.3e-01)	(5.3e-01)
$\hat{\lambda}_E$	**0.00021**	**0.00310**	**0.00260**	-1.05857	**0.35997**	**0.00547**	-4.01110	-3.68723
	(1.7e-03)	(2.8e-03)	(3.3e-03)	(3.1e-02)	(7.7e-03)*	(4.4e-03)	(6.2e-01)	(5.1e-01)
$\hat{\lambda}_B$	**0.00019**	**0.00312**	**0.00243**	-1.05910	**0.36013**	**0.00530**	-4.01070	-3.68700
	(1.7e-03)	(2.7e-03)	(3.3e-03)	(3.0e-02)	(7.5e-03)*	(4.3e-03)	(6.1e-01)	(5.0e-01)
EURGBP	[3,1,9]	[4,1,6]	[5,1,5]	[6,1,9]	[7,1,8]	[8,1,10]	[9,1,9]	[10,1,6]
$\hat{\lambda}_F$	**0.00295**	-0.00628	**0.00191**	-0.00699	**0.00172**	-0.00039	-0.00297	-0.00253
	(3.6e-04)*	(2.0e-03)	(9.6e-05)*	(2.9e-03)	(7.2e-05)*	(2.9e-05)	(1.2e-04)	(2.5e-04)
$\hat{\lambda}_N$	**0.00454**	-0.00375	**0.00495**	-0.00294	**0.00564**	**0.00397**	**0.00246**	**0.00306**
	(1.6e-03)	(2.5e-03)	(2.9e-03)	(3.3e-03)	(2.9e-04)*	(3.0e-03)	(3.5e-03)	(3.1e-03)
$\hat{\lambda}_E$	**0.00457**	-0.00391	**0.00501**	-0.00331	**0.00559**	**0.00394**	**0.00263**	**0.00278**
	(3.3e-03)	(3.3e-03)	(3.1e-03)	(4.1e-03)	(2.9e-04)*	(3.6e-03)	(4.3e-03)	(3.2e-03)
$\hat{\lambda}_B$	**0.00461**	-0.00402	**0.00504**	-0.00349	**0.00554**	**0.00393**	**0.00271**	**0.00270**
	(3.2e-03)	(3.3e-03)	(3.1e-03)	(4.1e-03)	(2.9e-04)*	(3.7e-03)	(4.2e-03)	(3.2e-03)
EURJPY	[3,1,4]	[4,1,9]	[5,1,7]	[6,1,7]	[7,1,10]	[8,1,10]	[9,1,9]	[10,1,7]
$\hat{\lambda}_F$	-0.00121	-0.00070	-0.00100	-0.00096	**0.31488**	-0.00031	-0.00094	**0.09524**
	(1.9e-05)	(1.6e-05)	(8.7e-06)	(7.4e-06)	(1.5e-04)*	(5.1e-05)	(9.4e-06)	(4.4e-04)*
$\hat{\lambda}_N$	**0.00048**	**0.00196**	**0.00223**	**0.00314**	**0.30518**	**0.00487**	**0.00467**	**0.08872**
	(1.7e-03)	(1.4e-03)	(3.2e-03)	(3.4e-03)	(5.0e-03)*	(3.9e-03)	(3.5e-03)	(5.9e-03)*
$\hat{\lambda}_E$	**0.00050**	**0.00214**	**0.00244**	**0.00330**	**0.30540**	**0.00496**	**0.00488**	**0.08869**
	(1.7e-03)	(1.4e-03)	(3.1e-03)	(3.3e-03)	(4.9e-03)*	(3.8e-03)	(3.4e-03)	(5.7e-03)*
$\hat{\lambda}_B$	**0.00059**	**0.00249**	**0.00259**	**0.00362**	**0.30588**	**0.00513**	**0.00497**	**0.08874**
	(1.7e-03)	(1.4e-03)	(3.0e-03)	(3.1e-03)	(4.9e-03)*	(3.8e-03)	(3.3e-03)	(5.6e-03)*
EURUSD	[3,1,9]	[4,1,8]	[5,1,9]	[6,1,7]	[7,1,10]	[8,1,9]	[9,1,8]	[10,1,4]
$\hat{\lambda}_F$	-0.00020	-0.00028	**0.00036**	-0.00044	-0.00061	-0.00078	-0.00080	-0.00082
	(5.8e-06)	(8.7e-06)	(2.2e-05)*	(1.1e-05)	(1.1e-05)	(1.3e-05)	(1.4e-05)	(1.3e-05)
$\hat{\lambda}_N$	**0.00268**	**0.00410**	**0.00577**	**0.00597**	**0.00635**	**0.00716**	**0.00759**	**0.00825**
	(2.7e-03)	(4.2e-03)	(4.4e-03)	(4.6e-03)	(4.4e-03)	(4.8e-03)	(5.0e-03)	(4.9e-03)
$\hat{\lambda}_E$	**0.00279**	**0.00433**	**0.00582**	**0.00614**	**0.00659**	**0.00730**	**0.00774**	**0.00840**
	(2.6e-03)	(4.1e-03)	(4.3e-03)	(4.4e-03)	(4.3e-03)	(4.7e-03)	(4.8e-03)	(4.7e-03)
$\hat{\lambda}_B$	**0.00292**	**0.00460**	**0.00597**	**0.00648**	**0.00682**	**0.00753**	**0.00797**	**0.00861**
	(2.6e-03)	(4.1e-03)	(4.3e-03)	(4.3e-03)	(4.3e-03)	(4.7e-03)	(4.8e-03)	(4.7e-03)
NZDUSD	[3,1,9]	[4,1,10]	[5,1,6]	[6,1,8]	[7,1,9]	[8,1,6]	[9,1,5]	[10,1,6]
$\hat{\lambda}_F$	-0.00289	**0.00308**	**0.00299**	**0.00441**	**0.08501**	-0.00213	-0.00007	-0.00041
	(1.1e-03)	(1.7e-04)*	(4.6e-04)*	(3.0e-04)*	(6.4e-04)*	(4.6e-05)	(3.6e-05)	(5.8e-05)
$\hat{\lambda}_N$	**0.00044**	**0.00751**	**0.00673**	**0.01092**	**0.08241**	**0.00398**	**0.00633**	**0.00762**
	(3.2e-03)	(4.9e-03)	(3.7e-03)	(5.9e-02)	(4.3e-03)*	(5.3e-03)	(5.2e-03)	(4.6e-03)
$\hat{\lambda}_E$	**0.00042**	**0.00765**	**0.00164**	**0.01685**	**0.08308**	**0.00315**	**0.00631**	**0.00745**
	(3.3e-03)	(5.1e-03)	(2.8e-03)	(1.0e-02)	(7.4e-03)*	(5.6e-03)	(5.2e-03)	(4.5e-03)
$\hat{\lambda}_B$	**0.00043**	**0.00769**	**0.00169**	**0.01684**	**0.08281**	**0.00317**	**0.00631**	**0.00744**
	(3.3e-03)	(5.1e-03)	(2.8e-03)	(1.0e-02)	(7.9e-03)*	(5.3e-03)	(5.2e-03)	(4.5e-03)

GBPJPY	[3,1,7]	[4,1,9]	[5,1,6]	[6,1,8]	[7,1,4]	[8,1,10]	[9,1,9]	[10,1,7]
$\hat{\lambda}_F$	-0.00253	-0.00218	-0.00224	-0.00247	-0.00252	-0.00229	-0.00236	-0.00245
	(8.5e-06)	(3.4e-05)	(1.8e-05)	(2.9e-05)	(1.3e-05)	(3.3e-05)	(7.5e-05)	(1.5e-05)
$\hat{\lambda}_N$	-0.00055	**0.00096**	**0.00155**	**0.00224**	**0.00287**	**0.00319**	**0.00372**	**0.00405**
	(2.0e-03)	(3.1e-03)	(3.6e-03)	(3.9e-03)	(3.8e-03)	(4.0e-03)	(3.7e-03)	(3.6e-03)
$\hat{\lambda}_E$	-0.00032	**0.00110**	-0.00016	**0.00250**	**0.00299**	**0.00362**	**0.00389**	**0.00420**
	(1.9e-03)	(3.0e-03)	(2.8e-03)	(3.8e-03)	(3.7e-03)	(3.8e-03)	(3.6e-03)	(3.4e-03)
$\hat{\lambda}_B$	-0.00020	**0.00132**	-0.00009	**0.00272**	**0.00315**	**0.00386**	**0.00411**	**0.00440**
	(1.9e-03)	(3.0e-03)	(2.8e-03)	(3.8e-03)	(3.7e-03)	(3.8e-03)	(3.6e-03)	(3.3e-03)
GBPUSD	[3,1,8]	[4,1,8]	[5,1,8]	[6,1,8]	[7,1,10]	[8,1,10]	[9,1,9]	[10,1,7]
$\hat{\lambda}_F$	**0.00103**	**0.00005**	-0.00069	-0.00029	**0.03069**	**0.00082**	-0.00188	**0.01143**
	(5.1e-04)*	(4.0e-06)*	(1.7e-04)	(6.1e-06)	(2.5e-02)*	(1.2e-05)*	(2.8e-05)	(7.5e-04)*
$\hat{\lambda}_N$	**0.00317**	**0.00249**	**0.00219**	**0.00338**	**0.03394**	**0.00549**	**0.00356**	**0.01664**
	(2.1e-03)	(2.6e-03)	(3.1e-03)	(3.4e-03)	(3.4e-03)*	(3.5e-03)	(3.5e-03)	(3.7e-03)*
$\hat{\lambda}_E$	**0.00336**	**0.00267**	**0.00224**	**0.00360**	**0.00744**	**0.00554**	**0.00367**	**0.01785**
	(1.7e-03)	(2.5e-03)	(3.0e-03)	(3.3e-03)	(5.4e-04)*	(3.4e-03)	(3.3e-03)	(3.6e-03)*
$\hat{\lambda}_B$	**0.00352**	**0.00288**	**0.00249**	**0.00384**	**0.00927**	**0.00573**	**0.00382**	**0.01797**
	(1.7e-03)	(2.4e-03)	(3.0e-03)	(3.3e-03)	(6.4e-04)*	(3.4e-03)	(3.2e-03)	(3.6e-03)*
USDCAD	[3,1,3]	[4,1,9]	[5,1,7]	[6,1,2]	[7,1,10]	[8,1,10]	[9,1,8]	[10,1,2]
$\hat{\lambda}_F$	-0.02153	-0.15190	**0.03936**	-6.75103	-0.01487	-0.01567	-0.01118	**0.05012**
	(9.2e-06)	(5.1e-03)	(6.0e-03)*	(3.2e-04)	(1.8e-05)	(2.0e-05)	(8.1e-05)	(2.0e-03)*
$\hat{\lambda}_N$	-0.01872	-0.14033	**0.04711**	-6.56416	-0.00977	-0.01004	-0.00549	**0.05166**
	(2.7e-03)	(1.7e-02)	(1.4e-02)	(7.5e-01)	(4.5e-03)	(4.5e-03)	(4.5e-03)	(7.4e-03)*
$\hat{\lambda}_E$	-0.01872	-0.13578	**0.02374**	-6.56416	-0.00983	-0.01002	-0.00530	**0.02470**
	(2.7e-03)	(4.4e-02)	(3.5e-02)	(7.5e-01)	(4.4e-03)	(4.4e-03)	(4.5e-03)	(1.2e-03)*
$\hat{\lambda}_B$	-0.01872	-0.13899	**0.02487**	-6.56416	-0.00985	-0.01002	-0.00504	**0.02532**
	(2.7e-03)	(4.4e-02)	(3.4e-02)	(7.5e-01)	(4.4e-03)	(4.4e-03)	(4.5e-03)	(1.2e-03)*
USDCHF	[3,1,4]	[4,1,9]	[5,1,10]	[6,1,7]	[7,1,8]	[8,1,5]	[9,1,7]	[10,1,7]
$\hat{\lambda}_F$	-0.00173	-0.00119	-0.00043	-0.00086	-0.00141	-0.00058	-0.64682	**0.15933**
	(9.0e-06)	(1.0e-05)	(8.5e-05)	(1.6e-05)	(1.4e-05)	(2.5e-05)	(4.0e-05)	(6.0e-04)*
$\hat{\lambda}_N$	**0.00081**	**0.00209**	**0.00355**	**0.00377**	**0.00323**	**0.00427**	-0.62895	**0.15226**
	(2.4e-03)	(3.3e-03)	(3.8e-03)	(4.0e-03)	(3.9e-03)	(4.8e-03)	(6.8e-02)	(5.9e-03)*
$\hat{\lambda}_E$	**0.00081**	**0.00216**	**0.00349**	**0.00380**	**0.00320**	**0.00454**	-0.62895	**0.15279**
	(2.3e-03)	(3.2e-03)	(3.7e-03)	(3.9e-03)	(3.8e-03)	(4.7e-03)	(6.7e-02)	(5.6e-03)*
$\hat{\lambda}_B$	**0.00081**	**0.00232**	**0.00352**	**0.00380**	**0.00320**	**0.00473**	-0.62895	**0.15325**
	(2.3e-03)	(3.2e-03)	(3.6e-03)	(3.9e-03)	(3.8e-03)	(4.7e-03)	(6.7e-02)	(5.5e-03)*
USDJPY	[3,1,7]	[4,1,9]	[5,1,6]	[6,1,7]	[7,1,10]	[8,1,5]	[9,1,8]	[10,1,7]
$\hat{\lambda}_F$	-0.01334	**0.17763**	-0.00099	-0.00089	**0.36246**	**0.02696**	**0.00331**	**0.14069**
	(5.3e-03)	(3.5e-04)*	(1.1e-05)	(1.4e-05)	(5.1e-03)*	(2.2e-04)*	(1.5e-04)*	(2.6e-04)*
$\hat{\lambda}_N$	-0.01022	**0.17487**	**0.00296**	**0.00388**	**0.36575**	**0.02348**	**0.00745**	**0.13352**
	(3.0e-03)	(1.5e-03)*	(3.6e-03)	(4.0e-03)	(4.8e-03)*	(4.0e-03)*	(4.2e-03)	(8.7e-03)*
$\hat{\lambda}_E$	-0.01013	**0.17479**	**0.00296**	**0.00387**	**0.36713**	**0.01949**	**0.00738**	**0.13393**
	(7.8e-03)	(1.5e-03)*	(3.6e-03)	(4.0e-03)	(4.5e-03)*	(5.2e-03)*	(4.2e-03)	(1.0e-03)*
$\hat{\lambda}_B$	-0.01000	**0.17485**	0.00296	0.00387	**0.36957**	**0.01922**	0.00733	**0.13370**
	(7.8e-03)	(1.5e-03)*	(3.6e-03)	(4.0e-03)	(4.4e-03)*	(5.2e-03)*	(4.2e-03)	(1.0e-03)*

AUDJPY	[3,1,3]	[4,1,10]	[5,1,2]	[6,1,10]	[7,1,5]	[8,1,10]	[9,1,8]	[10,1,7]
$\hat{\lambda}_F$	-0.01850	-0.02368	-0.01269	-0.01559	-0.02013	-0.01440	-0.00891	-0.00835
	(4.7e-05)	(6.5e-05)	(6.9e-05)	(8.1e-05)	(8.7e-05)	(8.5e-05)	(7.1e-05)	(8.0e-05)
$\hat{\lambda}_N$	-0.01364	-0.01691	-0.00578	-0.00682	-0.01061	-0.00587	-0.00230	-0.00060
	(4.6e-03)	(6.6e-03)	(6.6e-03)	(6.5e-03)	(6.9e-03)	(6.3e-03)	(7.9e-03)	(6.4e-03)
$\hat{\lambda}_E$	-0.01355	-0.01646	-0.00590	-0.00671	-0.01057	-0.00587	-0.00222	-0.00056
	(4.5e-03)	(6.3e-03)	(6.4e-03)	(6.2e-03)	(6.7e-03)	(6.3e-03)	(7.7e-03)	(6.3e-03)
$\hat{\lambda}_B$	-0.01332	-0.01619	-0.00581	-0.00677	-0.01043	-0.00587	-0.00204	-0.00047
	(4.3e-03)	(6.2e-03)	(6.3e-03)	(6.1e-03)	(6.6e-03)	(6.3e-03)	(7.5e-03)	(6.1e-03)
AUDUSD	[3,1,4]	[4,1,9]	[5,1,7]	[6,1,7]	[7,1,10]	[8,1,10]	[9,1,8]	[10,1,2]
$\hat{\lambda}_F$	-0.00222	-0.00184	-0.00210	-0.00193	**0.23394**	**0.18002**	**0.16010**	**0.12989**
	(2.6e-05)	(1.4e-05)	(2.0e-05)	(1.8e-05)	(1.2e-03)*	(1.5e-03)*	(1.0e-03)*	(1.7e-03)*
$\hat{\lambda}_N$	-4.6e-06	**0.00164**	**0.00253**	**0.00340**	**0.23587**	**0.18045**	**0.15787**	**0.12622**
	(2.2e-03)	(3.4e-03)	(4.1e-03)	(4.0e-03)	(5.9e-03)*	(7.2e-03)*	(3.2e-03)*	(7.1e-03)*
$\hat{\lambda}_E$	-0.00001	**0.00165**	**0.00252**	**0.00341**	**0.23755**	**0.16294**	**0.15488**	**0.10748**
	(2.3e-03)	(3.2e-03)	(4.0e-03)	(4.0e-03)	(8.8e-03)*	(1.3e-03)*	(5.0e-03)*	(1.2e-03)*
$\hat{\lambda}_B$	-5.4e-06	**0.00160**	**0.00248**	**0.00338**	**0.23551**	**0.16243**	**0.15413**	**0.10737**
	(2.3e-03)	(3.4e-03)	(4.1e-03)	(4.0e-03)	(8.1e-03)*	(1.3e-03)*	(4.6e-03)*	(1.1e-03)*
CADJPY	[3,1,7]	[4,1,9]	[5,1,6]	[6,1,7]	[7,1,10]	[8,1,10]	[9,1,9]	[10,1,2]
$\hat{\lambda}_F$	**0.00123**	-1.26220	**0.00277**	-0.00167	**0.62789**	-5.44971	-0.03877	**0.17453**
	(2.1e-05)*	(7.1e-05)	(1.9e-05)*	(5.0e-05)	(2.1e-02)*	(1.1e-03)	(1.3e-02)	(9.0e-04)*
$\hat{\lambda}_N$	**0.00459**	-1.24807	**0.00542**	**0.00731**	**0.62853**	-5.32452	-0.04243	**0.17184**
	(3.2e-03)	(1.2e-02)	(4.4e-03)	(6.1e-03)	(1.2e-02)*	(5.2e-01)	(9.5e-03)	(2.7e-03)*
$\hat{\lambda}_E$	**0.00468**	-1.24800	**0.00533**	**0.00731**	**0.63251**	-5.32452	-0.04904	**0.17182**
	(3.0e-03)	(1.1e-02)	(4.2e-03)	(6.1e-03)	(5.1e-02)*	(5.1e-01)	(2.0e-02)	(6.4e-03)*
$\hat{\lambda}_B$	**0.00440**	-1.24815	**0.00524**	**0.00731**	**0.63320**	-5.32452	-0.04935	**0.17253**
	(3.1e-03)	(1.4e-02)	(4.6e-03)	(6.1e-03)	(5.1e-02)*	(4.9e-01)	(2.2e-02)	(6.1e-03)*
CHFJPY	[3,1,3]	[4,1,9]	[5,1,6]	[6,1,6]	[7,1,10]	[8,1,10]	[9,1,8]	[10,1,2]
$\hat{\lambda}_F$	-0.00129	-0.03035	-0.00098	-0.00231	-6.9e-06	-0.00917	**0.19180**	**0.16976**
	(7.4e-06)	(6.7e-03)	(1.2e-05)	(3.4e-04)	(9.4e-05)	(2.2e-03)	(1.1e-03)*	(6.6e-04)*
$\hat{\lambda}_N$	**0.00087**	-0.02099	**0.00263**	**0.00157**	**0.00529**	-0.00305	**0.18881**	**0.16452**
	(2.2e-03)	(8.6e-03)	(3.7e-03)	(3.8e-03)	(4.5e-03)	(4.3e-03)	(3.0e-03)*	(2.5e-03)*
$\hat{\lambda}_E$	**0.00053**	-0.01719	**0.00254**	**0.00180**	**0.00419**	-0.00193	**0.18527**	**0.16600**
	(2.4e-03)	(3.2e-02)	(3.1e-03)	(4.1e-03)	(4.3e-03)	(8.4e-03)	(8.2e-03)*	(6.1e-03)*
$\hat{\lambda}_B$	**0.00069**	-0.01783	**0.00266**	**0.00171**	**0.00423**	-0.00222	**0.18560**	**0.16617**
	(2.3e-03)	(3.5e-02)	(3.7e-03)	(4.6e-03)	(4.6e-03)	(8.5e-03)	(8.8e-03)*	(6.2e-03)*

Table 6.4.3: The Lyapunov exponent values based on several blocking methods by each currency pair. Intermediate results pertaining to the lowest day by volume of tick-by-tick data are shown. Numbers in parentheses are standard errors. Into brackets denote $[m, \tau, h]$. We have indicated in bold those Lyapunov exponent with positive values. We have also remarked with an asterisk $*$ those Lyapunov exponent that are statistically significant at the 99% confidence level. For the estimation based on blocking methods median values are presented. For the block length M we use $M = int\left[c \times (n/\log n)^{1/6}\right]$ with $c = 36.2$ where $int\,[A]$ signifies the integer part of A. The number of blocks B depends on the sample size n of each currency pair.

EURCHF	[3,1,8]	[4,1,9]	[5,1,6]	[6,1,7]	[7,1,10]	[8,1,10]	[9,1,9]	[10,1,2]
$\hat{\lambda}_F$	-0.00323	-0.01013	-0.00242	-0.00243	-0.00248	**0.16506**	-0.00456	**0.06194**
	(1.6e-05)	(8.1e-03)	(1.1e-05)	(1.6e-05)	(2.5e-05)	(3.0e-03)*	(2.0e-05)	(1.1e-03)*
$\hat{\lambda}_N$	-0.00104	-0.00243	**0.00087**	**0.00155**	**0.00229**	**0.16120**	**0.00122**	**0.05823**
	(2.2e-03)	(8.6e-03)	(3.8e-03)	(4.0e-03)	(4.4e-03)	(3.2e-03)*	(3.9e-03)	(1.9e-03)*
$\hat{\lambda}_E$	-0.00104	-0.00082	**0.00076**	**0.00154**	**0.00226**	**0.16179**	**0.00134**	**0.05617**
	(2.1e-03)	(5.4e-02)	(3.8e-03)	(4.2e-03)	(4.8e-03)	(3.6e-03)*	(3.2e-03)	(5.1e-03)*
$\hat{\lambda}_B$	-0.00113	-0.00133	**0.00090**	**0.00158**	**0.00225**	**0.16177**	**0.00144**	**0.05677**
	(2.0e-03)	(5.2e-02)	(3.5e-03)	(4.0e-03)	(4.4e-03)	(3.9e-03)*	(3.9e-03)	(5.3e-03)*
EURGBP	[3,1,7]	[4,1,9]	[5,1,7]	[6,1,7]	[7,1,8]	[8,1,10]	[9,1,8]	[10,1,7]
$\hat{\lambda}_F$	-0.00091	-0.00067	-0.00066	-0.00067	-0.00063	**0.02817**	**0.42361**	**0.10923**
	(6.8e-04)	(1.5e-05)	(1.8e-05)	(1.7e-05)	(2.7e-05)	(9.9e-04)*	(1.2e-05)*	(9.5e-05)*
$\hat{\lambda}_N$	**0.00124**	**0.00288**	**0.00394**	**0.00475**	**0.00524**	**0.04684**	**0.41981**	**0.10013**
	(2.2e-03)	(3.4e-03)	(4.0e-03)	(4.2e-03)	(3.e-038)	(1.5e-02)	(2.6e-03)*	(7.1e-03)*
$\hat{\lambda}_E$	**0.00123**	**0.00288**	**0.00395**	**0.00476**	**0.00528**	**0.05013**	**0.41981**	**0.09987**
	(2.2e-03)	(3.4e-03)	(4.0e-03)	(4.2e-03)	(3.8e-03)	(2.0e-02)	(2.5e-03)*	(6.2e-03)*
$\hat{\lambda}_B$	**0.00121**	**0.00287**	**0.00396**	**0.00476**	**0.00530**	**0.04952**	**0.41980**	**0.09998**
	(2.2e-03)	(3.4e-03)	(4.0e-03)	(4.2e-03)	(3.8e-03)	(2.0e-02)	(2.5e-03)*	(6.3e-03)*
EURJPY	[3,1,8]	[4,1,10]	[5,1,3]	[6,1,7]	[7,1,10]	[8,1,9]	[9,1,5]	[10,1,7]
$\hat{\lambda}_F$	-0.00221	-0.00160	-0.00057	-0.00106	-0.00175	**0.00666**	**0.04076**	**0.10118**
	(8.4e-06)	(1.2e-05)	(1.6e-05)	(1.8e-05)	(2.2e-05)	(2.1e-05)*	(2.1e-05)*	(1.5e-05)*
$\hat{\lambda}_N$	**0.00036**	**0.00212**	**0.00463**	**0.00468**	**0.00506**	**0.00787**	**0.03649**	**0.09698**
	(2.5e-03)	(3.6e-03)	(4.3e-03)	(4.2e-03)	(5.5e-03)	(6.0e-03)	(4.2e-03)*	(3.9e-03)*
$\hat{\lambda}_E$	**0.00040**	**0.00201**	**0.00448**	**0.00452**	**0.00512**	**0.00786**	**0.03624**	**0.09634**
	(2.3e-03)	(3.7e-03)	(4.1e-03)	(4.1e-03)	(5.7e-03)	(6.1e-03)	(4.1e-03)*	(3.3e-03)*
$\hat{\lambda}_B$	**0.00023**	**0.00227**	**0.00451**	**0.00447**	**0.00567**	**0.00754**	**0.03687**	**0.09676**
	(2.2e-03)	(3.8e-03)	(4.0e-03)	(4.5e-03)	(5.3e-03)	(6.3e-03)	(4.0e-03)*	(3.7e-03)*
EURUSD	[3,1,4]	[4,1,9]	[5,1,7]	[6,1,10]	[7,1,8]	[8,1,5]	[9,1,9]	[10,1,7]
$\hat{\lambda}_F$	-0.00551	**0.00020**	-0.00090	-0.52460	**0.16062**	**0.07029**	**0.14388**	**0.20320**
	(5.1e-04)	(1.5e-05)*	(1.4e-05)	(8.7e-05)	(1.9e-05)*	(2.1e-03)*	(7.5e-04)*	(1.2e-03)*
$\hat{\lambda}_N$	-0.00320	**0.00445**	**0.00189**	-0.51479	**0.15407**	**0.07265**	**0.14191**	**0.19798**
	(2.3e-03)	(4.3e-03)	(2.9e-03)	(8.9e-03)	(4.5e-03)*	(9.4e-03)*	(1.2e-02)*	(1.3e-02)*
$\hat{\lambda}_E$	-0.00319	**0.00445**	**0.00189**	-0.51472	**0.15408**	**0.06953**	**0.14017**	**0.18523**
	(2.3e-03)	(4.3e-03)	(2.9e-03)	(8.9e-03)	(4.5e-03)*	(2.7e-03)*	(1.2e-02)*	(6.0e-03)*
$\hat{\lambda}_B$	-0.00321	**0.00445**	**0.00189**	-0.51475	**0.15407**	**0.06759**	**0.14013**	**0.18490**
	(2.3e-03)	(4.3e-03)	(2.9e-03)	(8.9e-03)	(4.5e-03)*	(2.7e-03)*	(1.2e-02)*	(7.0e-03)*
NZDUSD	[3,1,4]	[4,1,9]	[5,1,10]	[6,1,10]	[7,1,8]	[8,1,5]	[9,1,9]	[10,1,7]
$\hat{\lambda}_F$	-0.00200	-0.00268	**0.00099**	**0.33450**	-0.00152	**0.29497**	-0.00216	**0.33312**
	(2.0e-05)	(5.6e-05)	(1.0e-03)	(5.6e-03)*	(8.5e-05)	(8.2e-04)*	(7.0e-05)	(2.0e-02)*
$\hat{\lambda}_N$	-0.00007	**0.00202**	**0.00659**	**0.33119**	**0.00504**	**0.29438**	**0.00613**	**0.33550**
	(2.0e-03)	(4.5e-03)	(3.9e-03)	(3.5e-03)*	(4.8e-03)	(3.4e-03)*	(6.4e-03)	(1.1e-02)*
$\hat{\lambda}_E$	-0.00012	**0.00200**	**0.00701**	**0.33131**	**0.00514**	**0.29734**	**0.00614**	**0.33799**
	(2.2e-03)	(4.4e-03)	(3.9e-03)	(8.7e-03)*	(4.7e-03)	(7.6e-03)*	(6.3e-03)	(6.0e-02)*
$\hat{\lambda}_B$	-0.00014	**0.00195**	**0.00693**	**0.33097**	**0.00513**	**0.29703**	**0.00614**	**0.33718**
	(2.3e-03)	(4.3e-03)	(3.8e-03)	(8.6e-03)*	(4.7e-03)	(7.7e-03)*	(6.1e-03)	(6.1e-02)*

GBPJPY	[3,1,7]	[4,1,9]	[5,1,9]	[6,1,5]	[7,1,10]	[8,1,3]	[9,1,9]	[10,1,6]
$\hat{\lambda}_F$	-0.00241	-0.00261	-0.00473	-0.00049	-0.00081	-0.00043	-0.00139	**0.02425**
	(2.6e-05)	(3.2e-05)	(6.1e-05)	(4.6e-05)	(5.0e-05)	(6.1e-05)	(6.6e-05)	(4.8e-05)*
$\hat{\lambda}_N$	**0.00077**	**0.00125**	**0.00288**	**0.00520**	**0.00528**	**0.00717**	**0.00647**	**0.02116**
	(3.1e-03)	(3.7e-03)	(5.8e-03)	(4.9e-03)	(5.2e-03)	(6.0e-03)	(5.7e-03)	(1.5e-03)*
$\hat{\lambda}_E$	**0.00063**	**0.00147**	**0.00271**	**0.00512**	**0.00528**	**0.00749**	**0.00612**	**0.02098**
	(3.3e-03)	(3.8e-03)	(5.3e-03)	(4.3e-03)	(5.4e-03)	(6.2e-03)	(5.2e-03)	(1.0e-03)*
$\hat{\lambda}_B$	**0.00056**	**0.00118**	**0.00259**	**0.00598**	**0.00528**	**0.00782**	**0.00626**	**0.02114**
	(3.2e-03)	(3.5e-03)	(5.1e-03)	(4.6e-03)	(5.8e-03)	(6.1e-03)	(5.5e-03)	(1.1e-03)*
GBPUSD	[3,1,7]	[4,1,9]	[5,1,10]	[6,1,10]	[7,1,8]	[8,1,10]	[9,1,9]	[10,1,7]
$\hat{\lambda}_F$	-0.00437	-0.00199	-0.00266	**0.04706**	-0.00090	-0.01164	**0.02665**	**0.05687**
	(1.1e-04)	(1.3e-05)	(2.6e-05)	(3.9e-02)	(1.4e-05)	(3.7e-03)	(4.5e-03)*	(9.1e-04)*
$\hat{\lambda}_N$	-0.00351	**0.00065**	**0.00032**	**0.05533**	**0.00283**	-0.00994	**0.03286**	**0.05692**
	(1.0e-03)	(2.5e-03)	(3.1e-03)	(7.6e-02)	(3.1e-03)	(4.7e-03)	(5.5e-03)*	(5.8e-03)*
$\hat{\lambda}_E$	-0.00351	**0.00050**	**0.00002**	**0.05293**	**0.00282**	-0.00952	**0.02690**	**0.04407**
	(1.1e-03)	(2.6e-03)	(2.9e-03)	(3.1e-02)	(3.0e-03)	(5.4e-03)	(1.2e-03)*	(8.1e-03)*
$\hat{\lambda}_B$	-0.00339	**0.00044**	**0.00004**	**0.05198**	**0.00281**	-0.00952	**0.02686**	**0.04418**
	(1.1e-03)	(2.6e-03)	(2.9e-03)	(3.2e-02)	(3.0e-03)	(5.5e-03)	(1.2e-03)*	(8.0e-03)*
USDCAD	[3,1,3]	[4,1,8]	[5,1,6]	[6,1,6]	[7,1,10]	[8,1,4]	[9,1,9]	[10,1,2]
$\hat{\lambda}_F$	**0.46185**	-24.47825	**0.35679**	**0.49990**	**0.61118**	**0.23541**	**0.30535**	**0.18213**
	(1.5e-05)*	(1.2e-03)	(1.8e-05)*	(2.8e-03)*	(2.6e-02)*	(4.3e-03)*	(1.0e-02)*	(7.3e-04)*
$\hat{\lambda}_N$	**0.45933**	-24.09700	**0.35666**	**0.49756**	**0.61610**	**0.23386**	**0.30477**	**0.17927**
	(2.2e-03)*	(3.0e-01)	(1.9e-02)*	(3.5e-03)*	(8.6e-03)*	(4.3e-03)*	(7.1e-03)*	(2.8e-03)*
$\hat{\lambda}_E$	**0.45921**	-24.09732	**0.35683**	**0.49570**	**0.61349**	**0.22960**	**0.30664**	**0.18147**
	(2.5e-03)*	(3.1e-01)	(1.5e-02)*	(5.7e-03)*	(6.5e-02)*	(1.9e-02)*	(2.9e-02)*	(6.9e-03)*
$\hat{\lambda}_B$	**0.45952**	-24.09756	**0.35611**	**0.49621**	**0.61437**	**0.22943**	**0.30498**	**0.18084**
	(2.1e-03)*	(3.3e-01)	(1.7e-02)*	(5.6e-03)*	(6.7e-02)*	(1.9e-02)*	(3.1e-02)*	(6.8e-03)*
USDCHF	[3,1,3]	[4,1,9]	[5,1,6]	[6,1,7]	[7,1,10]	[8,1,10]	[9,1,9]	[10,1,2]
$\hat{\lambda}_F$	-0.00051	**0.02524**	-0.00018	**0.00012**	**0.50712**	**0.19928**	**0.11949**	**0.14118**
	(6.0e-06)	(4.5e-03)*	(1.0e-05)	(1.6e-05)*	(5.4e-03)*	(2.3e-03)*	(2.1e-04)*	(1.1e-03)*
$\hat{\lambda}_N$	**0.00084**	**0.03446**	**0.00213**	**0.00212**	**0.50986**	**0.19629**	**0.10928**	**0.13634**
	(1.4e-03)	(1.7e-02)	(2.9e-03)	(3.2e-03)	(4.6e-02)*	(6.5e-03)*	(5.1e-03)*	(2.1e-03)*
$\hat{\lambda}_E$	**0.00070**	**0.04082**	**0.00224**	**0.00289**	**0.49451**	**0.17871**	**0.10893**	**0.13615**
	(1.7e-03)	(3.8e-02)	(3.0e-03)	(3.2e-03)	(4.2e-02)*	(9.8e-03)*	(4.1e-03)*	(5.0e-03)*
$\hat{\lambda}_B$	**0.00062**	**0.03408**	**0.00246**	**0.00291**	**0.49359**	**0.17882**	**0.10885**	**0.13596**
	(1.9e-03)	(3.5e-02)	(3.2e-03)	(3.2e-03)	(4.0e-02)*	(1.0e-02)*	(4.0e-03)*	(5.1e-03)*
USDJPY	[3,1,7]	[4,1,10]	[5,1,6]	[6,1,6]	[7,1,10]	[8,1,10]	[9,1,9]	[10,1,2]
$\hat{\lambda}_F$	-0.00001	**0.08804**	**0.11246**	**0.18811**	**0.15643**	**0.3365**	**0.08192**	**0.12525**
	(3.9e-05)	(1.3e-03)*	(1.5e-05)*	(2.8e-05)*	(1.7e-05)*	(2.6e-04)*	(9.0e-05)*	(4.7e-04)*
$\hat{\lambda}_N$	**0.00646**	**0.08341**	**0.11326**	**0.18460**	**0.15557**	**0.33253**	**0.08391**	**0.12112**
	(6.1e-03)	(4.6e-03)*	(5.2e-03)*	(6.5e-03)*	(1.1e-02)*	(3.9e-03)*	(8.9e-03)*	(1.9e-03)*
$\hat{\lambda}_E$	**0.00650**	**0.08400**	**0.11322**	**0.18447**	**0.15523**	**0.33091**	**0.08384**	**0.12132**
	(6.2e-03)	(9.8e-03)*	(5.3e-03)*	(6.6e-03)*	(1.4e-02)*	(4.5e-03)*	(9.0e-03)*	(3.8e-03)*
$\hat{\lambda}_B$	**0.00652**	**0.08350**	**0.11375**	**0.18492**	**0.15510**	**0.33095**	**0.08385**	**0.12149**
	(6.4e-03)	(9.7e-03)*	(5.1e-03)*	(6.8e-03)*	(1.0e-02)*	(4.4e-03)*	(9.3e-03)*	(3.5e-03)*

6.5 Discussion

The theoretical framework in economics and finance during the last decades has been based on (at least) the following assumptions: (*i*) economy tends towards a determinate and intrinsically steady state in the absence of exogenous shocks; (*ii*) the economic agents, resumed in the behaviour of representative agent, are described as rational individuals which maximise their utility or profits on the basis of complete information about the market; (*iii*) the ceteris paribus clause is commonly used as a response to exogenous and unknown shocks; (*iv*) most economic phenomena are usually explained with linear and comparative static models, even if they are inherently nonlinear, because these equations are easy to manipulate and usually provide unique solutions. Hence when the irregular behaviour of some nonlinear relations is found, it is not appreciated by economists because it is difficult and intractable to deal with. The fact that some simple nonlinear models could show complex dynamics pushed economists to study nonlinear dynamics and chaos theory as a possible framework for modelling economic phenomena.

In this sense it has been more than three decades since ideas from chaos began appearing in the literature showing that it is possible to design economic models in regime of chaotic behaviour from a theoretical point of view. However there is no clear evidence that economic time series behave chaotically. So far researchers have found substantial evidence for nonlinearity but relatively weak evidence for chaos. In this chapter we have cleared up this chaos model-data paradox showing a possible answer to it, at least when studying financial time series. Chaotic systems are sensitive to initial conditions, so temporal dependence in chaotic time series is lost as the series are sampled at too long time intervals, appearing as independent, even though their underlying generator system could come from a chaotic dynamical system. In the case of financial time series, which quotes are continuously traded on markets, the daily sampling may be too long.

We have used 14 currency pairs on the Foreign Exchange Market (FOREX) to test our suggested solution to the chaos model-data paradox. Our main hypothesis is that chaos is elusive in financial empirical studies because of loss of information that occurs when daily quotes are used. This could hinder the detection of chaos in those time series. We propose taking into account all the information available in the financial markets (full sample information on quotes) instead of daily data. If one wants to detect chaotic signals inside financial time series, we also recommend using lagged returns instead of classical daily returns with the purpose of preserving the dynamic dependence on information. In our case we have calculated them considering a lag or order of differentiation less than a day, between 9 and 16 hours, as indicated by the first minimum of the mutual information function for each currency pair studied.

Note that most papers published in financial literature so far deal with time-series data equally spaced in time (uniform time-frequency) both in low-frequency (monthly, weekly, daily) as in high-frequency (1-hour, 30-min, 10-min, 1-min and so on). However the FOREX market is characterised by its almost continuity. In this sense we have considered intra-daily data with non-uniform frequency (tick-by-tick) which are usually not equally spaced in time. This fact entails several difficulties due to process a huge quantity of information and regarding the attractor reconstruction. In this sense we have proposed a new algorithm who extend the reconstruction theorem when the dynamic system is sampled with non-uniform time-frequency. Their results have allowed us to get the non-uniform delayed-coordinate embedding vectors from the tick-by-tick time series. Then we have estimated robustly the Lyapunov exponents and tested the hypothesis of chaos based on the theoretical asymptotic properties of the global neural net estimator. The results provide by this chapter have been obtained using different algorithms from the **DChaos** package. These algorithms are publicly available at `www.CRAN.R-project.org/package=DChaos`.

To sum up, we have been able to detect chaotic signals inside real-world financial time-series data, at least in the 14 currency pairs and days chosen. Nevertheless, we have not intended to generalise this finding to all financial series or even to all the FX rates available. The main interest of this chapter has been to illustrate that by choosing a tick-by-tick frequency instead of a daily one, with the purpose of preserving the dynamic dependence on information, we could find chaos. At least, in the 14 tick-by-tick FX rates and the days considered. As far as we know these kind of FX tick-by-tick rates have never been tested for chaos. This fact could open up a new line of research in which new contributions may appear considering other financial assets, periods and jacobian indirect methods as the local polynomial kernel approach or the local neural net models.

Let us conclude this chapter point out a couple of issues: (*i*) the ability to access and analyse high-frequency dataset provides enormous potential for furthering our understanding of financial markets. Viewing markets as continuous processes may require rethinking many of our underlying precepts; (*ii*) we must keep in mind the following circumstances in order to apply properly chaos theory in non-experimental social sciences as economics or finance. On the one hand, we cannot carry out controlled experiments in the laboratory to study under which situations complex behaviours are presented. On the other hand, we do not know exactly which are the real laws that govern the economic phenomenon, so we cannot substitute experimentation by simulation using its laws of dynamic behaviour. These facts make the field of nonlinear time series analysis a fruitful research framework providing useful tools to build a bridge between chaos theory and tick-by-tick financial time series analysis.

Chapter 7

Conclusion

"¿Qué es el determinismo?; ¿una vanidad, un recurso perfeccionista, el resultado de una simplificación forzada, una aparente necesidad metodológica, una opción estética?" Un mundo poliédrico.

— Andrés Fernández Díaz

Abstract: This chapter presents a summary of the main closing remarks derived from the arguments and results provided by this PhD thesis (Section 7.1). Then we have remarked the main contributions proposed, communications presented at national and international conferences and research stays developed during the elaboration of this doctoral dissertation (Section 7.2). To conclude, some future research questions are briefly discussed in Section 7.3.

As we said at the beginning quantifying chaos from time-series data through the so-called *Lyapunov exponents* in a rigorous fashion is far from being a trivial exercise and poses a number of theoretical and practical challenges nowadays. This PhD thesis has led us through a progressive discussion of those theoretical and computational aspects which constitute, at present time, the main and successful ideas about quantifying chaos from time-series data through the Lyapunov exponents by various data-based computational methods. The ideas proposed in this doctoral dissertation has been supported by the statistical analysis, the suitable computational tools and the know-how about chaotic dynamic systems and nonlinear time series analysis. Let us provide the main conclusions that have reported the theoretical and empirical results obtained by each chapter of this PhD thesis. Particularly, we are going to illustrate an outline relating the objectives proposed initially with the contributions made in this doctoral dissertation.

7.1 Closing remarks

This section provides a summary of the main ideas that have been developed throughout this doctoral dissertation. In this sense it is opportune to make a stop along the way and illustrate an outline relating the aims proposed at the beginning with the contributions made in this PhD thesis as follows.

We were starting this doctoral dissertation remarking that in deterministic dynamic systems is not possible to give a rigorous definition that account for all the aspects that the term chaos implies so far. So we have adopted an operational approach in order to characterise a chaotic dynamic system based on the study of stability. We have focused on local definitions which considers the behaviour of a system inside an attractor once the dynamic systems, i.e., the orbits, have reached some of the dynamic equilibria. In this context as we have seen if a dissipative nonlinear deterministic system is locally unstable, then there is the possibility that no matter how close two initial values are to each other, they will lead to drastically different orbits. Thus, randomness can be generated by a strictly deterministic system.

Remember that this attracting set called strange attractor can be described as a feature of chaos which denote a finite region of the state space by an infinite number of points where an orbit or trajectory inside it will show an aperiodic cycle. That is, there is no regular finite period (tends to infinity) and it is highly unstable. In order to study the stability of an orbit inside this kind of attractor we have focused on the so-called Lyapunov exponents. So quantifying chaos through this kind of quantitative measure is a key point for understanding chaos. In this sense we have provided a detailed review of the state-of-the-art about the Lyapunov exponent. We have also illustrated how it can be used to characterise a chaotic dynamic system.

Since it was realised that even simple and deterministic dynamic systems can produce trajectories which look as a random process, there occurred some obvious questions: how can one distinguish chaos from randomness in real datasets?; how can we extract from time-series data relevant information about the unknown underlying generator system of them?. When dynamic systems are known, we have seen how we can directly calculate both the largest and the full spectrum of the theoretical Lyapunov exponent value to check when the dynamic system is in a regime of chaotic behaviour. However in most real-world observed time series the data-generating process is rarely known a priori. Then we have studied what happens when we assume that the true underlying generator system is unknown. Unfortunately, those theoretical results are not directly applicable when moving into the field of nonlinear time serie analysis.

The empirical analysis of chaotic dynamic systems is based on the study of observed time-series data. Thus if we take into account the empirical environment we do not have the advantage of observing directly the state of the dynamic system let alone knowing the functional form that generates the dynamic associated with it. Instead of that, there will be an observer function that include an additive measurement error which generates observations. As far as this doctoral dissertation is concerned we have considered just only one observable variable. In this sense we have assumed that all information available is a noise-contaminated sequence as a scalar strictly stationary univariate time serie.

Note that we have considered that it is appropriate to add a measurement noise term in the observer function because most real-world observed time-series data are usually noise-contaminated signals. In this case, we wanted to know the effects of considering a normal multinomial measurement noise term on the observation function). The presence of a measurement noise can lead to confusion between a chaotic deterministic system and a purely stochastic one. In such cases the interplay between order and disorder, determinism and stochasticity, stability and instability or predictability and unpredictability is causing profound changes with very exciting implications on nonlinear time series analysis and chaos.

Generally speaking devise a dynamic system to model some time-evolving phenomena involves specifying a rule that governs the evolution of the process through time and the space in which the process takes values. If the rule that governs the evolution of the process is well understood it may be possible to write down the dynamic system more or less from first principles, without much need for observational data except perhaps to fix some parameters. Otherwise we may be much less clear about the underlying rules. That is, we would have to deal with the following situations:

(i) we might suspect that the system is governed by e.g., a set of difference equations, but not know what these are, even we may be uncertain what dynamic variables the equations involve; (ii) we may have available a quantity of experimental data relating to how the system is observed to change through time. Then in this uncertainty context we have considered that all information available is the sequence of measurements in the form of time-series data because we have assumed that the true data-generating process is unknown. Thus it has not been possible to consider the true orbit of the dynamic system in the original state space. Instead of that, we have obtained an approximation (reconstruction) of it that result equivalent in a topological sense (equivalence in the dynamic and geometric properties).

The embedding procedure has provide us a framework to reconstruct an unknown dynamic system which gave rise to a given observed scalar time-series data simply by reconstructing a new state space out of successive values from time series. In this sense, we have provided a detailed review about the attractor reconstruction procedure in order to make sure that the embedding process allows us to get all the relevant information about the unknown underlying dynamic system that generates the time-series data (invariant properties). Particularly, we were interested in knowing if the Lyapunov exponents have approximately the same value in both the true and the reconstructed state space. This fact has allowed us to test the hypothesis of chaos in the unknown original dynamic system.

Note that a key point to create a suitable reconstruction of the state space is to fix a right criteria in order to estimate robustly the embedding parameters: the embedding dimension and the reconstruction time-delay. Researchers in this area usually estimate them using heuristic approaches and prescriptions that mostly rely on physical or geometrical arguments. The main drawbacks of these heuristic approaches are the following: (*i*) they are not intrinsically statistical; (*ii*) they lead to estimators whose properties are unknown or largely unexplored; (*iii*) they do not take into account the results of any model fit. In this sense we have extended these traditional techniques to more robust procedures from a statistical perspective when it comes to consistently estimating the embedding parameters. Particularly, we have focused on a statistical approach which solves those three disadvantages.

The statistical approach to state space reconstruction can be viewed as a best subset selection problem within this context. The idea behind it is to select the embedding parameters that provide the best fit in the estimation of any quantitative measure e.g., the Lyapunov exponents taking into account some information criteria. As far this doctoral dissertation is concerned we have used the statistical approach based on model selection procedures taking into account the Bayesian information criterion instead of heuristic techniques and we have also considered leave one out cross-validation techniques. In any case, we really think that the information derived from the heuristic approaches might be still useful and should not be disregarded as a complementary information. Keep in mind that the embedding parameters belong to the parameter set to be estimated when selecting the best model by several estimation methods of the Lyapunov exponent.

As we have seen the traditional attractor reconstruction procedure assumes that the dynamic system is sampled with uniform-time frequency e.g., 1-day, 1-hour, 30-min, 1-min and so on. However, there has been increasing interest in situations where observations are characterised by its almost continuity. That is, the data does not come from a series of values measured periodically in time. For instance, in financial markets the emergence of new technologies have had important implications for both the quantity and quality of existing data in finance. Nowadays trades realized by traders are replaced by algorithmic trades executed in an automated way providing a huge amount of data over many intra-daily disaggregated time intervals.

The information generated by the interactions between traders who buys and sells financial instruments such as stocks, commodities, currencies, derivatives and so on is apparently encoded in frequencies which are not usually equally spaced in time e.g., tick-by-tick. Thus if we observe a particular financial asset i.e., a currency pair on the Foreign Exchange Market, the quotes or rates of this financial asset are sampling by tick-by-tick intervals which do not follow a constant rhythm. Each tick will appear when there is a change, upward or downward, in the trade price of each transaction. Can we use this kind of information to build a dynamic system model, and if so, how is it related to the true data-generating process?. In this sense, we have extended the reconstruction procedure from time-series data with non-uniform time-frequency proposing a novel algorithm based on those results.

Continuing in the context of the financial markets recently some papers have been published looking for nonlinearities and chaos using intra-daily financial time series. All of them found little evidence to support the presence of chaotic behaviours in their datasets but they found strong evidences of nonlinearities. In this case we have cleared up this chaos model-data paradox showing a possible answer to it, at least when studying financial time series. Chaotic systems are sensitive to initial conditions, so temporal dependence in chaotic time series is lost as the series are sampled at too long time intervals, appearing as independent, even though their underlying generator system could come from a chaotic system. In financial time series, which quotes are continuously traded on markets, the daily sampling may be too long.

We have used 14 currency pairs on the Foreign Exchange Market (FOREX) to test our suggested solution to the chaos model-data paradox. Our main hypothesis is that chaos is elusive in financial empirical studies because of loss of information that occurs when daily quotes are used. This could hinder the detection of chaos in those time series. We have proposed taking into account all the information available in the financial markets (full sample information on quotes) instead of daily data. If one wants to detect chaotic signals inside financial time series, we also recommend using lagged returns instead of classical daily returns with the purpose of preserving the dynamic dependence on information.

In this sense, we have been able to detect chaotic signals inside real-world financial time-series data, at least in the 14 currency pairs and days chosen. Nevertheless, we have not intended to generalise this finding to all financial series or even to all the FX rates available. The main interest has been to illustrate that by choosing a tick-by-tick frequency instead of a daily one, with the purpose of preserving the dynamic dependence on information, we could find chaos. At least, in the 14 tick-by-tick FX rates and the days considered. As far as we know these kind of FX tick-by-tick rates have never been tested for chaos. This fact could open up a new line of research in which new contributions may appear considering other financial assets or periods.

Moving on, once we have reconstructed the attractor we wanted to know if the unknown data-generating process presented a simple dynamic, behaved chaotically or came from a purely stochastic process. Methods and techniques related to test the hypothesis of chaos try to quantify the initial-value sensitive property estimating the Lyapunov exponents. Note that if one knows the data-generating process behind the time series the theoretical Lyapunov exponent can be calculated directly using its own definition as we have illustrated in this doctoral dissertation. However we have assumed that the true dynamics of the system is unknown because in most real-world observed time series the data-generating process is rarely known a priori.

There are two main methods in the literature that provide the estimated Lyapunov exponent from time-series data. We have provided an overview about the main theoretical and computational aspects of the traditional direct methods which directly measures the growth rate of divergence between two trajectories with an infinitesimal difference in their initial conditions. Then the underlying algorithms have been described in detail. We have illustrated empirically the main drawbacks of the direct estimators of the Lyapunov exponent. To sum up, the traditional direct methods do not provide us consistent estimators and robustness to the presence of (small) measurement errors; they do not have a satisfactory performance in detecting existing non-linearities on time-series data of moderate sample sizes; they do not allow the estimation of the full spectrum of Lyapunov exponents; the asymptotic distribution of the estimator does not exist which means that it does not allow the building of formal tests. For those reasons we have focused on the so-called *jacobian indirect methods* which solve those disadvantages.

Particularly, we have focus on the global neural net approach by approximating the unknown nonlinear system through a feed-forward single hidden layer neural network. Those neural net methods provide a well-fit of any unknown linear or nonlinear model. They also have the advantage to allow the analytical derivation, rather than numerically the jacobian needed for the estimation of the Lyapunov exponents. This process has been illustrated by examples.

We have also described the right procedure to obtain a consistent estimator of the Lyapunov exponent from the partial derivates computed previously by two different procedures and four ways of subsampling by blocks. At this point, we have provided a discussion about the optimal sample size used for estimating the partial derivatives of the jacobian and the optimal block length considered which is the number of evaluation points (number of products of the jacobian) used for estimating the Lyapunov exponents. Then a feasible test statistics were introduced and a one-sided test was proposed for the purpose of testing the hypothesis of chaos based on the theoretical asymptotic properties of the neural net estimator of the Lyapunov exponent and the consistent estimator of its variance.

Then we have focus on studying the robustness of the algorithms proposed for estimating the Lyapunov exponent. For this purpose we have compared the results provided by the traditional direct methods with those of the jacobian indirect methods. In this sense we can remark that the indirect methods provide better estimates than direct methods in all the experiments we have conducted. We have also shown empirically that our algorithms are robust to the presence of (small) measurement errors. Although as the noise increases the error committed increases but it is not proportional in any case.

Note that we have only focused on the largest Lyapunov exponent as direct methods do not estimate the full spectrum. In addition those methods do not allow us to make inference about it. In our case, we will be able to do so. Hence we have tested the reliability of the algorithms proposed. The results have reported a satisfactory performance in moderate sample sizes. This fact is really important for those researchers who work with short time-series data. The empirical size decreased and the empirical power increased as the sample size increased which means that our tests are consistent.

Later we have provided a discussion about the right procedure to obtain the partial derivatives based on the local jacobian approach. In this sense, we have highlighted the main differences with the global jacobian approach. We have also illustrated how to get a consistent estimator of the Lyapunov exponent based on the local jacobian approach. Note that the asymptotic properties of the Lyapunov exponent estimator are common to all ijacobian indirect methods irrespective of the choice of the approach used for estimating the function v in the global jacobian or the functions v_j in the local jacobian. This is due to the fact that the Lyapunov exponent, which does not depend on the estimation method, will be dominant in the limit distribution. Thus we have used these results to calculate the standard error of the Lyapunov exponent estimator and investigate the statistical significance of the sign of the exponents considering the local jacobian context.

Traditionally the main contributions proposed by the scientific community regarding the local jacobian indirect methods have focused on local linear approaches. The main drawbacks of these local linear estimators of the Lyapunov exponent were the following: (*i*) the estimated partial derivatives will be a constant for each embedding dimension considered. That is, it will have the same value regardless of the point at which it is valued since it will only depend on the values taken by the estimated coefficients; (*ii*) approximate linearly the functions υ_j locally at each point j through the result provided by linear weighted least squares regressions may not be the best approach since chaotic behaviour is a phenomenon in nonlinear dynamic systems. It does not exist in linear systems.

For these reasons we have extended those results by proposing two novel alternatives which solve the disadvantages of the local linear estimators. Particularly, we have proposed the following two local nonlinear approaches: (*i*) polynomial kernel method; (*ii*) neural net kernel method. We have also provided the suitable algorithms in order to computes the local jacobian from the best-fitted polynomial kernel model and neural net kernel model respectively. These procedures has been illustrated by several examples. Then we have focused on studying the robustness of the algorithms proposed for estimating the Lyapunov exponent extending it to the local jacobian indirect methods.

For this purpose we have compared the results provided by the traditional direct methods with those of the jacobian indirect methods based on three nonlinear approaches. In this sense we can remark that the indirect methods provide better estimates than direct methods in all the experiments we have conducted. In particular, the local jacobian indirect methods report better estimates than direct methods and global jacobian indirect methods. Both local nonlinear approaches behave similarly although local neural net kernel models are somewhat better than local polynomial kernel models.

Then we have shown empirically that our algorithms are robust to the presence of (small) measurement errors because the results obtained are comparable to those which are noise free. Although as the noise increases the error committed increases but it is not proportional in any case. The jacobian indirect methods seems to perform well for every noisy time-series data. The price we have to pay is a greater computational complexity from two points of view, the computing time and the tuning parameters. This fact has a greater effect on local jacobian indirect methods than might be expected.

Last but not least, we have provided a detailed review of the main algorithms publicly available at the Comprehensive R Archive Network (CRAN) related to nonlinear time series analysis and chaos. There are three R packages recently developed about this topic. These R packages are based on ideas inspired by the time series analysis (TISEAN) project. All those packages implement its algorithms based exclusively on the direct methods for estimating the Lyapunov exponent. In this sense we have presented a novel R package called **DChaos** which contains the algorithms that we have implemented in this doctoral dissertation. These algorithms are publicly available at `www.CRAN.R-project.org/package=DChaos`.

As far as we know the **DChaos** package is the first R library to provide the global and local jacobian indirect methods in this context. The package's functionality has been illustrated by examples. R users might make statistical inference about Lyapunov exponents significance and test consistently the hypothesis of chaos from time-series data by various data-based computational methods. To sum up, the **DChaos** package provides a R interface for researchers interested in the field of chaotic dynamic systems and nonlinear time series analysis and professors (and students) who teach (learn) courses related to those topics. Now let us highlight the main fruits obtained from this harvest collecting on paper everything discussed in this section.

7.2 Contributions proposed

This section provides a summary of the main contributions proposed, communications presented at national and international conferences and research stays developed during the elaboration of this doctoral dissertation. Before listing them (by chronological order) I would like to reiterate my deepest gratitude to all those people and institutions that have been involved in one way or another and have accompanied me throughout this fruitful journey.

§1. Julio E. Sandubete and Lorenzo Escot. Some topics about chaotic behaviour in high frequency economic time series. 10th International Conference on Chaotic Modeling and Simulation on May 30 - June 2, 2017.

§2. Julio E. Sandubete. Project: Nonlinear analysis of high-frequency time series: new regularities on very small time scales. Research stay at Real Colegio Complutense (Harvard University) under the supervision of Prof. José Manuel Martínez Sierra on March 7 - June 4, 2018.

§3. Julio E. Sandubete and Lorenzo Escot. A new approach about how to make reliable predictions inside chaotic regions. 11th International Conference on Chaotic Modeling and Simulation on June 5 - 8, 2018.

§4. Julio E. Sandubete and Lorenzo Escot. Generalising the BDS test to check i.i.d. in high-frequency financial time series. 11th International Conference on Nonlinear Mathematics and Physics on June 12 - 15, 2018.

§5. Julio E. Sandubete. Project: Nonlinear time series analysis and Chaos. Research stay at Dipartimento di Scienze Statistiche (Università di Bologna) under the supervision of my external suppervisor Prof. Simone Giannerini and the participation of Prof. Howell Tong on April 26 - June 4, 2019.

§6. Julio E. Sandubete and Lorenzo Escot. DChaos: An R Package for Detecting Chaotic Signals inside Time Series. 2nd Spanish Young Statisticians and Operational Researchers Meeting on June 5 - 8, 2019.

§7. Julio E. Sandubete and Lorenzo Escot. DChaos: Chaotic Time Series Analysis. R package version 0.1-5 publicly available at the Comprehensive R Archive Network (2020).

§8. Julio E. Sandubete and Lorenzo Escot. Chaotic signals inside some tick-by-tick financial time series. Chaos, Solitons & Fractal 137, 109852 (2020).

§9. Julio E. Sandubete and Lorenzo Escot. DChaos: R Package for Chaotic Time Series Analysis. R Journal (Review process).

§10. Julio E. Sandubete, Lorenzo Escot and Simone Giannerini. Quantifying chaos by various jacobian indirect methods (In preparation).

§11. Julio E. Sandubete, Lorenzo Escot and Simone Giannerini. A novel algorithm for estimating the Lyapunov exponent: Local Neural Nets (In preparation).

§12. Lukasz Pietrych, Julio E. Sandubete and Lorenzo Escot. Analysis of Nonlinearities and Chaos inside Bitcoin time series (In preparation).

§13. Lorenzo Escot, Julio E. Sandubete and Andrés Fernández-Díaz. A Lyapunov exponent approach to make reliable predictions (In preparation).

7.3 Future research questions

Let us conclude this doctoral dissertation making one last stop along the way and provide some ideas that could not be answered in this PhD thesis but which have been proposed as working papers in the near future. Particularly, we are still interested in exploring the following key research questions:

(*i*) We were starting this doctoral dissertation remarking that in deterministic dynamic systems is not possible to give a rigorous definition that account for all the aspects that the term chaos implies so far. So we have adopted an operational approach in order to characterise a chaotic dynamic system through the so-called Lyapunov exponents.

In this sense, we want to know if we could use this quantitative measure to simulate time-series data whose underlying generating system has a certain value of the Lyapunov exponent assigned a priori, and if so, how is it related to the true data-generating process?; how could we use Lyapunov exponents for estimating a plausible prediction horizon?; how we could use it recursively as a test of structural change?.

(*ii*) Remember that if we take into account the empirical environment we do not usually have the advantage of observing directly the state of the dynamic system let alone knowing the functional form that generates the dynamic associated with it. Instead of that, we have assumed that there will be an observer function that include an additive measurement error which generates observations. In this sense, we want to know the effect of adding a dynamic noise to an observer function instead of a measurement noise; does it interact with the data-generating process?, and if so, how it interacts?.

We would also be interested in knowing how a small dynamic noise could be amplified non-uniformly through the dynamics if the system has the initial-value sensitivity property (it is chaotic). This fact would be very interesting in order to make reliable predictions with chaotic signals even in a stochastic context, because we could see how the dynamic noise in certain areas of the attractor determines us more certainty (it is a true friend).

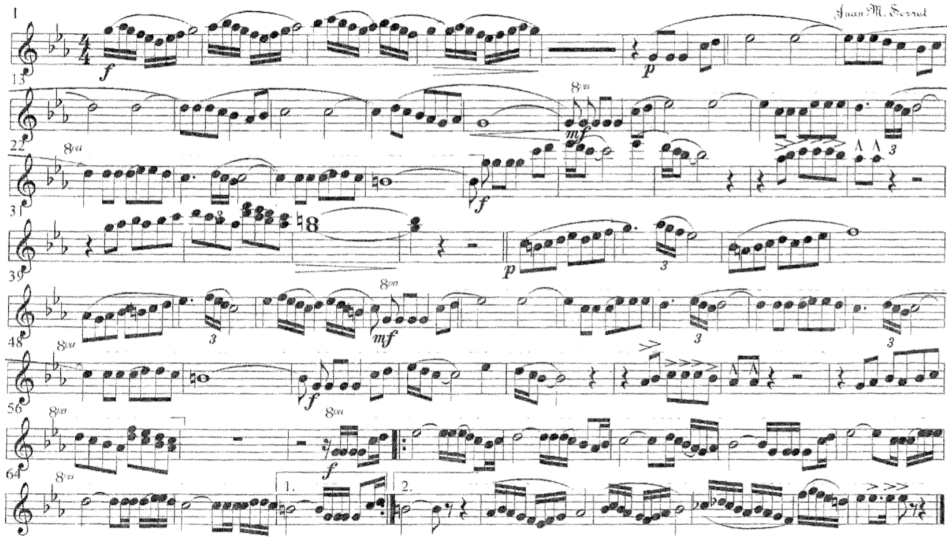
(*iii*) In this doctoral dissertation we have consider just one observable variable because we have assumed that all information available is a noise-contaminated sequence as a scalar strictly stationary univariate time serie. In this sense, we want to know how we could extend the results shown in this PhD thesis to a multivariate environment?; would it make sense to use the embedding procedure in this case?; how would we estimate the embedding parameters in this context?. We are also really interested in developing the algorithms proposed in this doctoral dissertation from multivariate time-series data. This approach could open up a fruitful research framework for nonlinear multivariate time series analysis and their real-world applications.

(*vi*) Note that the different contributions that compose the jacobian indirect methods differ in the algorithm used for the estimation of the jacobian. So far we have focused on the global neural net approach, the local polynomial kernel regressions and the local neural net kernel models. We are interested in considering (at least) two more nonlinear approaches which try to estimate the jacobian by convolutional neural nets models and by recurrent neural networks. Will we get a better fit?; how could we use ensemble methods based on multiple learning algorithms in this context?

Keep in mind that in our case the main reason for using these machine learning techniques and algorithms is not to look for the best predictive model but to estimate a model that captures the nonlinear time dependence well enough and, additionally, allows us to obtain in an analytical way (instead of numerical) the jacobians from the unknown data-generating process. The robust estimation of these jacobians will allow us to contrast consistently the hypothesis of chaos based on the Lyapunov exponent estimator.

Finally we are interested in knowing how to apply parallelisation and big data techniques on the design of the algorithms proposed in order to obtain computational efficiency when applying them on massive databases. We will keep extending the functionality of the **DChaos** package providing useful tools and robust algorithms in the context of computational data science.

Permítanme dar el penúltimo paso de este largo caminar al son que marcan las notas de la partitura que aquí comparto concluyendo así de manera armónica esta tesis doctoral que comenzó de un modo tan caótico.



Bibliography

Abarbanel, H. D. *Analysis of observed chaotic data*. Springer, 1996.

Aeyels, D. Generic observability of differentiable systems. *SIAM Journal on Control and Optimization*, vol. 19(5), pp. 595–603, 1981.

Akaike, H. Maximum likelihood identification of gaussian autoregressive moving average models. *Biometrika*, vol. 60(2), pp. 255–265, 1973.

Alligood, K. T., Sauer, T. D. and Yorke, J. A. *Chaos: An Introduction to Dynamical Systems*. Springer, 1996.

Anagnostidis, P. and Emmanouilides, C. J. Nonlinearity in high-frequency stock returns: Evidence from the Athens Stock Exchange. *Physica A*, vol. 421, pp. 473–487, 2015. ISSN 03784371.

Andrews, D. W. K. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica*, vol. 59(3), pp. 817–858, 1991.

Aslan, A. and Sensoy, A. Intraday efficiency-frequency nexus in the cryptocurrency markets. *Finance Research Letters*, 2019. ISSN 1544-6123.

Barnett, W. A., Gallant, A., Hinich, M. J., Jungeilges, J. A., Kaplan, D. T. and Jensen, M. J. A single-blind controlled competition among tests for nonlinearity and chaos. *J Econom*, vol. 82(1), pp. 157 – 192, 1997. ISSN 0304-4076.

Bartlett, M. Chance or chaos? *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 153(3), pp. 321–330, 1990.

Bask, M. Dimensions and Lyapunov exponents from exchange rate series. *Chaos Solitons Fractals*, vol. 7(12), pp. 2199–2214, 1996. ISSN 09600779.

Bask, M., Liu, T. and Widerberg, A. The stability of electricity prices: estimation and inference of the lyapunov exponents. *Physica A: Statistical Mechanics and its Applications*, vol. 376, pp. 565–572, 2007.

Bendixson, I. Sur les courbes définies par des équations différentielles. *Acta Mathematica*, vol. 24, pp. 1–88, 1901.

BENSAÏDA, A. Noisy chaos in intraday financial data: Evidence from the american index. *Appl Math Comput*, vol. 226, pp. 258 – 265, 2014. ISSN 0096-3003.

BENSAÏDA, A. and LITIMI, H. High level chaos in the exchange and index markets. *Chaos Solitons Fractals*, vol. 54, pp. 90 – 95, 2013. ISSN 0960-0779.

BERGÉ, P., POMEAU, Y. and VIDAL, C. *L'Ordre dans le Chaos*. Hermann, Paris, 1984.

BERLINER, L. M. Statistics, probability and chaos. *Statistical Science*, pp. 69–90, 1992.

BLACKBURN, J. A., GRØNBECH-JENSEN, N. and SMITH, H. Stochastic noise and chaotic transients. *Physical review letters*, vol. 74(6), pp. 908, 1995.

BRADLEY, E. and KANTZ, H. Nonlinear time-series analysis revisited. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25(9), pp. 097610, 2015.

BROCK, W., HOMMES, C., SCHUMACHER, J., HEY, C., HANZON, B. and PRAAGMAN, C. Models of complexity in economics and finance. *System Dynamics in Economic and Financial Models.*, pp. 3–41, 1997.

BROCK, W. A., HSIEH, D. A., LEBARON, B. D., BROCK, W. E. ET AL. *Nonlinear dynamics, chaos, and instability: statistical theory and economic evidence*. MIT press, 1991.

BROER, H. and TAKENS, F. Reconstruction and time series analysis. In *Dynamical Systems and Chaos*, pp. 205–242. Springer, 2011.

BROWN, R., BRYANT, P. and ABARBANEL, H. D. Computing the lyapunov spectrum of a dynamical system from an observed time series. *Physical Review A*, vol. 43(6), pp. 2787, 1991.

BROWN, R. and CHUA, L. O. Clarifying chaos: Examples and counterexamples. *International Journal of Bifurcation and Chaos*, vol. 6(02), pp. 219–249, 1996.

BRUNO, B., FAGGINI, M., PARZIALE, A. ET AL. Complexity modelling in economics: The state of the art. *Economic Thought*, vol. 5(2), pp. 29–43, 2016.

CAO, L., MEES, A. and JUDD, K. Dynamics from multivariate time series. *Physica D: Nonlinear Phenomena*, vol. 121(1-2), pp. 75–88, 1998.

CHAN, K.-S. *Exploration of a Nonlinear World: An Appreciation of Howell Tong's Contributions to Statistics*. World Scientific, 2009.

CHAN, K.-S. and TONG, H. *Chaos: a statistical perspective*. Springer-Verlag, 2001.

CHIU, S.-T. Some stabilized bandwidth selectors for nonparametric regression. *The Annals of Statistics*, vol. 19(3), pp. 1528–1546, 1991.

CLAESKENS, G. and HJORT, N. L. The focused information criterion. *Journal of the American Statistical Association*, vol. 98(464), pp. 900–916, 2003.

COX, D. and SMITH, W. The superposition of several strictly periodic sequences of events. *Biometrika*, vol. 40(1/2), pp. 1–11, 1953.

CRUTCHFIELD, J. P., FARMER, J. D. and HUBERMAN, B. A. Fluctuations and simple chaotic dynamics. *Physics Reports*, vol. 92(2), pp. 45–82, 1982.

D'ALESSIO, L., KAFRI, Y., POLKOVNIKOV, A. and RIGOL, M. From quantum chaos and eigenstate thermalization to statistical mechanics and thermodynamics. *Adv Phys*, vol. 65(3), pp. 239–362, 2016.

DECHERT, W. D. and GENÇAY, R. Lyapunov exponents as a nonparametric diagnostic for stability analysis. *J Appl Econ*, vol. 7(S1), pp. S41–S60, 1992.

DEVANEY, R. *An introduction to chaotic dynamical systems*. CRC Press, 2018.

DEVANEY, R. L. *An introduction to chaotic dynamical systems*. Addison-Wesley, 1989.

DEYLE, E. R. and SUGIHARA, G. Generalized theorems for nonlinear state space reconstruction. *PLoS One*, vol. 6(3), 2011.

DINARDO, J. and TOBIAS, J. L. Nonparametric density and regression estimation. *Journal of Economic Perspectives*, vol. 15(4), pp. 11–28, 2001.

DITTRICH, B., HAHN, P. A., KOSLOWSKI, T. A. and NELSON, M. I. Can chaos be observed in quantum gravity? *Phys Lett B*, vol. 769, pp. 554 – 560, 2017.

ECKMANN, J. P., KAMPHORST, S. O., RUELLE, D. and CILIBERTO, S. Liapunov exponents from time series. *Phys Rev A*, vol. 34, pp. 4971–4979, 1986.

ECKMANN, J. P. and RUELLE, D. Ergodic theory of chaos and strange attractors. *Rev Mod Phys*, vol. 57, pp. 617–656, 1985.

ELLNER, S., GALLANT, A., MCCAFFREY, D. and NYCHKA, D. Convergence rates and data requirements for jacobian-based estimates of lyapunov exponents from data. *Phys Lett A*, vol. 153(6), pp. 357 – 363, 1991.

ESCOT, L. *Dinámica económica caótica: una aplicación al estudio del ciclo y el crecimiento económico*. Tesis doctoral, Universidad Complutense de Madrid, 2000.

FAGGINI, M. Chaotic time series analysis in economics: Balance and perspectives. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 24(4), pp. 042101, 2014.

FAGGINI, M. and PARZIALE, A. More than 20 years of chaos in economics. *Mind & Society*, vol. 15(1), pp. 53–69, 2016.

FAN, J. Design-adaptive nonparametric regression. *Journal of the American statistical Association*, vol. 87(420), pp. 998–1004, 1992.

FAN, J. and GIJBELS, I. Variable bandwidth and local linear regression smoothers. *The Annals of Statistics*, pp. 2008–2036, 1992.

FAN, J. and GIJBELS, I. *Local polynomial modelling and its applications: monographs on statistics and applied probability*, vol. 66. CRC Press, 1996.

FAN, J. and YAO, Q. *Nonlinear time series: Nonparametric and parametric methods*. Springer-Verlag, 2003.

FAN, J. and YAO, Q. *Nonlinear time series: nonparametric and parametric methods*. Springer Science & Business Media, 2008.

FERNÁNDEZ-DÍAZ, A. *La economía de la complejidad: economía dinámica caótica*. McGraw-Hill, 1994.

FERNÁNDEZ-DÍAZ, A. *Chaos theory: Current and future research and applications*. McGraw-Hill, 2019.

FERNÁNDEZ-DIAZ, A., GRAU-CARLES, P. and MANGAS, L. E. Nonlinearities in the exchange rates returns and volatility. *Physica A: Statistical Mechanics and its Applications*, vol. 316(1-4), pp. 469–482, 2002.

FORD, J. Chaos: Solving the unsolvable, predicting the unpredictable! In *Chaotic dynamics and fractals*, pp. 1–52. Elsevier, 1986.

FRASER, A. M. and SWINNEY, H. L. Independent coordinates for strange attractors from mutual information. *Phys Rev A*, vol. 33, pp. 1134–1140, 1986.

FURQUIM, G., PESSIN, G., FAIÇAL, B. S., MENDIONDO, E. M. and UEYAMA, J. Improving the accuracy of a flood forecasting model by means of machine learning and chaos theory. *Neural computing and applications*, vol. 27(5), pp. 1129–1141, 2016.

GANDOLFO, G. *Economic Dynamics*. Springer-Verlag Berlin Heidelberg, 2009.

GARCIA, C. A. *nonlinearTseries: Nonlinear Time Series Analysis*, 2019. R package version 0.2.6.

GARCIA, S. P. and ALMEIDA, J. S. Multivariate phase space reconstruction by nearest neighbor embedding with different time delays. *Physical Review E*, vol. 72(2), pp. 027205, 2005.

GAUSS, K. F. Disquisitiones generales circa area superficies curvas. *Gott. gel. Anz*, vol. 177, pp. 1761–1768, 1827.

GENÇAY, R. and DECHERT, W. D. An algorithm for the n lyapunov exponents of an n-dimensional unknown dynamical system. *Physica D*, vol. 59(1), pp. 142–157, 1992.

GENÇAY, R. and DECHERT, W. D. The identification of spurious lyapunov exponents in jacobian algorithms. *Studies in Nonlinear Dynamics & Econometrics*, vol. 1(3), 1996.

GIANNERINI, S. *Sensitive dependence on initial conditions: chaos and stochastic processes*. PhD thesis, Università di Bologna, 2002.

GIANNERINI, S. and ROSA, R. New resampling method to assess the accuracy of the maximal lyapunov exponent estimation. *Physica D: Nonlinear Phenomena*, vol. 155(1-2), pp. 101–111, 2001.

GIANNERINI, S. and ROSA, R. Assessing chaos in time series: Statistical aspects and perspectives. *Studies in Nonlinear Dynamics & Econometrics*, vol. 8(2), 2004.

GIANNERINI, S., ROSA, R. and GONZALEZ, D. L. Testing chaotic dynamics in systems with two positive lyapunov exponents: a bootstrap solution. *International Journal of Bifurcation and Chaos*, vol. 17(01), pp. 169–182, 2007.

GLEICK, J. *Chaos. Making a new science*. New York: Viking Penguin Inc., 1987.

GRASSBERGER, P. and PROCACCIA, I. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, vol. 9(1-2), pp. 189–208, 1983.

GRAU-CARLES, P. *Economía dinámica caótica: una aplicación al mercado de capitales español*. Tesis doctoral, Universidad Complutense de Madrid, 1996.

GRAU-CARLES, P. Empirical evidence of long-range correlations in stock returns. *Physica A: Statistical Mechanics and its Applications*, vol. 287(3-4), pp. 396–404, 2000.

HANADA, M., SHIMADA, H. and TEZUKA, M. Universality in chaos: Lyapunov spectrum and random matrix theory. *Phys Rev E*, vol. 97(2), pp. 022224, 2018.

HEGGER, R., KANTZ, H. and SCHREIBER, T. Practical implementation of nonlinear time series methods: The tisean package. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 9(2), pp. 413–435, 1999.

HEIDENREICH, N.-B., SCHINDLER, A. and SPERLICH, S. Bandwidth selection for kernel density estimation: a review of fully automatic selectors. *Advances in Statistical Analysis*, vol. 97(4), pp. 403–433, 2013.

HÉNON, M. A two-dimensional mapping with a strange attractor. *Communications in Mathematical Physics*, vol. 50(1), pp. 69–77, 1976.

HERZEL, H., EBELING, W. and SCHULMEISTER, T. Nonuniform chaotic dynamics and effects of noise in biochemical systems. *Zeitschrift für Naturforschung A*, vol. 42(2), pp. 136–142, 1987.

HIRSCH, M. W., SMALE, S. and DEVANEY, R. L. *Differential equations, dynamical systems, and an introduction to chaos*. Academic press, 2012.

HORNIK, K., STINCHCOMBE, M. and WHITE, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, vol. 2(5), pp. 359 – 366, 1989.

HUFFAKER, R., BITTELLI, M. and ROSA, R. *Nonlinear time series analysis with R*. Oxford University Press, 2017.

HUKE, J. P. and BROOMHEAD, D. S. Embedding theorems for non-uniformly sampled dynamical systems. *Nonlinearity*, vol. 20(9), pp. 2205–2244, 2007.

ISPOLATOV, I., MADHOK, V., ALLENDE, S. and DOEBELI, M. Chaos in high-dimensional dissipative dynamical systems. *Scientific reports*, vol. 5(1), pp. 1–6, 2015.

JENSEN, J. Chaotic dynamical systems with a view towards statistics: a review. *Networks and Chaos-Statistical and Probabilistic Aspects*, pp. 201–250, 1993.

JONES, M. C., MARRON, J. S. and SHEATHER, S. J. A brief survey of bandwidth selection for density estimation. *Journal of the American statistical association*, vol. 91(433), pp. 401–407, 1996.

KANTZ, H. A robust method to estimate the maximal lyapunov exponent of a time series. *Physics Letters A*, vol. 185(1), pp. 77 – 87, 1994.

KANTZ, H., RADONS, G. and YANG, H. The problem of spurious lyapunov exponents in time series analysis and its solution by covariant lyapunov vectors. *Journal of Physics A: Mathematical and Theoretical*, vol. 46(25), pp. 254009, 2013.

KANTZ, H. and SCHREIBER, T. Determinism and predictability. *Nonlinear time series analysis*, pp. 42–57, 1997.

KANTZ, H. and SCHREIBER, T. *Nonlinear time series analysis*, vol. 7. Cambridge university press, 2004.

KATOK, A. Lyapunov exponents, entropy and periodic orbits for diffeomorphisms. *Publications Mathématiques de l'IHÉS*, vol. 51, pp. 137–173, 1980.

KENNEL, M. B., BROWN, R. and ABARBANEL, H. D. I. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys Rev A*, vol. 45, pp. 3403–3411, 1992.

KIFER, Y. General random perturbations of hyperbolic and expanding transformations. *Journal d'Analyse Mathématique*, vol. 47(1), pp. 111–150, 1986.

KUTTA, W. Beitrag zur naherungsweisen integration totaler differentialgleichungen. *Z. Math. Phys.*, vol. 46, pp. 435–453, 1901.

LAHMIRI, S. Investigating existence of chaos in short and long term dynamics of moroccan exchange rates. *Physica A*, vol. 465, pp. 655 – 661, 2017. ISSN 0378-4371.

LAHMIRI, S. and BEKIROS, S. Chaos, randomness and multi-fractality in bitcoin market. *Chaos, solitons & fractals*, vol. 106, pp. 28–34, 2018.

LASOTA, A. and MACKEY, M. C. *Chaos, fractals, and noise: stochastic aspects of dynamics*, vol. 97. Springer Science & Business Media, 2013.

LAYEK, G. *An introduction to dynamical systems and chaos*. Springer, 2015.

LEBARON, B. Chaos and Nonlinear Forecastability in Economics and Finance. *Philos Trans A Math Phys Eng Sci*, vol. 348(1688), pp. 397–404, 1994. ISSN 1364-503X, 1471-2962.

LEONE, V. and KWABI, F. High frequency trading, price discovery and market efficiency in the ftse100. *Economics Letters*, vol. 181, pp. 174 – 177, 2019. ISSN 0165-1765.

LI, T.-Y. and YORKE, J. A. Period three implies chaos. *American Mathematical Monthly*, vol. 82, pp. 985–992, 1975.

LO, M. and LEE, C.-F. A reexamination of the market efficiency hypothesis: Evidence from an electronic intra-day, inter-dealer fx market. *The Quarterly Review of Economics and Finance*, vol. 46(4), pp. 565 – 585, 2006. ISSN 1062-9769.

LOADER, C. *Local regression and likelihood*. Springer Science & Business Media, 2006.

LORENZ, E. N. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, vol. 20(2), pp. 130–141, 1963.

LORENZ, E. N. *The essence of chaos*. University of Washington press, 1995.

LU, Z., HUNT, B. R. and OTT, E. Attractor reconstruction by machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28(6), pp. 061104, 2018.

LU, Z.-Q. *Estimating lyapunov exponents in chaotic time series with locally weighted regression*. Tesis Doctoral, 1994.

LU, Z.-Q. and SMITH, R. L. Estimating local lyapunov exponents. *Fields Institute Communications*, vol. 11, pp. 135–151, 1997.

MAÑÉ, R. On the dimension of the compact invariant sets of certain nonlinear maps. In *Dynamical Systems and Turbulence, Lecture Notes in Mathematics*, pp. 230–242. Springer, 1981.

MAY, R. M. Simple mathematical models with very complicated dynamics. *Nature*, vol. 261(5560), pp. 459–467, 1976.

MCCAFFREY, D. F., ELLNER, S., GALLANT, A. R. and NYCHKA, D. W. Estimating the lyapunov exponent of a chaotic system with nonparametric regression. *J Am Stat Assoc*, vol. 87(419), pp. 682–695, 1992.

MEDIO, A. *Chaotic Dynamics. Theory and Applications to Economics*. Cambridge University Press, 1992.

NARZO, A. F. D. *tseriesChaos: Analysis of Nonlinear Time Series*, 2019. R package version 0.1-13.1.

NYCHKA, D., ELLNER, S., GALLANT, A. R. and MCCAFFREY, D. Finding chaos in noisy systems. *J R Stat Soc Series B Stat Methodol*, vol. 54(2), pp. 399–426, 1992.

OLMEDO, E. Is there chaos in the spanish labour market? *Chaos, Solitons & Fractals*, vol. 44(12), pp. 1045–1053, 2011.

OSELEDEC, V. A multiplicative ergodic theorem. ljapunov characteristic number for dynamical systems. *Transactions of the Moscow Mathematical Society*, vol. 19, pp. 197–231, 1968.

OTT, E. *Chaos in dynamical systems*. Cambridge university press, 2002.

PACKARD, N. H., CRUTCHFIELD, J. P., FARMER, J. D. and SHAW, R. S. Geometry from a time series. *Physical review letters*, vol. 45(9), pp. 712, 1980.

PARK, J. Y. and WHANG, Y.-J. Random walk or chaos: A formal test on the lyapunov exponent. *Journal of Econometrics*, vol. 169(1), pp. 61–74, 2012.

PARLITZ, U. Identification of true and spurious lyapunov exponents from time series. *International Journal of Bifurcation and Chaos*, vol. 2(01), pp. 155–165, 1992.

PATHAK, J., LU, Z., HUNT, B. R., GIRVAN, M. and OTT, E. Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27(12), pp. 121102, 2017.

POINCARÉ, H. Mémoire sur les courbes définies par une équation différentielle (i). *Journal de mathématiques pures et appliquées*, vol. 7, pp. 375–422, 1881.

POITRAS, G. The pre-history of econophysics and the history of economics: Boltzmann versus the marginalists. *Physica A: Statistical Mechanics and its Applications*, vol. 507, pp. 89–98, 2018.

PUU, T. *Attractors, bifurcations, & chaos: Nonlinear phenomena in economics*. Springer Science & Business Media, 2013.

ROBINSON, J. C. A topological delay embedding theorem for infinite-dimensional dynamical systems. *Nonlinearity*, vol. 18(5), pp. 2135, 2005.

ROBINSON, R. C. *An introduction to dynamical systems: continuous and discrete*, vol. 19. American Mathematical Soc., 2012.

ROSENSTEIN, M. T., COLLINS, J. J. and LUCA, C. J. D. A practical method for calculating largest lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, vol. 65(1), pp. 117 – 134, 1993.

RÖSSLER, O. E. An equation for continuous chaos. *Physics Letters A*, vol. 57(5), pp. 397–398, 1976.

RUELLE, D. *Chance and chaos*, vol. 11. Princeton University Press, 1993.

RUELLE, D. and TAKENS, F. On the nature of turbulence. *Communications in Mathematical Physics*, vol. 20(3), pp. 167–192, 1971.

RUNGE, C. Ueber die numerische auösung von dierentialgleichungen. *Mathematische Annalen*, vol. 46(2), 1895.

RUPPERT, D. and WAND, M. P. Multivariate locally weighted least squares regression. *The annals of statistics*, pp. 1346–1370, 1994.

SANDUBETE, J. E. and ESCOT, L. *DChaos: Chaotic Time Series Analysis*, 2020. R package version 0.1-5.

SANJUÁN, M. A. Artificial intelligence, chaos, prediction and understanding in science. *arXiv preprint arXiv:2003.01771*, 2020.

SANO, M. and SAWADA, Y. Measurement of the lyapunov spectrum from a chaotic time series. *Physical review letters*, vol. 55(10), pp. 1082, 1985.

SAUER, T., YORKE, J. A. and CASDAGLI, M. Embedology. *Journal of statistical Physics*, vol. 65(3-4), pp. 579–616, 1991.

SAUER, T. D., TEMPKIN, J. A. and YORKE, J. A. Spurious lyapunov exponents in attractor reconstruction. *Physical review letters*, vol. 81(20), pp. 4341, 1998.

SCHWARZ, G. Estimating the dimension of a model. *The annals of statistics*, vol. 6(2), pp. 461–464, 1978.

SERLETIS, A. and SHINTANI, M. Chaotic monetary dynamics with confidence. *Journal of Macroeconomics*, vol. 28(1), pp. 228–252, 2006.

SHEATHER, S. J. and JONES, M. C. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 53(3), pp. 683–690, 1991.

SHINTANI, M. and LINTON, O. Is there chaos in the world economy? a nonparametric test using consistent standard errors. *International Economic Review*, vol. 44(1), pp. 331–357, 2003.

SHINTANI, M. and LINTON, O. Nonparametric neural network estimation of lyapunov exponents and a direct test for chaos. *Journal of Econometrics*, vol. 120(1), pp. 1–33, 2004.

SHONE, R. *Economic Dynamics: Phase diagrams and their economic application*. Cambridge University Press, 2002.

SHUSTER, G. Determinirovannyi khaos (deterministic chaos). *Moscow: Mir*, 1988.

SKOKOS, C. H., GOTTWALD, G. A. and LASKAR, J. *Chaos detection and predictability*, vol. 1. Springer, 2016.

STARK, J. Delay embeddings for forced systems. i. deterministic forcing. *Journal of Nonlinear Science*, vol. 9(3), pp. 255–332, 1999.

STARK, J., BROOMHEAD, D. S., DAVIES, M. and HUKE, J. Delay embeddings for forced systems. ii. stochastic forcing. *Journal of Nonlinear Science*, vol. 13(6), pp. 519–577, 2003.

STEWART, J. *God in the Chaos*. Harvest House Pub, 1991.

STOKER, T. M. Smoothing bias in density derivative estimation. *Journal of the American Statistical Association*, vol. 88(423), pp. 855–863, 1993.

TAKENS, F. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Lecture Notes in Mathematics*, pp. 366–381. Springer, 1981.

TAKENS, F. Heteroclinic attractors: time averages and moduli of topological conjugacy. *Boletim da Sociedade Brasileira de Matemática*, vol. 25(1), pp. 107–120, 1994.

TAKENS, F. Reconstruction theory and nonlinear time series analysis. *Handbook of Dynamical Systems*, vol. 3, pp. 345–377, 2010.

TANG, L., LV, H., YANG, F. and YU, L. Complexity testing techniques for time series data: A comprehensive literature review. *Chaos Solitons & Fractals*, vol. 81, pp. 117–135, 2015.

TEMPKIN, J. A. and YORKE, J. A. Spurious lyapunov exponents computed from data. *SIAM Journal on Applied Dynamical Systems*, vol. 6(2), pp. 457–474, 2007.

TERÄSVIRTA, T., TJØSTHEIM, D., GRANGER, C. W. J. ET AL. *Modelling nonlinear economic time series*. Oxford University Press Oxford, 2010.

TONG, H. *Non-linear time series: a dynamical system approach*. Oxford University Press, 1990.

TONG, H. Some comments on a bridge between nonlinear dynamicists and statisticians. *Physica D: Nonlinear Phenomena*, vol. 58(1-4), pp. 299–303, 1992.

TONG, H. *Chaos and Forecasting-Proceedings of the Royal Society Discussion Meeting*. World Scientific Publishing Co. Pte. Ltd., 1995.

TONG, H. Some comments on nonlinear time series analysis. *American Mathematical Society*, pp. 17–27, 1997.

TONG, H. and LIM, K. S. Threshold autoregression, limit cycles and cyclical data (with discussion). *Journal of the Royal Statistical Society, Series B*, vol. 42, pp. 245–292, 1980.

TSAY, R. S. and CHEN, R. *Nonlinear time series analysis*, vol. 891. John Wiley & Sons, 2019.

VAMVAKARIS, M. D., PANTELOUS, A. A. and ZUEV, K. M. Time series analysis of sp 500 index: A horizontal visibility graph approach. *Physica A*, vol. 497, pp. 41 – 51, 2018. ISSN 0378-4371.

VLACHOS, I. and KUGIUMTZIS, D. State space reconstruction from multiple time series. In *Topics on Chaotic Systems*, pp. 378–387. World Scientific, 2009.

WHANG, Y.-J. and LINTON, O. The asymptotic distribution of nonparametric estimates of the lyapunov exponent for stochastic time series. *J Econom*, vol. 91(1), pp. 1 – 42, 1999.

WHITNEY, H. Differentiable manifolds. *Annals of Mathematics*, pp. 645–680, 1936.

WIGGINS, S. *Introduction to applied nonlinear dynamical systems and chaos*, vol. 2. Springer Science & Business Media, 2003.

WOLF, A., SWIFT, J. B., SWINNEY, H. L. and VASTANO, J. A. Determining lyapunov exponents from a time series. *Physica D*, vol. 16(3), pp. 285 – 317, 1985.

WUERTZ, D., SETZ, T. and CHALABI, Y. *fNonlinear: Rmetrics - Nonlinear and Chaotic Time Series Modelling*, 2017. R package version 3042.79.

YANG, H.-L., RADONS, G. and KANTZ, H. Covariant lyapunov vectors from reconstructed dynamics: The geometry behind true and spurious lyapunov exponents. *Physical review letters*, vol. 109(24), pp. 244101, 2012.

YAO, Q. and TONG, H. On prediction and chaos in stochastic systems. *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences*, vol. 348(1688), pp. 357–369, 1994a.

YAO, Q. and TONG, H. Quantifying the influence of initial values on non-linear prediction. *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 56(4), pp. 701–725, 1994b.

ZHANG, W.-B. *Discrete dynamical systems, bifurcations and chaos in economics*. Elsevier, 2006.

DChaos package structure and documentation

Type Package
Version 0.1-5
Date 2020-05-09
Title Chaotic Time Series Analysis
Author Julio E. Sandubete [aut, cre], Lorenzo Escot [aut]
Maintainer Julio E. Sandubete <jsandube@ucm.es>
Imports xts, zoo, outliers, nnet, pracma, sandwich
License GPL (>= 2)
Encoding UTF-8
LazyData true
RoxygenNote 7.0.2
NeedsCompilation no

R topics documented:

Function	Description
embedding	Provides the delayed-coordinate embedding vectors backwards
gauss.sim	Simulates time-series data from the Gauss map
henon.sim	Simulates time-series data from the Hénon system
jacobian.net	Computes the partial derivatives from the best-fitted neural net model
logistic.sim	Simulates time-series data from the Logistic map
lyapunov	Estimates the Lyapunov exponent through several methods
lyapunov.max	Estimates the largest Lyapunov exponent by the Norma-2 procedure
lyapunov.spec	Estimates the Lyapunov exponent spectrum by the QR decomposition
netfit	Fits any standard feed-forward neural net model from time-series data
rossler.sim	Simulates time-series data from the Rössler system
summary.lyapunov	Summary method for a lyapunov object
w0.net	Estimates the initial parameter vector of the neural net model

---

`embedding` *Provides the delayed-coordinate embedding vectors backwards*

Description

This function generates both the uniform and non-uniform embedding vectors backwards using the method of delays from univariate time-series data.

Usage

```
embedding(x, m = 2, lag = 1, timelapse = c("FIXED", "VARIABLE"))
```

Arguments

`x`	a `vector`, a time-series object `ts` or `xts`, a `data.frame`, a `data.table` or a `matrix` depending on the method selected in `timelapse`.
`m`	a non-negative integer denoting the embedding dimension (Default 2).
`lag`	a non-negative integer denoting the reconstruction delay (Default 1).
`timelapse`	a character denoting if the time-series data are sampled at uniform time-frequency e.g., 1-month, 1-day, 1-hour, 30-min, 5-min, 1-min and so on `FIXED` or non-uniform time-frequency which are not equally spaced in time `VARIABLE` (Default `FIXED`).

Value

The uniform or non-uniform delayed-coordinate embedding vectors backwards by columns from an univariate time-series data considering the parameter set selected by the user. If `FIXED` has been selected `data` must be a `vector` or a time-series object `ts` or `xts`. Otherwise `VARIABLE` has to be specified. In this case `data` must be a `data.frame`, a `data.table` or a `matrix` with two columns, the date and the univariate time series as a sequence of numerical values, in that order. The date can have the following three classes: `POSIXt`, `Date` or `Factor`. In the latter case the date should come in the following format `YMD H:M:OS3` considering milliseconds e.g., 20190407 00:00:03.347. If you don't consider milliseconds you must put .000 after the seconds.

Note

Note that a key point to create a suitable reconstruction of the state-space is to fix a criteria in order to estimate the embedding parameters. Researchers usually estimate them using heuristic approaches based on prescriptions proposed by e.g., H.D. Abarbanel (1996) or H. Kantz and T. Schreiber (2004). The main drawbacks of these heuristic approaches are the following: they are not intrinsically statistical; their results are not robust; they lead to estimators whose properties are unknown or largely unexplored; they do not take into account the results of any model fit. The alternative proposed by the statistical approach solves those disadvantages. The statistical approach to state-space reconstruction can be viewed as a best subset selection problem within the nonparametric regression context as argued K.-S. Chan and H. Tong (2001). The DChaos package allows the R users to choose between both methods. By default it uses the statistical approach based on model selection procedures instead of heuristic techniques, see `netfit` function.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Ruelle, D., Takens, F. 1971 On the nature of turbulence. Communications in Mathematical Physics 20(3):167-192.

Takens, F. 1981 Detecting strange attractors in turbulence. Springer Berlin Heidelberg.

Abarbanel, H.D. 1996 Analysis of observed chaotic data. Springer.

Cha, K.-S., Tong, H. 2001 Chaos: a statistical perspective. Springer-Verlag.

Kantz, H., Schreiber, T. 2004 Nonlinear time series analysis, volume 7. Cambridge university press.

Huke, J.P., Broomhead, D.S. 2007 Embedding theorems for non-uniformly sampled dynamical systems. Nonlinearity 20(9):205-244.

Examples

```
## set.seed(34)
## Simulates time-series data from the Logistic map with chaos
## ts        <- DChaos::logistic.sim(n=1000, a=4)
## show(head(ts, 5))

## Provides the uniform delayed-coordinate embedding vectors (Backward)
## data      <- DChaos::embedding(ts, m=5, lag=2, timelapse="FIXED")
## show(head(data, 5))

## Simulates tick-by-tick data (bid price) for Starbucks company
## ts        <- highfrequency::sbux
## show(head(ts, 5))

## Provides the non-uniform delayed-coordinate embedding vectors (Backward)
## data      <- DChaos::embedding(ts, m=3, lag=4, timelapse="VARIABLE")
## show(head(data, 5))
```

`gauss.sim` *Simulates time-series data from the Gauss map*

Description

This function simulates time-series data from the Gauss map considering the parameter set selected by the user. The initial condition is a random number between 0 and 1. Some initial conditions may lead to an unstable system that will tend to infinity.

Usage

```
gauss.sim(
  alpha = 6.2,
  beta = -0.5,
  s = 0,
  x0 = runif(1, 0, 1),
  n = 1000,
  n.start = 50
)
```

Arguments

alpha	a non-negative integer denoting the value of parameter alpha (Default 6.2).
beta	a non-negative integer denoting the value of parameter beta (Default -0.5).
s	a non-negative integer denoting the variance value of the error term. If $s = 0$ gives the standard deterministic map (Default 0).
x0	a non-negative integer denoting the initial condition (Default random number between 0 and 1).
n	a non-negative integer denoting the length (Default 1000).
n.start	a non-negative integer denoting the number of observations that will be discarded to ensure that the values are in the attractor (Default 50).

Value

A time-series data object generated from the Gauss map with or without an additive measurement noise term. This dataset could be useful for researchers interested in the field of chaotic dynamic systems and non-linear time series analysis and professors (and students) who teach (learn) courses related to those topics.

Note

This function provides also noisy time-series data from the deterministic gauss map adding an additive measurement noise term if $s > 0$. We have added to each time-series data a normal multinomial error term denoted by $\varepsilon_t \sim N(0, s)$ with different variance values (s). In this sense we have considered it appropriate to add a measurement noise term because most real-world observed time-series data are usually noise-contaminated signals, characterised by an erratic and persistent volatility in certain periods and there is almost always a source of noise linked to measurement errors in real-world datasets.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Hilborn, R.C. 2004 Chaos and nonlinear dynamics: an introduction for scientists and engineers. Oxford, Univ. Press, New York.

Examples

```
## set.seed(34)
## Simulates time-series data from the deterministic gauss map
## with a chaotic behaviour.
## ts <- gauss.sim(alpha=6.2, beta=-0.5, s=0, n=1000)
##
## Simulates time-series data from the deterministic gauss map
## with a non-chaotic behaviour.
## ts <- gauss.sim(alpha=4.9, beta=-0.58, s=0, n=1000)
```

henon.sim *Simulates time-series data from the Henon map*

Description

This function simulates time-series data from the Henon map considering the parameter set selected by the user. The initial condition is a random number between -0.5 and 0.5. Some initial conditions may lead to an unstable system that will tend to infinity.

Usage

```
henon.sim(
  a = 1.4,
  b = 0.3,
  s = 0,
  x0 = runif(1, -0.5, 0.5),
  y0 = runif(1, -0.5, 0.5),
  n = 1000,
  n.start = 50
)
```

Arguments

a	a non-negative integer denoting the value of parameter a (Default 1.4).
b	a non-negative integer denoting the value of parameter b (Default 0.3).
s	a non-negative integer denoting the variance value of the error term. If $s = 0$ gives the standard deterministic map (Default 0).
x0	a non-negative integer denoting the initial condition of x-coordinate (Default random number between -0.5 and 0.5).
y0	a non-negative integer denoting the initial condition of y-coordinate (Default random number between -0.5 and 0.5).
n	a non-negative integer denoting the length (Default 1000).
n.start	a non-negative integer denoting the number of observations that will be discarded to ensure that the values are in the attractor (Default 50).

Value

A time-series data object generated from the Henon map with or without an additive measurement noise term. This dataset could be useful for researchers interested in the field of chaotic dynamic systems and non-linear time series analysis and professors (and students) who teach (learn) courses related to those topics.

Note

This function provides also noisy time-series data from the deterministic henon map adding an additive measurement noise term if $s > 0$. We have added to each time-series data a normal multinomial error term denoted by $\varepsilon_t \sim N(0, s)$ with different variance values (s). In this sense we have considered it appropriate to add a measurement noise term because most real-world observed time-series data are usually noise-contaminated signals, characterised by an erratic and persistent volatility in certain periods and there is almost always a source of noise linked to measurement errors in real-world datasets.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Hénon, M. 1976 A two-dimensional mapping with a strange attractor. Communications in Mathematical Physics 50(1):69-77.

Examples

```
## set.seed(34)
## Simulates time-series data from the deterministic henon map
## with a chaotic behaviour.
ts <- henon.sim(a=1.4, b=0.3, s=0, n=1000)
##
## Simulates time-series data from the deterministic henon map
## with a non-chaotic behaviour.
ts <- henon.sim(a=1.2, b=0.1, s=0, n=1000)
```

`jacobian.net` *Computes the partial derivatives from the best-fitted neural net model*

Description

This function computes analytically the partial derivatives from the best-fitted neural net model.

Usage

```
jacobian.net(
  model,
  data,
  m = 1:4,
  lag = 1:1,
  timelapse = c("FIXED", "VARIABLE"),
  h = 2:10,
  w0maxit = 100,
  wtsmaxit = 1e+06,
  pre.white = TRUE,
  trace = 1,
  seed.t = TRUE,
  seed = 56666459
)
```

Arguments

`model`	a neural network model fitted using the `netfit` function.
`data`	a `vector`, a time-series object `ts` or `xts`, a `data.frame`, a `data.table` or a `matrix` depending on the method selected in `timelapse`.
`m`	a non-negative integer denoting a lower and upper bound for the embedding dimension (Default 1:4).

lag	a non-negative integer denoting a lower and upper bound for the the reconstruction delay (Default 1:1).
timelapse	a character denoting if the time-series data are sampled at uniform time-frequency e.g., 1-month, 1-day, 1-hour, 30-min, 5-min, 1-min and so on FIXED or non-uniform time-frequency which are not equally spaced in time VARIABLE (Default FIXED).
h	a non-negative integer denoting a lower and upper bound for the number of neurones (or nodes) in the single hidden layer (Default 2:10).
w0maxit	a non-negative integer denoting the maximum iterations to estimate the initial parameter vector of the neural net models (Default 100).
wtsmaxit	a non-negative integer denoting the maximum iterations to estimate the weights parameter vector of the neural net models (Default 1e6).
pre.white	a logical value denoting if the user wants to use as points to evaluate the partial derivatives the delayed vectors filtered by the neural net model chosen TRUE or not FALSE (Default TRUE).
trace	a binary value denoting if the user wants to print the output on the console 1 or not 0 (Default 1).
seed.t	a logical value denoting if the user wants to fix the seed TRUE or not FALSE (Default TRUE).
seed	a non-negative integer denoting the value of the seed selected if seed.t = TRUE (Default 56666459).

Value

This function returns several objects considering the parameter set selected by the user. Partial derivatives are calculated analytically from the best-fitted neural net model. It also contains some useful information about the best-fitted feed-forward single hidden layer neural net model saved, the best set of weights found, the fitted values, the residuals obtained or the best embedding parameters set chosen. This function allows the R user uses the data previously obtained from the best-fitted neural network estimated by the netfit function if model is not empty. Otherwise data has to be specified.

Note

The main reason for using neural network models is not to look for the best predictive model but to estimate a model that captures the non-linear time dependence well enough and, additionally, allows us to obtain in an analytical way (instead of numerical) the jacobian functional of the unknown underlying generator system. The estimation of this jacobian or partial derivatives will later allow us to contrast our hypothesis of chaos estimating the Lyapunov exponents.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Eckmann, J.P., Ruelle, D. 1985 Ergodic theory of chaos and strange attractors. Rev Mod Phys 57:617–656.

Gencay, R., Dechert, W.D. 1992 An algorithm for the n lyapunov exponents of an n-dimensional unknown dynamical system. Physica D 59(1):142–157.

Shintani, M., Linton, O. 2004 Nonparametric neural network estimation of Lyapunov exponents and a direct test for chaos. Journal of Econometrics 120(1):1-33.

Examples

```
## set.seed(34)
## Simulates time-series data from the Logistic map with chaos
## ts        <- DChaos::logistic.sim(n=1000, a=4)
## show(head(ts, 5))

## Computes analytically the partial derivatives from the best-fitted neural net model
## showed in the netfit example
## model    <- DChaos::netfit(ts, m=1:4, lag=1:3, timelapse="FIXED", h=2:10)
## jacobian <- DChaos::jacobian.net(model=model)
## summary(jacobian)

## Partial derivatives are calculated analytically without setting previously any neural net model
## jacobian <- DChaos::jacobian.net(data=ts, m=3:3, lag=1:1, timelapse="FIXED", h=2:10)
## summary(jacobian)
```

`logistic.sim` *Simulates time-series data from the Logistic map*

Description

This function simulates time-series data from the Logistic map considering the parameter set selected by the user. The initial condition is a random number between 0 and 1. Some initial conditions may lead to an unstable system that will tend to infinity.

Usage

```
logistic.sim(a = 4, s = 0, x0 = runif(1, 0, 1), n = 1000, n.start = 50)
```

Arguments

`a`	a non-negative integer denoting the value of parameter a (Default 4).
`s`	a non-negative integer denoting the variance value of the error term. If $s = 0$ gives the standard deterministic map (Default 0).
`x0`	a non-negative integer denoting the initial condition (Default random number between 0 and 1).
`n`	a non-negative integer denoting the length (Default 1000).
`n.start`	a non-negative integer denoting the number of observations that will be discarded to ensure that the values are in the attractor (Default 50).

Value

A time-series data object generated from the Logistic map with or without an additive measurement noise term. This dataset could be useful for researchers interested in the field of chaotic dynamic systems and non-linear time series analysis and professors (and students) who teach (learn) courses related to those topics.

Note

This function provides also noisy time-series data from the deterministic logistic map adding an additive measurement noise term if $s > 0$. We have added to each time-series data a normal multinomial error term denoted by $\varepsilon_t \sim N(0, s)$ with different variance values (s). In this sense we have considered it appropriate to add a measurement noise term because most real-world observed time-series data are usually noise-contaminated signals, characterised by an erratic and persistent volatility in certain periods and there is almost always a source of noise linked to measurement errors in real-world datasets.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

May, R.M. 1976 Simple mathematical models with very complicated dynamics. Nature (261):459-467.

Examples

```
## set.seed(34)
## Simulates time-series data from the deterministic logistic map
## with a chaotic behaviour.
## ts <- logistic.sim(a=4, s=0, n=1000)
##
## Simulates time-series data from the deterministic logistic map
## with a non-chaotic behaviour.
## ts <- logistic.sim(a=3.2, s=0, n=1000)
```

---

lyapunov *Estimates the Lyapunov exponent through several methods*

Description

This is an all-in-one function. It provides, at the same time, the delayed-coordinate embedding vector (`embedding`), estimates the best neural net model (`netfit`), calculates the partial derivatives directly from the chosen neural network model (`javcobian.net`). Finally, this function estimates both the largest Lyapunov exponent through the Norma-2 procedure (`lyapunov.max`) and the Lyapunov exponent spectrum through the QR decomposition procedure (`lyapunov.spec`) taking into account the full sample and three different methods of subsampling by blocks.

Usage

```
lyapunov(
  data,
  m = 1:4,
  lag = 1:1,
  timelapse = c("FIXED", "VARIABLE"),
  h = 2:10,
  w0maxit = 100,
  wtsmaxit = 1e+06,
  pre.white = TRUE,
```

```
  lyapmethod = c("SLE", "LLE", "ALL"),
  blocking = c("BOOT", "NOVER", "EQS", "FULL", "ALL"),
  B = 1000,
  trace = 1,
  seed.t = TRUE,
  seed = 56666459,
  doplot = TRUE
)
```

Arguments

data	a vector, a time-series object ts or xts, a data.frame, a data.table or a matrix depending on the method selected in timelapse.
m	a non-negative integer denoting a lower and upper bound for the embedding dimension (Default 1:4).
lag	a non-negative integer denoting a lower and upper bound for the the reconstruction delay (Default 1:1).
timelapse	a character denoting if the time-series data are sampled at uniform time-frequency e.g., 1-month, 1-day, 1-hour, 30-min, 5-min, 1-min and so on FIXED or non-uniform time-frequency which are not equally spaced in time VARIABLE (Default FIXED).
h	a non-negative integer denoting a lower and upper bound for the number of neurones (or nodes) in the single hidden layer (Default 2:10).
w0maxit	a non-negative integer denoting the maximum iterations to estimate the initial parameter vector of the neural net models (Default 100).
wtsmaxit	a non-negative integer denoting the maximum iterations to estimate the weights parameter vector of the neural net models (Default 1e6).
pre.white	a logical value denoting if the user wants to use as points to evaluate the partial derivatives the delayed vectors filtered by the neural net model chosen TRUE or not FALSE (Default TRUE).
lyapmethod	a character denoting the procedure chosen to estimate the Lyapunov exponent. If LLE has been selected the function will estimate only the largest Lyapunov exponent through the Norma-2 method. If SLE has been selected the function will estimate the Lyapunov exponent spectrum through the QR decomposition. Otherwise ALL has to be specified. In this case the function will estimate the Lyapunov exponent considering both procedures (Default SLE).
blocking	a character denoting the blocking method chosen for figuring out the Lyapunov exponent. Available options are FULL if the user considers the full sample, NOVER if the user considers the non-overlapping sample, EQS if the user considers the equally spaced sample, BOOT if the user considers the bootstrap sample or ALL if the user considers all of them (Default BOOT).
B	a non-negative integer denoting the number of bootstrap iterations (Default 1000).
trace	a binary value denoting if the user wants to print the output on the console 1 or not 0 (Default 1).
seed.t	a logical value denoting if the user wants to fix the seed TRUE or not FALSE (Default TRUE).
seed	a non-negative integer denoting the value of the seed selected if seed.t = TRUE (Default 56666459).

`doplot` a logical value denoting if the user wants to draw a plot `TRUE` or not `FALSE`. If it is `TRUE` the evolution of the Lyapunov exponent values are represented for the whole period considering the blocking method chosen by the user. It shows as many graphs as embedding dimensions have been considered (Default `TRUE`).

Value

This function returns several objects considering the parameter set selected by the user. The largest Lyapunov exponent (Norma-2 procedure) and the Lyapunov exponent spectrum (QR decomposition procedure) by each blocking method are estimated. It also contains some useful information about the estimated jacobian, the best-fitted feed-forward single hidden layer neural net model, the best set of weights found, the fitted values, the residuals obtained, the best embedding parameters set chosen, the sample size or the block length considered by each blocking method. This function provides the standard error, the z test value and the p-value for testing the null hypothesis $H0 : \lambda_k > 0 for k = 1, 2, 3, \ldots, m$. Reject the null hypothesis H_0 means lack of chaotic behaviour. That is, the data-generating process does not have a chaotic attractor because of it does not show the property of sensitivity to initial conditions.

Note

We have considered it appropriate to incorporate a function that unifies the whole process to make it easier and more intuitive for the R users. The DChaos package provides several ways to figure out robustly the neural net estimator of the k-th Lyapunov exponent. Particularly, there are 8 functions (one for each procedure and blocking method) which estimate the Lyapunov exponents consistently. Hence the DChaos package allows the R users to choose between two different procedures to obtain the neural net estimator of the k-th Lyapunov exponent and four ways of subsampling by blocks: full sample, non-overlapping sample, equally spaced sample and bootstrap sample. The blocking methods what they do is to split the time-series data into several blocks by estimating a Lyapunov exponent for each subsample. If the R users choose the non-overlapping sample (`blocking = "NOVER"`), the equally spaced sample (`blocking = "EQS"`) or the bootstrap sample (`blocking = "BOOT"`) the mean and median values of the Lyapunov exponent for each block or subsample are saved. By default we recommend using the median value as it is more robust to the presence of outliers. Notice that the parameter `B` will only be considered if the R users choose the bootstrap blocking method.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Ellner, S., Gallant, A., McCaffrey, D., Nychka, D. 1991 Convergence rates and data requirements for jacobian-based estimates of lyapunov exponents from data. Physics Letters A 153(6):357-363.

McCaffrey, D.F., Ellner, S., Gallant, A.R., Nychka, D.W. 1992 Estimating the lyapunov exponent of a chaotic system with nonparametric regression. Journal of the American Statistical Association 87(419):682-695.

Nychka, D., Ellner, S., Gallant, A.R., McCaffrey, D. 1992 Finding chaos in noisy systems. Journal of the Royal Statistical Society 54(2):399-426.

Whang, Y.J., Linton, O. 1999 The asymptotic distribution of nonparametric estimates of the lyapunov exponent for stochastic time series. Journal of Econometrics 91(1):1-42.

Shintani, M., Linton, O. 2004 Nonparametric neural network estimation of Lyapunov exponents and a direct test for chaos. Journal of Econometrics 120(1):1-33.

See Also

lyapunov.max, lyapunov.spec

Examples

```
## set.seed(34)
## Simulates time-series data from the Logistic map with chaos
## ts        <- DChaos::logistic.sim(n=1000, a=4)
## show(head(ts, 5))

## Provides the Lyapunov exponent spectrum by the QR decomposition procedure considering the
## bootstrap blocking method directly from the Logistic map with chaos simulated.
## exponent <- DChaos::lyapunov(ts, m=3:3, lag=1:1, timelapse="FIXED", h=2:10, w0maxit=100,
##                      wtsmaxit=1e6, pre.white=TRUE, lyapmethod="SLE", blocking="ALL",
##                      B=100, trace=1, seed.t=TRUE, seed=56666459, doplot=FALSE))
## summmary(exponent)
```

---

lyapunov.max *Estimates the largest Lyapunov exponent*

Description

This function estimates the largest Lyapunov exponent through the Norma-2 procedure based on the partial derivatives computed by the jacobian.net function.

Usage

```
lyapunov.max(
  data,
  blocking = c("BOOT", "NOVER", "EQS", "FULL"),
  B = 1000,
  doplot = TRUE
)
```

Arguments

data	should be a jacobian object containing the partial derivatives computed by the jacobian.net function.
blocking	a character denoting the blocking method chosen for figuring out the largest Lyapunov exponent through the Norma-2 procedure. Available options are FULL if the user considers the full sample, NOVER if the user considers the non-overlapping sample, EQS if the user considers the equally spaced sample or BOOT if the user considers the bootstrap sample (Default BOOT).
B	a non-negative integer denoting the number of bootstrap iterations (Default 1000).
doplot	a logical value denoting if the user wants to draw a plot TRUE or not FALSE. If it is TRUE the evolution of the Lyapunov exponent values are represented for the whole period considering the blocking method chosen by the user. It shows as many graphs as embedding dimensions have been considered (Default TRUE).

Value

This function returns several objects considering the parameter set selected by the user. The largest Lyapunov exponent considering the Norma-2 procedure by each blocking method are estimated. It also contains some useful information about the estimated jacobian, the best-fitted feed-forward single hidden layer neural net model, the best set of weights found, the fitted values, the residuals obtained, the best embedding parameters set chosen, the sample size or the block length considered by each blocking method. This function provides the standard error, the z test value and the p-value for testing the null hypothesis $H0 : \lambda_k > 0 for k = 1$ (largest). Reject the null hypothesis H_0 means lack of chaotic behaviour. That is, the data-generating process does not have a chaotic attractor because of it does not show the property of sensitivity to initial conditions.

Note

The DChaos package provides several ways to figure out robustly the neural net estimator of the k-th Lyapunov exponent. On the one hand if the R users have previously obtained the partial derivatives from the `jacobian.net` function they can apply directly the function `lyapunov.spec` which estimates the Lyapunov exponent spectrum taking into account the QR decomposition procedure. They can also use the function `lyapunov.max` which estimates only the largest Lyapunov exponent considering the Norma-2 procedure. Hence the DChaos package allows the R users to choose between two different procedures to obtain the neural net estimator of the k-th Lyapunov exponent and four ways of subsampling by blocks: full sample, non-overlapping sample, equally spaced sample and bootstrap sample. The blocking methods what they do is to split the time-series data into several blocks by estimating a Lyapunov exponent for each subsample. If the R users choose the non-overlapping sample (`blocking = "NOVER"`), the equally spaced sample (`blocking = "EQS"`) or the bootstrap sample (`blocking = "BOOT"`) the mean and median values of the Lyapunov exponent for each block or subsample are saved. By default we recommend using the median value as it is more robust to the presence of outliers. Notice that the parameter `B` will only be considered if the R users choose the bootstrap blocking method.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Ellner, S., Gallant, A., McCaffrey, D., Nychka, D. 1991 Convergence rates and data requirements for jacobian-based estimates of lyapunov exponents from data. Physics Letters A 153(6):357-363.

McCaffrey, D.F., Ellner, S., Gallant, A.R., Nychka, D.W. 1992 Estimating the lyapunov exponent of a chaotic system with nonparametric regression. Journal of the American Statistical Association 87(419):682-695.

Nychka, D., Ellner, S., Gallant, A.R., McCaffrey, D. 1992 Finding chaos in noisy systems. Journal of the Royal Statistical Society 54(2):399-426.

Whang, Y.J., Linton, O. 1999 The asymptotic distribution of nonparametric estimates of the lyapunov exponent for stochastic time series. Journal of Econometrics 91(1):1-42.

Shintani, M., Linton, O. 2004 Nonparametric neural network estimation of Lyapunov exponents and a direct test for chaos. Journal of Econometrics 120(1):1-33.

Examples

```
## set.seed(34)
## Simulates time-series data from the Logistic map with chaos
## ts        <- DChaos::logistic.sim(n=1000, a=4)
```

```
## show(head(ts, 5))

## Provides the largest Lyapunov exponent by the Norma-2 procedure considering the
## bootstrap blocking method from the best-fitted neural net model and the partial
## derivatives showed in the jacobian.net example.
## jacobian <- DChaos::jacobian.net(data=ts, m=3:3, lag=1:1, timelapse="FIXED", h=2:10)
## summary(jacobian)
## exponent <- DChaos::lyapunov.max(data=jacobian, blocking="BOOT", B=100, doplot=FALSE)
## summary(exponent)
```

`lyapunov.spec` *Estimates the Lyapunov exponent spectrum*

Description

This function estimates the Lyapunov exponent spectrum through the QR decomposition procedure based on the partial derivatives computed by the `jacobian.net` function.

Usage

```
lyapunov.spec(
  data,
  blocking = c("BOOT", "NOVER", "EQS", "FULL"),
  B = 1000,
  doplot = TRUE
)
```

Arguments

`data`	should be a `jacobian` object containing the partial derivatives computed by the `jacobian.net` function.
`blocking`	a character denoting the blocking method chosen for figuring out the Lyapunov exponent spectrum through the QR decomposition procedure. Available options are `FULL` if the user considers the full sample, `NOVER` if the user considers the non-overlapping sample, `EQS` if the user considers the equally spaced sample or `BOOT` if the user considers the bootstrap sample (Default `BOOT`).
`B`	a non-negative integer denoting the number of bootstrap iterations (Default 1000).
`doplot`	a logical value denoting if the user wants to draw a plot `TRUE` or not `FALSE`. If it is `TRUE` the evolution of the Lyapunov exponent values are represented for the whole period considering the blocking method chosen by the user. It shows as many graphs as embedding dimensions have been considered (Default `TRUE`).

Value

This function returns several objects considering the parameter set selected by the user. The Lyapunov exponent spectrum considering the QR decomposition procedure by each blocking method are estimated. It also contains some useful information about the estimated jacobian, the best-fitted feed-forward single hidden layer neural net model, the best set of weights found, the fitted values, the residuals obtained, the best embedding parameters set chosen, the sample size or the block length considered by each blocking method. This function provides the standard error, the z test value and the p-value for testing the null hypothesis $H0 : \lambda_k > 0 for k = 1, 2, 3, \ldots, m$

(full spectrum). Reject the null hypothesis H_0 means lack of chaotic behaviour. That is, the data-generating process does not have a chaotic attractor because of it does not show the property of sensitivity to initial conditions.

Note

The DChaos package provides several ways to figure out robustly the neural net estimator of the k-th Lyapunov exponent. On the one hand if the R users have previously obtained the partial derivatives from the `jacobian.net` function they can apply directly the function `lyapunov.spec` which estimates the Lyapunov exponent spectrum taking into account the QR decomposition procedure. They can also use the function `lyapunov.max` which estimates only the largest Lyapunov exponent considering the Norma-2 procedure. Hence the DChaos package allows the R users to choose between two different procedures to obtain the neural net estimator of the k-th Lyapunov exponent and four ways of subsampling by blocks: full sample, non-overlapping sample, equally spaced sample and bootstrap sample. The blocking methods what they do is to split the time-series data into several blocks by estimating a Lyapunov exponent for each subsample. If the R users choose the non-overlapping sample (`blocking = "NOVER"`), the equally spaced sample (`blocking = "EQS"`) or the bootstrap sample (`blocking = "BOOT"`) the mean and median values of the Lyapunov exponent for each block or subsample are saved. By default we recommend using the median value as it is more robust to the presence of outliers. Notice that the parameter `B` will only be considered if the R users choose the bootstrap blocking method.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Ellner, S., Gallant, A., McCaffrey, D., Nychka, D. 1991 Convergence rates and data requirements for jacobian-based estimates of lyapunov exponents from data. Physics Letters A 153(6):357-363.

McCaffrey, D.F., Ellner, S., Gallant, A.R., Nychka, D.W. 1992 Estimating the lyapunov exponent of a chaotic system with nonparametric regression. Journal of the American Statistical Association 87(419):682-695.

Nychka, D., Ellner, S., Gallant, A.R., McCaffrey, D. 1992 Finding chaos in noisy systems. Journal of the Royal Statistical Society 54(2):399-426.

Whang, Y.J., Linton, O. 1999 The asymptotic distribution of nonparametric estimates of the lyapunov exponent for stochastic time series. Journal of Econometrics 91(1):1-42.

Shintani, M., Linton, O. 2004 Nonparametric neural network estimation of Lyapunov exponents and a direct test for chaos. Journal of Econometrics 120(1):1-33.

Examples

```
## set.seed(34)
## Simulates time-series data from the Logistic map with chaos
## ts        <- DChaos::logistic.sim(n=1000, a=4)
## show(head(ts, 5))

## Provides the Lyapunov exponent spectrum by the QR decomposition procedure considering the
## bootstrap blocking method from the best-fitted neural net model and the partial
## derivatives showed in the jacobian.net example.
## jacobian <- DChaos::jacobian.net(data=ts, m=3:3, lag=1:1, timelapse="FIXED", h=2:10)
## summary(jacobian)
## exponent <- DChaos::lyapunov.spec(data=jacobian, blocking="BOOT", B=100, doplot=FALSE)
```

```
## summary(exponent)
```

`netfit` *Fits any standard feedforward neural net model from time-series data*

Description

This function fits any standard feedforward neural net model from time-series data.

Usage

```
netfit(
  serie,
  m = 1:4,
  lag = 1:1,
  timelapse = c("FIXED", "VARIABLE"),
  h = 2:10,
  w0maxit = 100,
  wtsmaxit = 1e+06,
  pre.white = TRUE,
  trace = 1,
  seed.t = TRUE,
  seed = 56666459
)
```

Arguments

`serie`	a `vector`, a time-series object `ts` or `xts`, a `data.frame`, a `data.table` or a `matrix` depending on the method selected in `timelapse`.
`m`	a non-negative integer denoting a lower and upper bound for the embedding dimension (Default 1:4).
`lag`	a non-negative integer denoting a lower and upper bound for the the reconstruction delay (Default 1:1).
`timelapse`	a character denoting if the time-series data are sampled at uniform time-frequency e.g., 1-month, 1-day, 1-hour, 30-min, 5-min, 1-min and so on `FIXED` or non-uniform time-frequency which are not equally spaced in time `VARIABLE` (Default `FIXED`).
`h`	a non-negative integer denoting a lower and upper bound for the number of neurones (or nodes) in the single hidden layer (Default 2:10).
`w0maxit`	a non-negative integer denoting the maximum iterations to estimate the initial parameter vector of the neural net models (Default 100).
`wtsmaxit`	a non-negative integer denoting the maximum iterations to estimate the weights parameter vector of the neural net models (Default 1e6).
`pre.white`	a logical value denoting if the user wants to use as points to evaluate the partial derivatives the delayed vectors filtered by the neural net model chosen `TRUE` or not `FALSE` (Default `TRUE`).
`trace`	a binary value denoting if the user wants to print the output on the console `1` or not `0` (Default 1).

`seed.t` a logical value denoting if the user wants to fix the seed `TRUE` or not `FALSE` (Default TRUE).

`seed` a non-negative integer denoting the value of the seed selected if `seed.t = TRUE` (Default 56666459).

Value

This function returns several objects considering the parameter set selected by the user. The best-fitted feed-forward single hidden layer neural net model is saved. It also contains some useful information about the best set of weights found, the fitted values, the residuals obtained or the best embedding parameters set chosen. The best 10 models are displayed on the console. The first column is the neural net number, the second column is the embedding dimension, the third column is the lag or reconstruction delay considered, the fourth column is the number of neurones (or nodes) in the single hidden layer and the fifth column is the Bayesian Information Criterion (BIC) value corresponding to that neural net. Notice that the neural net models are sorted from lowest to highest BIC values.

Note

The process of adjustment to a neural net model often suffers from being trapped in local optima and different initialization strategies should be taken into account. For this reason the function `w0.net` have been implemented. This function estimates previously the initial parameter vector of the neural net model being able to set the maximum number of iterations that the user wants to obtain setting `w0maxit`. In addition, by default the neural network estimation is initialized with a fixed seed denoted by `seed.t=TRUE` with a value equal to `seed=56666459`. The R user can let the seed be fixed either randomly by `seed.t=FALSE` or even fix other value of the seed to be able to replicate the results obtained.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Ripley, B.D. 1996 Pattern Recognition and Neural Networks. Cambridge.

Venables, W.N., Ripley, B.D. 2002 Modern Applied Statistics with S. Fourth edition. Springer.

Hornik, K., Stinchcombe, M., White, H. 1989 Multilayer feedforward networks are universal approximators. Neural Networks 2(5):359-366.

Examples

```
## set.seed(34)
## Simulates time-series data from the Logistic map with chaos
## ts        <- DChaos::logistic.sim(n=1000, a=4)
## show(head(ts, 5))

## Provides the best-fitted neural network models for certain parameter set
## model    <- DChaos::netfit(ts, m=1:4, lag=1:3, timelapse="FIXED", h=2:10)
## summary(model)
```

`rossler.sim` *Simulates time-series data from the Rossler system*

Description

This function simulates time-series data from the Rossler system considering the parameter set selected by the user. The initial condition is a random number from the normal distribution with mean equal to 0 and variance equal to 1. Some initial conditions may lead to an unstable system that will tend to infinity.

Usage

```
rossler.sim(
  a = 0.2,
  b = 0.2,
  c = 5.7,
  s = 0,
  x0 = rnorm(1),
  y0 = rnorm(1),
  z0 = rnorm(1),
  time = seq(0, 100, 0.01),
  n.start = 50
)
```

Arguments

`a`	a non-negative integer denoting the value of parameter `a` (Default 0.2).
`b`	a non-negative integer denoting the value of parameter `b` (Default 0.2).
`c`	a non-negative integer denoting the value of parameter `c` (Default 5.7).
`s`	a non-negative integer denoting the variance value of the error term. If $s = 0$ gives the standard deterministic map (Default 0).
`x0`	a non-negative integer denoting the initial condition of x-coordinate (Default random number from the normal distribution).
`y0`	a non-negative integer denoting the initial condition of y-coordinate (Default random number from the normal distribution).
`z0`	a non-negative integer denoting the initial condition of z-coordinate (Default random number from the normal distribution).
`time`	a numeric vector denoting the time-lapse and the time-step (Default time-lapse equal to 10000 with a time-step of 0.01 seconds)
`n.start`	a non-negative integer denoting the number of observations that will be discarded to ensure that the values are in the attractor (Default 50).

Value

A time-series data object generated from the Rossler system with or without an additive measurement noise term. This dataset could be useful for researchers interested in the field of chaotic dynamic systems and non-linear time series analysis and professors (and students) who teach (learn) courses related to those topics.

Note

This function provides also noisy time-series data from the deterministic rossler system adding an additive measurement noise term if $s > 0$. We have added to each time-series data a normal multinomial error term denoted by $\varepsilon_t \sim N(0, s)$ with different variance values (s). In this sense we have considered it appropriate to add a measurement noise term because most real-world observed time-series data are usually noise-contaminated signals, characterised by an erratic and persistent volatility in certain periods and there is almost always a source of noise linked to measurement errors in real-world datasets. It has been implemented the classical Runge–Kutta method (RK4) in order to generate time-series data from continuous-time dynamical system as the Rossler system.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Rössler, O. 1976 An equation for continuous chaos. Physics Letters A 57(5):397-398.

Examples

```
## set.seed(34)
## Simulates time-series data from the deterministic rossler system
## with a chaotic behaviour.
ts <- rossler.sim(a=0.2, b=0.2, c=5.7, s=0, time=seq(0,100,0.1))
##
## Simulates time-series data from the deterministic rossler system
## with a non-chaotic behaviour.
ts <- rossler.sim(a=0.1, b=0.1, c=7, s=0, time=seq(0,100,0.1))
```

---

summary.lyapunov *Summary method for a lyapunov object*

Description

summary method for class "lyapunov".

Usage

```
## S3 method for class 'lyapunov'
summary(object, ...)
```

Arguments

object	an object of class "lyapunov" provided by lyapunov.max, lyapunov.spec or lyapunov functions.
...	further arguments passed to or from other methods.

Value

This function summary.lyapunov computes and returns a list of summary statistics of the results given in a lyapunov object using the components (list elements) from its argument.

Author(s)

Julio E. Sandubete, Lorenzo Escot

Examples

```
## set.seed(34)
## Simulates time-series data from the Logistic map with chaos
## ts        <- DChaos::logistic.sim(n=1000, a=4)
## show(head(ts, 5))

## Summary method for a lyapunov object (only 1 method)
## jacobian <- DChaos::jacobian.net(data=ts, m=3:3, lag=1:1, timelapse="FIXED", h=2:10)
## exponent <- DChaos::lyapunov.spec(data=jacobian, blocking="BOOT", B=100, doplot=FALSE)
## summary(exponent)

## Summary method for a lyapunov object (> 1 method)
## exponent <- DChaos::lyapunov(ts, m=3:3, lag=1:1, timelapse="FIXED", h=2:10, w0maxit=100,
##                      wtsmaxit=1e6, pre.white=TRUE, lyapmethod="SLE", blocking="ALL",
##                      B=100, trace=1, seed.t=TRUE, seed=56666459, doplot=FALSE))
## summmary(exponent)
```

`w0.net` *Estimates the initial parameter vector of the neural net model*

Description

This function estimates the initial parameter vector of the neural net model.

Usage

```
w0.net(
  x,
  y,
  m = 2,
  h = 2,
  rangx = 1/max(abs(x)),
  w0maxit = 100,
  seed.t = TRUE,
  seed = 56666459
)
```

Arguments

`x`	a `matrix` or a `data.frame` denoting the explanatory variables.
`y`	a `vector`, a `matrix` or a `data.frame` denoting the response variable.
`m`	a non-negative integer denoting the embedding dimension (Default 2).
`h`	a non-negative integer denoting the number of neurones (or nodes) in the single hidden layer (Default 2).
`rangx`	a non-negative integer denoting the range of the explanatory variables (Default 1/max(abs(x)).

w0maxit	a non-negative integer denoting the maximum iterations to estimate the initial parameter vector of the neural net models (Default 100).
seed.t	a logical value denoting if the user wants to fix the seed TRUE or not FALSE (Default TRUE).
seed	a non-negative integer denoting the value of the seed selected if seed.t = TRUE (Default 56666459).

Value

The optimal initial parameter vector of the neural net model considering the argument set selected by the user.

Note

The process of adjustment to a neural network often suffers from being trapped in local optima and different initialization strategies should be taken into account. For this reason the function w0.net have been implemented. This function estimates previously the initial parameter vector of the neural net model being able to set the maximum number of iterations that the user wants to obtain setting w0maxit. In addition, by default the neural network estimation is initialized with a fixed seed denoted by seed.t=TRUE with a value equal to seed=56666459. The R user can let the seed be fixed either randomly by seed.t=FALSE or even fix other value of the seed to be able to replicate the results obtained.

Author(s)

Julio E. Sandubete, Lorenzo Escot

References

Ripley, B.D. 1996 Pattern Recognition and Neural Networks. Cambridge.

Venables, W.N., Ripley, B.D. 2002 Modern Applied Statistics with S. Fourth edition. Springer.

Hornik, K., Stinchcombe, M., White, H. 1989 Multilayer feedforward networks are universal approximators. Neural Networks 2(5):359-366.