Nonlinear Time Series Analysis

S. Borovkova

February 21, 2001

1

Dynamical systems and nonlinear models

1.1 Introduction

In various situations information about a certain environment is represented as a sequence of measurements made consecutively in time, i.e. a *time series*. The crucial difference between the analysis of time series, and situations usually considered in classical statistics, is that the measurements within a time series will be stochastically dependent, while in classical statistics the fundamental assumption is that the observations are independent.

One possible source of complex behaviour of time series is the presence of random factors, such as measurement errors, system noise, etc. The highest degree of randomness we can encounter would be a time series which represents a sequence of outcomes of independent random variables, with no further structure. This, however, is not of interest for time series analysis.

The opposite of pure randomness is considered by the theory of deterministic dynamical systems. Here the future evolution is uniquely determined by the initial state and the evolution law. The behaviour of a deterministic system is not necessarily simple. In fact, developments in the area of nonlinear dynamical systems show the existence of deterministic time series which display highly erratic behaviour and may look like a realisation of a random process. In this case one speaks of the so-called *chaotic dynamics*. We shall consider this in more detail later.

One of the main aims of time series analysis is to model the dependence of future observations on the previous ones. Beginning with Yule's invention in 1927 of linear autoregression for the analysis of sunspot data, linear models have dominated time series analysis for about half a century. In these models the dependence of future observations on past ones is linear. To model complex behaviour by such a simple system, the presence of external random perturbations must be assumed. In traditional linear models driven by noise, such as the AR and ARMA models, a future observation is taken to be a linear combination of a certain number of previous observations and random, mostly Gaussian, disturbances, the so-called it innovations. However, as we shall see, there are simple examples of time series, for instance those related to chaotic dynamical systems, for which linear models are inadequate. This creates new problems: how to recognise such time series, and which methods to use for their modelling and prediction. These problems motivated many researches in the field of nonlinear time series analysis during the last decade.

The question of separating nonlinear time series from linear ones is rather complex and cannot be answered just by visual inspection. As an example, consider the four time series segments in Fig. 1.1-4 (A-D). Only one of them was obtained from a linear model driven by noise and the other three have strongly nonlinear features. We suggest to the reader to guess which one is linear.

One can see that it is difficult, even for an experienced eye, to discriminate nonlinear time series from linear ones just by visual inspection. However, there are

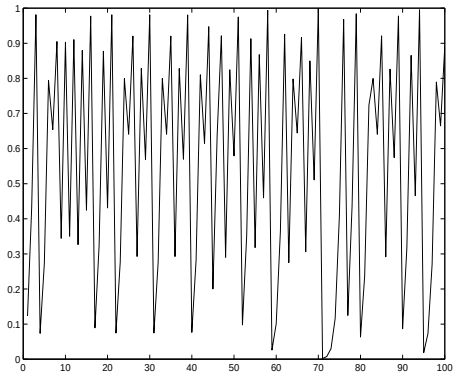


FIGURE 1.1. A

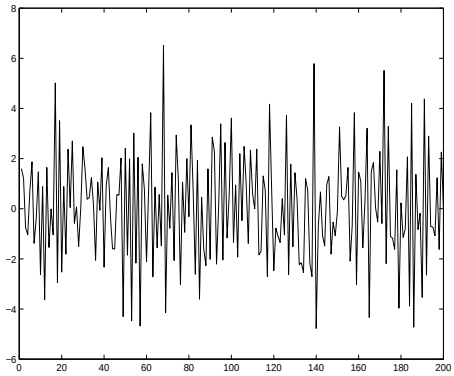


FIGURE 1.2. B

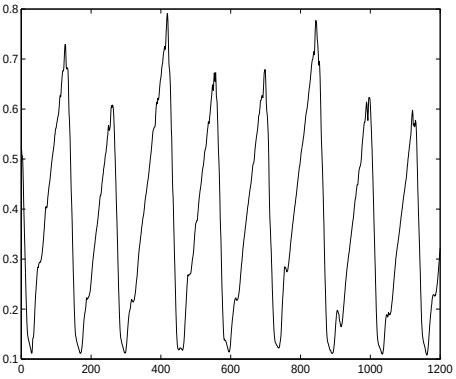


FIGURE 1.3. C

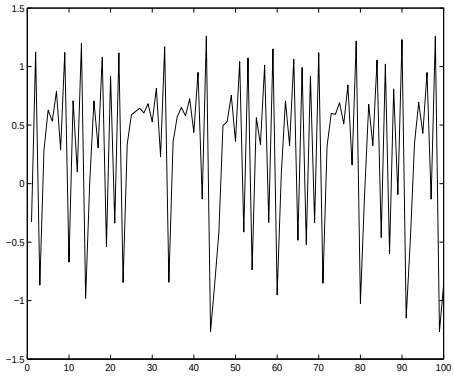


FIGURE 1.4. D

ways (graphical as well as analytical) to detect nonlinear time series and time series arising from deterministic systems. An important recent development in the theory of dynamical systems, the so-called *state space reconstruction*, assures that it is possible to reconstruct characteristic features of the underlying dynamics from the observed output, i.e. the actual time series. We shall consider this more in detail later.

Other crucial developments of the last decade, which also encouraged wide-spread interest in nonlinear methods, are the increased availability of powerful computers and the emergence of the field of machine learning, such as neural networks. This shifted modelling more towards data-driven methods and allowed for the exploration of a larger set of potential models, including nonlinear ones.

1.2 Nonlinear models

Let a time series $\{y_n\}_{n\in\mathbf{N}}$ be obtained by the rule

$$y_{n+1} = f(y_n), \tag{1.1}$$

where $f : \mathbf{R} \longrightarrow \mathbf{R}$ is a nonlinear function and $y_0 \in \mathbf{R}$ is an initial value. Even in this simple deterministic one-dimensional case the nonlinear function f can cause

complex behaviour of the resulting time series such that it appears random. Perhaps the best known example is the *logistic map* $f : [0,1] \longrightarrow [0,1]$ given by $f(x) = \lambda x(1-x)$, which is known to exhibit chaotic behaviour for certain parameter values, e.g. $\lambda = 4$. The time series A in Fig. 1.1 was generated by the logistic map

$$y_{n+1} = 4y_n(1 - y_n). \tag{1.2}$$

Deterministic models of the type (1.1) can be generalised to multidimensional ones of the form

$$X_{n+1} = F(X_n), \tag{1.3}$$

where $X_n \in \mathbf{R}^k$, $n \geq 0$ and X_0 is an initial value. The function $F : \mathbf{R}^k \longrightarrow \mathbf{R}^k$ is vector-valued. The difference equation (1.3) describes a discrete dynamical system in $\mathbf{R}^k$, with the evolution map F.

In reality, observations rarely evolve according to the model (1.3): usually there is some observational or measurement noise, as well as other random disturbances may be present in the system. Moreover, a deterministic model will inevitably be inadequate for modelling of real data. Therefore, it is more realistic to replace (1.3) by a model

$$X_{n+1} = F(X_n, e_{n+1}), \tag{1.4}$$

where $F : \mathbf{R}^{2k} \longrightarrow \mathbf{R}^k$, and $\{e_n\}_{n\in\mathbf{N}}$ is a sequence of k-dimensional random vectors, such that e_n is independent of $\{X_i\}_{i<n}$ (which, too, is quite an assumption). We will call (1.4) a (discrete) *stochastic dynamical system*. The sequence of random vectors $\{e_n\}$ is called the *dynamical noise* (also the *system*, or the *intrinsic noise*). For convenience of analysis we shall further assume that the dynamic noise is *additive*, so that (1.4) reduces to the model with additive noise

$$X_{n+1} = F(X_n) + e_{n+1}, \tag{1.5}$$

here $F : \mathbf{R}^k \longrightarrow \mathbf{R}^k$. If we have the following vector representations:

$$\begin{aligned} X_n &= (y_n, y_{n-1}, ..., y_{n-k+1}), \\ F(X_n) &= (f(X_n), y_n, ..., y_{n-k+2}), \text{ and} \\ e_n &= (\epsilon_n, 0, ..., 0), \end{aligned} \tag{1.6}$$

then (1.5) together with (1.6) results into

$$y_{n+1} = f(y_n, y_{n-1}, ..., y_{n-k+1}) + \epsilon_{n+1}, \tag{1.7}$$

and the function f is nonlinear if F is. Relation (1.7) defines a *nonlinear autoregression model of order k* for the time series $\{y_n\}$. Conversely, the model (1.7) for the time series $\{y_n\}$ can be written as a stochastic dynamical system (1.5) by vectorising $\{y_n\}$.

Using the example of the time series above we introduce a common exploratory tool for detecting the nonlinearity of a time series, the so-called *delay-1 map*: $y_n \longrightarrow y_{n+1}$. The delay-1 map will also be called return map. In practice one plots y_{n+1} vs. y_n. For the time series A, due to the relation (1.2) between the present and the next value, the points of the return map (see Fig.1.5) fall on the parabola, revealing the deterministic structure of this time series.

In most real-life situations the observational noise or other disturbances cause points to spread out, but in many cases the character of the underlying dynamics

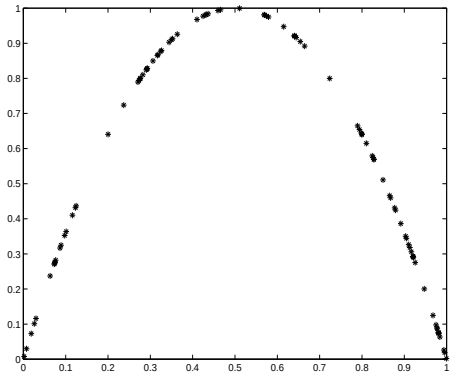


FIGURE 1.5. A

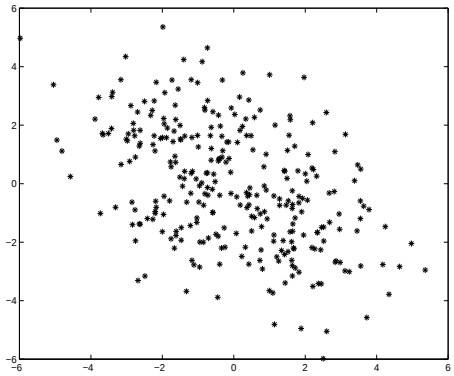


FIGURE 1.6. B

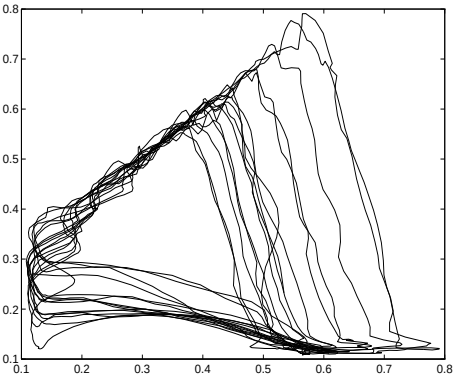


FIGURE 1.7. C

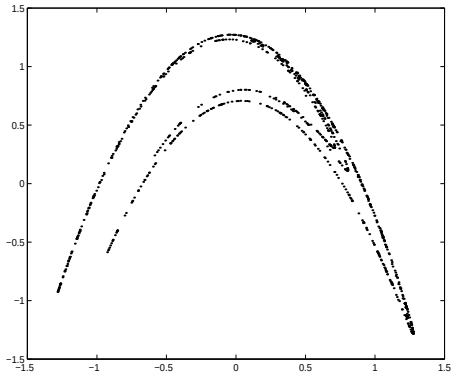


FIGURE 1.8. D

and nonlinearities can still be partially deduced from the structure of the return map. The return maps of the other three time series are shown in Fig. 1.6-8. The return map of the time series D (Fig. 1.8) immediately reveals its nonlinear structure. This is a time series coming from the so-called Henon dynamical system (we shall describe it later in this section).

Since the time series C is sampled with a very small time delay (i.e. the measurements are made with a very short time interval), we plotted not the delay-1, but delay-35 map, i.e. not y_{n+1} vs. y_n, but y_{n+35} vs. y_n (35 was taken because it is approximately a quarter of the "pseudocycle" of this time series). The fact that the points on Fig. 1.7 fall along a clearly visible though erratic curve indicates the nonlinear structure of this time series. This time series comes from a laboratory experiment of pressure measurement in a fluidized bed, which is believed to be a good example of a low-dimensional chaotic dynamical system with a low level of noise.

In contrast to the other three maps, the return map in Fig. 1.6 does not reveal any familiar structure, with points spreading over the whole graph. This time series is a realisation of a linear autoregression of order 1, driven by Gaussian noise.

The use of return maps for recognising nonlinear time series is closely related to the technique of the *state space reconstruction*. This technique is based on a remarkable result in the area of chaotic time series, the *Takens reconstruction theorem* [28]. We shall formulate it later, and here we try to give a flavour of the reconstruction

technique.

When studying return maps for a time series, we are interested in the behaviour of pairs of observations (y_{n-1}, y_n). In case of a nonlinear time series this can provide a better understanding of the dynamics that generated this time series. Intuitively, one may expect the behaviour of the vectors (y_{n-1}, y_n) to provide more insight into the underlying dynamics than the behaviour of the scalar observations y_n. Consequently, for more complicated cases more information about the dynamics can be gained by studying three-dimensional return maps, i.e. vectors (y_{n-2}, y_{n-1}, y_n), or even k-dimensional vectors $(y_{n-k+1}, ..., y_n)$ for some higher value of k. This is the basic idea of reconstruction: one replaces a time series $\{y_n\}$ by a sequence of vectors $(y_{n-k+1}, ..., y_n)$, which are called *reconstruction vectors*. The reconstruction vectors "live" in the space $\mathbf{R}^k$, also called the *embedding space*, and its dimension k is the *embedding dimension*. Note that both the space $\mathbf{R}^k$, where the stochastic dynamical system (1.5) evolves, and its dimension are naturally related to the embedding space and the embedding dimension.

We illustrate this technique using the example of the Henon dynamical system. This system evolves in phase space $\mathbf{R}^2$ according to the law:

$$(x, y) \longrightarrow (1 - ax^2 + y,\ bx). \tag{1.8}$$

For the values $a = 1.4,\ b = 0.3$ it is known to exhibit chaotic behaviour. In Fig. 1.9 a trajectory of this system in the original coordinates (x, y) is shown. The corresponding sequence of first coordinates $\{x_n\}$, is shown in Fig. 1.4. In Fig. 1.8 we visualised the reconstruction vectors generated by this time series. The two-dimensional embedding space is in this case given by the delay coordinates (x_n, x_{n+1}). Comparing Fig 1.8 with Fig. 1.9, we see a very similar structure in the embedding space and in the original phase space.

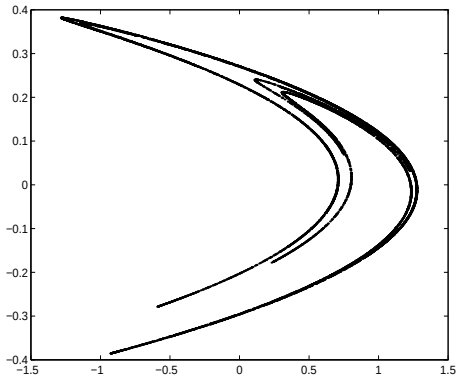


FIGURE 1.9. A trajectory of the Henon dynamical system

In contrast to this example, in most real-life situations neither an original phase space nor a dynamical law are known. What is actually observed is a one-dimensional time series of measurements made on some physical system. If one tries to study the underlying system on the basis of the observed time series, a fundamental problem arises: the physical system and the observed data live in different spaces. The original system evolves in a multidimensional phase space, its the coordinates are given

by the fundamental dynamical variables that describe the evolution. In practice it is impossible to measure and record all variables that define such a multidimensional phase space. One has at most measurements of one or a few dynamical variables, or possibly of a function of them, i.e. some derived quantity. For example, in a fluidized bed the mechanism of formation and movement of bubbles needs to be described by various dynamical variables, such as position, size and velocity of all bubbles, etc. But we only observe the series of pressure measurements at some fixed location in a fluidized bed. It is presumed that the pressure changes are caused by a dynamical process, but the local pressure itself is hardly a fundamental variable.

Most real-life systems by nature are processes in continuous time. In those cases we can consider our time series $(y_i)_{i\in N}$ as a realization of a continuous time process $(y_t)_{t\in R}$ at discrete time points. Suppose we want to reconstruct the underlying dynamics from observations on such a process. When trying to visualize the reconstructed time series obtained from (y_t), it is natural to start with two-dimensional reconstructions, i.e. those with elements $(y_t, y_{t+\delta})$, since they can be easily viewed on a plot. The question here is: how to choose the *delay time* δ? In the example of fluidized bed time series we chose $\delta = 35$. There is an extensive literature on this subject, the general guidelines are: one should not take too small or too large values of δ (why?) but seek intermediate values for which reconstruction plot exhibits some nontrivial structure.

The reconstruction theorem assures that a time series of only one scalar variable contains enough "information" to reconstruct the dynamics in the original multivariate phase space, without any prior knowledge of it. In particular, if the embedding dimension k is sufficiently high, then the trajectory generated by the series of reconstruction vectors in the embedding space mimics the evolution of the system in the phase space. Here the question of how big k should be to allow such reconstruction, is essential. If the original state space is d-dimensional, then the smallest value of the embedding dimension k which assures successful reconstruction is given by $k = 2d + 1$.

This criterion can be clarified by the following example. If we have a one-dimensional object, say a loop, then to visualise it properly we may need to live in dimension higher than one. If the loop twists into the figure 8 but is not self-intersecting, then to visualise such object a projection onto a two-dimensional space might not suffice. In general, to view a one-dimensional object unambiguously we need to live in $\mathbf{R}^3$. By unambiguity we mean roughly that there is a one-to-one map from that object to $\mathbf{R}^k$ which preserves differential information.

The reconstruction technique assures that essential features of the original dynamics can be deduced from the behaviour of the reconstruction vectors. In turn, this opens the way for classification of time series according to characteristics of the underlying dynamics. Two quantities, which are often used for classification, are the so-called *correlation integral* and the *correlation dimension.*

The correlation integral was originally defined by Grassberger and Proccacia [10] for the classification of time series arising from deterministic dynamical systems. They defined it via the *sample correlation integral*, which is the fraction of pairs of reconstruction vectors within distance r, in a finite segment of length N of the time series:

$$C_N^{(k)}(r) = \frac{2}{N(N-1)} \#\{(i,j) : \ 1 \le i < j \le N, \ \| X_i - X_j \|_{\max} \le r\}.$$

They then defined the correlation integral as the limit

$$C^{(k)}(r) = \lim_{N \longrightarrow \infty} C_N^{(k)}(r).$$

Of course in practice one does not have an infinite time series, but only a finite segment. The correlation integral is then estimated by its sample analogue $C_N^{(k)}(r)$. This quantity offers one possible way of classification: we estimate the correlation integrals and then compare the obtained estimates to distinguish different types of time series. Initially, however, correlation integrals were used to distinguish deterministic time series from stochastic ones (by a stochastic time series we mean here those driven by noise). In this context another quantity, the correlation dimension, is even more important.

If the correlation integral behaves as $C^{(k)}(r) \sim \text{const} \cdot r^{\alpha^{(k)}}$ for r in some neighbourhood of 0, then the exponent $\alpha^{(k)}$ is called the *correlation dimension* of the sequence of k-dimensional reconstruction vectors. The limit of this sequence

$$\alpha^{(\infty)} = \lim_{k \longrightarrow \infty} \alpha^{(k)}$$

is sometimes also called the correlation dimension of the time series, again, provided such limit exists(see, for instance, [21]). The correlation dimension $\alpha^{(k)}$ can be estimated by various methods, often using the sample correlation integrals.

The correlation dimension is defined above in terms of a time series, or, more precisely, the sequence of reconstruction vectors generated by it. It turns out that if the time series arises from a deterministic dynamical system, then, by virtue of the reconstruction theorem, the correlation dimension of this time series can give an idea about the complexity of the underlying dynamics, i.e. the number of active degrees of freedom. In fact, it coincides with the corresponding characteristic of the underlying dynamical system, namely the correlation dimension of the subset of the original phase space, where trajectories are concentrated (it can be, in principle, of lower dimension than the phase space). Moreover, its numerical value is close to other kinds of dimensions, introduced in the literature.

The basis for discrimination between time series is the study of the behaviour of the correlation dimension $\alpha^{(k)}$ as a function of k. For a time series induced by a deterministic dynamical system, when k is sufficiently large, the reconstruction vectors will concentrate on a subset of $\mathbf{R}^k$ of lower dimension, since the trajectories of the reconstruction vectors replicate those in the state space where the underlying dynamical system evolves. On the other hand, for a stochastic time series, as k grows in some range, the reconstruction vectors will always live on the whole space $\mathbf{R}^k$ (or a k-dimensional subset of it). Thus, one expects that for deterministic time series the estimates for correlation dimensions at some point remain constant even when further increasing k, while for stochastic time series the estimates increase together with the embedding dimension k.

These rather intuitive arguments have been widely used for distinguishing deterministic time series and still remain a useful tools for their classification. However, there are situations when this method can fail. According to a result of Osborne and Provenzale [21], there are cases of stochastically generated time series for which the sequence of correlation dimension estimates $\{\hat{\alpha}^{(k)}\}_{k=1,2,\ldots}$ also converges to a finite limit. This can happen, for example, when there is too much dependence in a time series. Another example, particularly relevant to financial time series, was

studied by C. Diks [9]. He discovered that ARCH-time series also exhibit lower dimensionality, while being stochastic.

To treat such cases a modification of the method above was suggested: first generate a linear time series driven by Gaussian noise, which has the same autocorrelations as the time series of interest, and then compare the behaviour of the correlation dimension estimates for these two series. A clear difference is an indication of determinism in the original time series. This is also the idea underlying the so-called *BDS-test*, which tests the hypothesis of time series admitting linear representation. Here also first a linear time series with the same autocovariance structure is generated, and then the behaviour of the correlation integrals is compared for both the original and generated series for different values of embedding dimensions. We shall return to this test in one of the following sections.

For a numerical illustration of the phenomenon described above, let us again consider the time series D induced by the Henon dynamical system and the linear time series B, which is an example of a stochastic time series. We estimated the correlation dimensions in both cases for increasing values of the embedding dimension k. Plots of the obtained estimates vs. k are shown in Fig. 1.10. Note that, for values of k larger than 2 the correlation dimension estimates for the time series D (Henon) stabilise around the theoretical value $\alpha = 1.24$ (see, for instance, [31]). For the linear time series B the estimates keep on growing together with k.

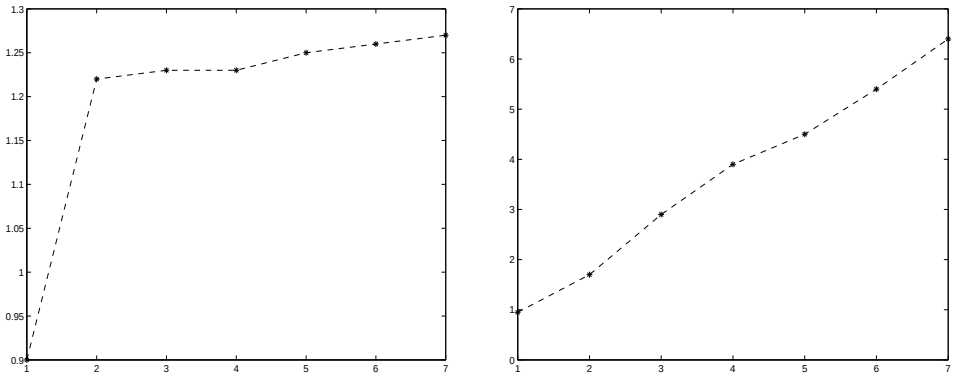


FIGURE 1.10. Estimated correlation dimension vs. embedding dimension: series D and B

This is just one of many methods of discrimination between deterministic and stochastic time series. There are other methods introduced in the literature, such as the BDS test mentioned above [3], Diks test for reversibility [9], classification based on prediction [5], etc. For a good review on this subject see [27].

Knowledge of the presence of determinism and nonlinearities in a time series can be used to build better predictors. Many nonlinear methods of prediction have been introduced in the literature. Among these methods the local methods of prediction are of particular importance. The main aim is to capture the local dynamics of the time series, when the effects of nonlinearities and the amplification of noise are not yet that strong. On a small scale the nonlinear dynamics can successfully be approximated by a linear function. This is the basic idea behind the local linear predictors - perhaps the most well-known prediction technique for nonlinear and chaotic time series. Other methods are closely related with methods of computer

learning, such as neural networks.

Prediction can also be useful for discrimination between deterministic and stochastic time series. For a discussion on classification by prediction see [5]. The main ideas behind these classification methods are often similar to the idea behind the methods based on the correlation dimension estimation. For example, one can study the quality of local linear predictors, applied to nonlinear time series, as a function of the order k of the autoregression (in (1.7)), i.e. the embedding dimension. The same reasoning as above tells us that if the time series comes from a deterministic dynamical system, then, if k is sufficiently large, considering an autoregression of even higher order will not bring any additional information. Therefore, an increase of the embedding dimension does not essentially improve the forecasts. In fact, it can even lead to worse predictions due to the growing effect of nonlinearities. Other argument against further increase of the embedding dimension is increasing the number of unknown parameters in the model that have to be estimated. For stochastic time series, however, due to the presence of noise, models with higher values of k may be required to obtain better predictions. (Here, however, the same argument against increasing k as above holds.) Since in nonlinear stochastic time series the effects of noise and nonlinearities are combined, an autoregression with intermediate values of k would provide the best forecasts. Another, quite elegant criterion was suggested by Casdagli [5] (the so-called DVS criterion). It is based on finding the optimal number of neighbours that is necessary to fit a local linear model. It exploits the ideas similar to those considered above. This test has been shown to work well not only for discriminating between deterministic and stochastic time series, but also for distinguishing nonlinear and linear stochastic models.

1.3 Chaotic dynamical systems

In this section we discuss some aspects of chaotic dynamical systems, define some general notions and give the main assumptions. We shall focus on systems with discrete time.

A *discrete time dynamical system* $(\mathcal{X}, T)$ is a pair consisting of the *state space* $\mathcal{X}$ - the set of all possible values, and the time evolution map $T : \mathcal{X} \longrightarrow \mathcal{X}$ - the law according to which a state evolves to other states at later times. For an initial state $x_0 \in \mathcal{X}$, the iterations of T give rise to a *trajectory*, or an *orbit* $\{T^n x_0\}_{n \in \mathbf{N}}$ (or, if T is invertable, $n \in Z$).

A dynamical system is related to a time series by means of the *read-out function*, or the *observable function* $f : \mathcal{X} \longrightarrow \mathbf{R}$, which assigns to each possible state in $\mathcal{X}$ the recorded value when the system is in that state. (As was mentioned in the section above, this is a particular model of a nonlinear time series.)

The problem of discrimination between deterministic and stochastic time series does not make sense unless some conditions on the underlying dynamical system are imposed. Without these conditions, as we shall see, the discrimination is impossible in principle, since a dynamical system which does not satisfy these conditions can generate any time series.

We shall assume that the state space is finite dimensional, i.e. that it is a (subset of) a finite dimensional Euclidean space $\mathcal{X} \subseteq \mathbf{R}^d$, or can be embedded into such a vector space. Also we shall assume that all positive orbits of the dynamical system

(i.e. those indexed by $\mathbf{N}^+$) $\{T^n x_0\}_{n\geq 0}$ are bounded. Now we present two examples of dynamical systems which violate one of these assumptions and can generate any time series.

First, let a state space be infinite dimensional: the vector space $\mathbf{G}$ of all functions $g: \mathbf{N} \longrightarrow \mathbf{R}$. Let the dynamical law be the map $T: \mathbf{G} \longrightarrow \mathbf{G}$, which assigns to the function $g \in \mathbf{G}$ the function Tg defined by $(Tg)(n) = g(n+1)$. Let the read-out function $f: \mathbf{G} \longrightarrow \mathbf{R}$ be given by $f(g) = g(0)$. For an initial state $g_0 \in \mathbf{G}$ the orbit is $g_0, Tg_0, T^2g_0, \ldots$, and the time series is $\{f(T^n g_0)\}_{n\in\mathbf{N}} = \{(T^n g_0)(0)\}_{n\in\mathbf{N}} = \{g_0(n)\}_{n\in\mathbf{N}}$. Now this time series can be made anything we want by choosing an appropriate initial state - function g_0.

The second example is a dynamical system with unbounded positive orbits. Let the state space be $\mathbf{R}$ and let the map $T: \mathbf{R} \longrightarrow \mathbf{R}$ be defined by $Tx = x + 1$. For $x_0 = 0$ the orbit is unbounded. Let the read-out function be $f: \mathbf{R} \longrightarrow \mathbf{R}$. Here we see that the corresponding time series $\{f(T^n x_0)\}_{n\in\mathbf{N}} = \{f(n)\}_{n\in\mathbf{N}}$ depends completely on the choice of the read-out function, and therefore can be chosen arbitrary as well.

These two conditions, together with the assumptions that both T and f are differentiable, allow us to avoid some obvious exceptional cases, such as those mentioned above.

A way to get an impression of the behaviour of a dynamical system is to study what happens asymptotically as $n \longrightarrow \infty$. The simplest case occurs when the trajectories converge to a single point or limit cycle, called the *attracting point* or the *attracting cycle*, respectively. In the latter case the limit behaviour is periodic. More complicated limit behaviour appears as *quasiperiodic motion*, when the trajectories are attracted to a d-dimensional torus. In these cases the limiting set $\mathcal{A} \subseteq \mathcal{X}$, which is called an *attractor*, is a simple geometrical object. It is possible that the attractor is none of the simple objects mentioned above, for example, it can be some Cantor-like set with a non-integer dimension. In such cases one speaks of a *strange attractor*.

There is no unique definition of attractor in the literature on dynamical systems. Here we shall not give an entirely precise mathematical definition of it. For a thorough discussion on different notions and definitions of an attractor see [19]. For our purposes we can say that an attractor is the limiting set where the experimental orbits $\{T^n x\}_{n\in\mathbf{N}}$ accumulate for large n. An example of an attractor - the attractor of Henon dynamical system - was given in Fig.1.9.

A more precise definition is that of an attracting set. The set $\mathcal{A}$ is called an *attracting set* with fundamental neighbourhood U, if it satisfies the following properties:

(1) *Attractivity:* for every open set $V: A \subset V$ we have
$\{T^n x: x \in U\} \subset V$ for all sufficiently large n;
(2) *Invariance:* for all $x \in \mathcal{A}$ and all n we have $T^n x \in \mathcal{A}$.

Together with the two properties above one usually requires an attractor to be *irreducible* in some sense. In practice the attracting sets defined above are generally referred to as attractors. In what follows, when we talk about an evolution on the attractor, we actually mean "in a neighbourhood of the attractor".

The notion of *invariant measure* is associated with a dynamical system. A finite measure μ on the Borel σ-field $\mathcal{F}$ of $\mathcal{X}$ is called *T-invariant* if, for any set $B \in \mathcal{F}$, $\mu(T^{-1}B) = \mu(B)$. Since μ is finite, we can assume without loss of generality that μ is a probability measure, i.e. that $\mu(\mathcal{X}) = 1$. To indicate the link between the

dynamical system and the corresponding invariant measure, we shall sometimes write $(\mathcal{X}, T, \mu)$ for a dynamical system.

A transformation T (if it is a homeomorphism) always has at least one invariant measure associated with it (if the state space is compact) [17]; in fact it can have more. Typically, there are many invariant measures on an attractor. Of particular interest for us are the so-called *ergodic measures.*

A T-invariant measure μ is called *ergodic* if all T-invariant sets in $\mathcal{X}$ (i.e. all $A \in \mathcal{F}$ for which $T^{-1}(A) = A$) have measure $\mu(A)$ either 0 or 1.

It is exceptional that an attractor carries only one ergodic invariant measure. In typical cases there are uncountably many distinct ergodic measures. Intuitively, however, it seems that there is one natural, *physical measure* on an attractor produced by the evolution - the one that describes how much time a trajectory spends on average in various parts of the attractor. A candidate for such a physical measure is the so-called *SRB measure* (from Sinai, Ruelle, Bowen). It can be specified by taking a point x_0 at random with respect to the Lebesgue measure on $\mathcal{X}$, and, for $A \subset \mathcal{X}$, considering the time averages

$$\frac{1}{n} \sum_{k=0}^{n-1} \delta_{T^k x_0}(A),$$

where δ is the Dirac delta-measure. Then the SRB measure ρ is defined for all x_0 in a set $B \subset \mathcal{X}$ with Lebesgue measure $m(B) > 0$ by

$$\rho(A) = \lim_{n \longrightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} \delta_{T^k x_0}(A). \tag{1.9}$$

For certain dynamical systems (the so-called *Axiom A systems*) it has been shown (Ruelle [24]) that the limits in (1.9) exist for all x_0 in a set of positive Lebesgue measure, and provide a unique SRB measure with support on the attractor.

We shall not go into further details as to when such a measure exists on an attractor, or how it is related to other ergodic measures. For a complete discussion on these aspects see [23]. Our further investigation will concentrate more on the properties of the physical ergodic measures carried by attractors, which are naturally defined by a dynamical evolution or the corresponding time series.

An attractor and an invariant measure carried by it provide us a global description of the asymptotic behaviour of a dynamical system. The dynamic on the attractor itself does not need to be simple. For some dynamical systems the evolution on the neighbourhood of the attractor may depend sensitively on initial conditions, i.e. the trajectories starting in nearby initial points diverge from each other at an exponential rate and after some time can be found in totally different parts of the attractor. This property, called the *sensitive dependence on initial conditions*, is the characteristic feature of chaos.

We say that the evolution on the attractor $\mathcal{A}$ exhibits *sensitive dependence on initial conditions* if there is a positive constant C such that for any $\epsilon > 0$ and $x \in \mathcal{A}$, there are $x' \in \mathcal{A}$ and $N > 0$ such that

(1) $\rho(x, x') < \epsilon$

(2) $\rho(T^N x, T^N x') > C$,

where $\rho(\cdot, \cdot)$ is some distance on $\mathcal{A}$.

Sensitive dependence on initial conditions is related to the so-called *Lyapunov*, or *characteristic exponents.* It measures the mean exponential rate at which nearby

trajectories diverge with time. Suppose that $\{x_i\}_{i\in\mathbf{N}}$ is the orbit of a chaotic discrete dynamical system corresponding to the initial condition x_0. If we slightly change the position of the initial point: $x_0 \longrightarrow x_0+\delta x_0$, the point at time n will also be different. In general, one might expect that, if δx_0 is small, δx_n is also small. But, due to the sensitive dependence on the initial condition, when n becomes large, the small distance between the initial values grows exponentially fast: $\delta x_n \sim \delta x_0 \exp(\lambda n)$, where the mean rate of divergence of the trajectories λ is the Lyapunov exponent.

A 1-dimensional dynamical system has exactly one Lyapunov exponent. It can be defined as

$$\lambda = \mathbf{E}_\mu(\log|T'(X)|), \tag{1.10}$$

where the expectation is taken with respect to an ergodic invariant probability measure, the existence of which is assumed. Another way is to consider

$$\lambda = \lambda(x) = \lim_{n\longrightarrow\infty} \frac{1}{n}\log|(T^n)'(x)|, \tag{1.11}$$

where $(T^n)'_x$ is the derivative of T^n evaluated at x. In the case of ergodic invariant measure μ, the ergodic theorem implies that the limit in (1.11) exists and is constant μ-almost everywhere, and, moreover, the two definition (1.10) and (1.11) of λ are equivalent.

d-dimensional dynamical system for $d > 1$ has a spectrum of d Lyapunov exponents $\lambda_1 \geq \lambda_2 \geq \cdot \geq \lambda_d$, measuring the exponential rate of divergence or contraction in different directions. They are defined as logarithms of the eigenvalues of the matrix

$$\Lambda_x = \lim_{n\longrightarrow\infty}[(D_xT^n)^T D_xT^n]^{1/2n} \tag{1.12}$$

where D_xT^n is the matrix of the partial derivatives of the components of T^n at x. In the case of an ergodic invariant measure μ, The multiplicative ergodic theorem of Oseledec [22] implies that the limit in (1.12) exists and is constant μ-almost everywhere if μ is an ergodic measure.

For a one-dimensional dynamical system positivity of the Lyapunov exponent implies sensitive dependence on initial conditions. In higher dimensions the existence of at least one positive Lyapunov exponent is evidence for the sensitive dependence on initial conditions, i.e. chaotic behaviour. This happens because the behaviour of the system is essentially determined by the largest Lyapunov exponent λ_1. If it is positive, then there is at least one direction in which the expansion at an exponential rate takes place. Then the neighbouring orbits diverge, which results in chaotic behaviour. In general, the exponential rate of growth of distances in $\mathcal{X}$ when iterating T is given by λ_1, and t he rate of growth of the d-dimensional volume element by $\lambda_1 + \lambda_2 + \cdots + \lambda_d$.

The exponential divergence of trajectories means that two orbits that are so close that they cannot be distinguished at time zero, become distinguishable after some time. In this case we may speak of the "creation of information". The *entropy* is another characteristic of a chaotic dynamical system. It describes the asymptotic rate of production of information when iterating T. The subject of entropy is outside the scope of this manuscript, and so, we shall not define it here.

As we mentioned above, positive orbits of deterministic dynamical systems can be attracted to low-dimensional subsets of the state space. Other quantities which give us an idea about the complexity of the system are a dimension of the attractor

and of an invariant measure on the attractor. Loosely speaking, a dimension is a measure of the amount of information needed to specify points accurately. Precise definition of a dimension can be obtained in many ways. For example, the so-called *box-counting dimension*, or *capacity* of the attractor $\mathcal{A}$, is defined via the minimum number of closed balls required to cover $\mathcal{A}$. Let $\epsilon > 0$ be small and suppose that $\mathcal{A}$ is an interval in $\mathbf{R}$. Then the required number of balls of diameter ϵ needed to cover $\mathcal{A}$ is approximately the reciprocal of ϵ. If $\mathcal{A}$ is a rectangular plane segment in $\mathbf{R}^2$, then this number is inversely proportional to ϵ^2, etc. This scaling behaviour motivates the following definition: if the number of closed balls of radius ϵ needed to cover a set E scales as $(1/\epsilon)^{d_B(E)}$, then $d_B(E)$ is the box-counting dimension of E. Other dimensions studied in the literature are: the Hausdorff dimension, the information dimension, the correlation dimension; in the context of strange attractors they are often referred to as "fractal dimension", because non-integer values are possible. For a good review on the question of dimensions see [7].

For purpose of statistical estimation, the correlation dimension is most appropriate because it characterises the invariant measure on the attractor, and it is relatively easy to estimate. Above we defined the correlation dimension in terms of a time series. Here we shall define the correlation dimension of an invariant measure on the attractor. It is again defined via the *correlation integral*:

$$C(r) = (\mu \times \mu)\{(X, Y) : \| X - Y \| \leq r\}$$

for $r > 0$, where X, Y are independently chosen points on the attractor. If the correlation integral scales as $C(r) \sim \text{const} \cdot r^\alpha$ as $r \longrightarrow 0$, then the exponent α is called the correlation dimension of the invariant measure μ.

To give a flavour of other dimensions, we mention one more here, the so-called *limit capacity*. On the contrary to correlation dimension, which is a characteristic of an underlying invariant probability distribution, this one is a characteristic of sets, namely of bounded subsets of the reals, or of a finite dimensional vector space. In order to define the limit capacity of such a set S, we first introduce for each ϵ a number $K(\epsilon)$ which is the minimal number of points one needs to cover S with the ϵ-neighbourhoods of these points. Just like in the case of correlation integral, we expect $K(\epsilon)$ to behave like $\epsilon^{-dimension}$ as $\epsilon \longrightarrow 0$ ("verify" this by analysing simple examples such as unit interval, unit square, unit cube), we define limit capacity in terms of $K(\epsilon)$ as

$$d = -\lim_{\epsilon \longrightarrow 0} \frac{\ln K(\epsilon)}{\ln \epsilon} \tag{1.13}$$

provided this limit exists.

Much of the essential information about the dynamical system $(\mathcal{X}, T)$ is contained in characteristics such as the fractal dimension of the attractor, the entropy or the Lyapunov exponents. In practice the dynamical system itself is rarely known, and, even if it is known, it is often impossible to compute these quantities precisely. Then we face the problem of estimating these characteristics from a single orbit of the dynamical system: $(x_0, Tx_0, T^2x_0, ...)$. As we have argued above, in reality the problem is even more complicated: what we actually observe is not the orbit in the original state space $\mathcal{X}$ but a real-valued time series $\{y_n\}_{n \in \mathbf{N}}$ of measurements made on the orbit, and all estimates must be based on $\{y_n\}$. The first step toward estimation is always the reconstruction of the original dynamics by embedding one dimensional data into a higher-dimensional space.

1.4 State space reconstruction

The technique of state space reconstruction is based on the reconstruction theorem of Takens [28]. It enables us to reconstruct the dynamics of a system from an observed time series.

Let $(\mathcal{X}, T)$ be a dynamical system with finite-dimensional state space $\mathcal{X}$ and bounded positive orbits $\{T^n x\}_{n\geq 0}$. To relate a dynamical system to a time series, we introduced a read-out function $f : \mathcal{X} \longrightarrow \mathbf{R}$. Usually $f(x)$ represents a real-valued measurement made on a point $x \in \mathcal{X}$. If $(x_0, Tx_0, T^2x_0, ...)$ is an orbit of the dynamical system with initial state x_0, then the corresponding time series is obtained by applying f to each point of the orbit: $(f(x_0), f(Tx_0), f(T^2x_0), ...)$.

Let both T and f be continuously differentiable (or $T, f \in C^1$). Define the vector-valued *reconstruction map* $\mathrm{Rec}_k : \mathcal{X} \longrightarrow \mathbf{R}^k$ by

$$\mathrm{Rec}_k(x) = (f(x), f(Tx), ..., f(T^{k-1}x)) \in \mathbf{R}^k. \tag{1.14}$$

Theorem (Takens, [28]) *In the Cartesian product of the space of C^1-mappings on $\mathcal{X}$ and the space of C^1-functions from $\mathcal{X}$ to $\mathbf{R}$ there exists an open and dense subset U, such that if $(T, f) \in U$, then the reconstruction map Rec_k, defined in (1.14), is an embedding, whenever $k > 2 \cdot dim(\mathcal{X})$.*

We give some remarks to clarify this statement and its consequences.

1. The interpretation of the condition $(T, f) \in U$ is that the statement of the theorem holds for "almost all", or "generic" pairs (T, f). For a more complete discussion on the notion of *genericity* (and on reconstruction problems in general) see [26]. The reason why these conditions have to be imposed, is to exclude exceptional cases, for which the reconstruction will obviously fail. These are cases such as the observable function f being constant, or the map T being the identity. In the former case, applying a constant function to the points of the orbit destroys all the information about the orbit, so it is impossible to make the reconstruction. In the later case, the second, the third, and all other consecutive points of the orbit do not contain any additional information, so the reconstruction also fails here.

2. The key word in the theorem is "embedding". The transformation from $\mathcal{X}$ to $\mathbf{R}^k$ given by the reconstruction map Rec_k is an embedding if the mapping of $\mathcal{X}$ into $Rec_k(\mathcal{X})$ is continuously differentiable with continuously differentiable inverse. In less technical terms it means that $\mathcal{X}$ and its image under the reconstruction map $\mathrm{Rec}_k(\mathcal{X})$ are the same up to a diffeomorphic transformation. Furthermore, in the presence of an attractor $\mathcal{A} \subset \mathcal{X}$ the reconstruction maps transform $\mathcal{A}$ into its image in $\mathbf{R}^k$. Under this transformation the differential structure of the attractor is preserved. Also the invariant measure on the reconstructed attractor is an image of the invariant measure of the original transformation. Moreover, since a diffeomorphism restricted to a bounded set gives only the distortion of distances by a factor which is bounded and bounded away from zero, the correlation dimensions of the original and the reconstructed attractor are the same (see [28]).

Here we also mention that this theorem is a variation of the well-known Whitney embedding theorem, but restricted to mappings of special form, namely the reconstruction maps.

3. For practical purposes, i.e. for reconstructing the attractor from the observed

real-valued time series $\{y_n\}_{n\in\mathbf{N}}$, we define the reconstruction vectors by

$$X_n = \mathrm{Rec}(T^n x_0) = (y_n,\ y_{n+1},\ \ldots,\ y_{n+k-1})$$

for $k \geq 2\dim(\mathcal{X}) + 1$. Then the trajectory $\{T^n x_0\}_{n\in\mathbf{N}}$ in $\mathcal{X}$ is an image of the trajectory generated by the sequence of reconstruction vectors $\{X_n\}_{n\in\mathbf{N}}$ in $\mathbf{R}^k$ under the diffeomorphism. Moreover, the reconstruction vectors accumulate in $\mathbf{R}^k$ in a neighbourhood of the limit set which is diffeomorphic to the attracting set of orbits of T in $\mathcal{X}$. The reconstructed invariant measure on this limit set, the correlation dimension and some other characteristics that are defined by the sequence $\{X_n\}_{n\in\mathbf{N}}$ do not depend on the read-out function f but describe intrinsic properties of the dynamics.

4. If it is not known that the time series is produced by a deterministic dynamical system, e.g. because significant noise is present, then it is harder to interpret the results of our reconstruction. Also one should not forget that the choice of the embedding dimension k is essential. The problem of selecting a sufficiently large k can be difficult since the dimension of $\mathcal{X}$ is usually unknown, and, hence, the criteria $k > 2 \cdot \dim(\mathcal{X})$ of the Takens theorem cannot be applied directly. There is a number of methods for choosing k introduced in the literature (see [6], [5]). We shall not concentrate on this problem here and simply assume that a sequence of reconstruction vectors $\{X_n\}_{n\in\mathbf{N}}$ or z, $X_i \in \mathbf{R}^k$ is obtained from the outcome of the actual time series $\{y_n\}$. All further analysis and the estimates considered below will be based on this sequence. The question, for which dynamical systems the sequence of reconstruction vectors provides a stationary stochastic process with respect to some ergodic probability measure, is in general very complex. For certain types of systems, the so-called *Axiom A systems*, this has been shown by Ruelle [24]. We shall not address this problem here and assume that a given sequence $\{X_n\}$ is the outcome of a stationary stochastic process.

1.5 Dimension estimation

Estimating the fractal dimension of a strange attractor from a corresponding time series has attracted considerable attention in the past few years and has become one of the main tools in the analysis of the underlying dynamics. Of all types of dimensions, most attention has been given to the *correlation dimension.* This is mainly because this type of dimension is easier to estimate than others and also because it provides a good measure of the complexity of the dynamics, i.e. of the number of active degrees of freedom.

Let $(\mathcal{X}, T, \mu)$ be a dynamical system. Recall that the correlation dimension is defined via the correlation integral

$$C(r) = \mathbf{P}\{(X, Y) : \| X - Y \| \leq r\}.$$

where X, Y are independent, each having marginal distribution μ. If there exists a constant α, such that

$$C(r) \sim \mathrm{const} \cdot r^{\alpha} \quad \text{as} \quad r \longrightarrow 0, \tag{1.15}$$

then α is called the correlation dimension of μ. Note that

$$\alpha = \lim_{r\longrightarrow 0} \frac{\log C(r)}{\log r}, \tag{1.16}$$

provided this limit exists.

The correlation dimension is a characteristic of the underlying invariant measure μ. In a certain sense it characterises how smoothly μ is distributed over the attractor: if μ is a point measure, then $\alpha = 0$, and if μ is absolutely continuous with respect to Lebesgue measure, then α equals the topological dimension d of $\mathcal{X}$. These are two boundary cases, in general $0 \leq \alpha \leq d$.

A number of procedures for estimating the correlation dimension has been introduced in the literature. In the next sections we review some of these estimators and their properties.

1.5.1 Grassberger-Proccacia estimator

Grassberger and Proccacia [10] suggested a procedure of estimating α which immediately became widely used by mathematicians and applied scientists. According to their method, we first estimate the correlation integral on the basis of the part of a stationary sequence of reconstruction vectors $\{X_i\}_{i=1,\ldots,n}$ by the sample correlation integral

$$C_n(r) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \mathbf{1}(\| X_i - X_j \| \leq r) \tag{1.17}$$

(note that $C_n(r)$ is the proportion of pairs in the sample $X_1, ..., X_n$ no more than the distance r apart). For distance $\| \cdot \|$ one usually takes the maximum norm, i.e. for a k-dimensional vector $\mathbf{x}$ it is $\| \mathbf{x} \| = \max_{1 \leq i \leq k} |x_i|$. The estimate of α is, however, independent of the choice of the norm (see [28]), and the max norm is taken as the most convenient one. The correlation integral $C_n(r)$ is estimated for a vector of distances $(r_1, ..., r_l)$ and the Grassberger-Proccacia estimate for the correlation dimension $\hat{\alpha}_n^{GP}$ is then obtained by studying the least-squares linear regression of $\log C_n(r)$ vs. $\log r$ (or, alternatively, the weighted regression).

In fact, the correlation dimension was initially introduced and defined by Grassberger and Proccacia via the sample correlation integral $C_n(r)$ (and not $C(r)$) as a double limit

$$\alpha^{GP} = \lim_{r \longrightarrow 0+} \lim_{n \longrightarrow \infty} \frac{\log C_n(r)}{\log r}. \tag{1.18}$$

The variance of $C_n(r)$ and of $\hat{\alpha}_n^{GP}$ can be consistently estimated from data by different methods, such as using U-statistics, Monte-Carlo simulation or the bootstrap. The variance estimation problem is very important for applications, since it allows us to compute confidence intervals for the estimates of the dimension.

When estimating the dimension from a time series in practice, a number of important issues should be considered. First, for the purposes of discrimination of chaotic time series, the dimension should be estimated for a number of embedding dimensions k, up to some reasonable number. Here one should keep in mind that for growing embedding dimension the estimates of correlation dimension can only make sense if one has a very long time series, due to the so-called "curse of dimensionality" (explain this). In literature criteria such as "number of observations $\sim 10^k$" are mentioned.

Second, the choice of the region of ϵ's for correlation integrals estimates is important. Here the problem is two-fold: if ϵ's are taken too small, there will be none or very few pairs in the dataset withing such a small distances and the estimates of the correlation integral will be unreliable; on the other hand, if ϵ's are taken too

big, then you might be out of the region where the scaling relationship $C(\epsilon) \sim \epsilon^{\alpha}$ holds. In practice, these two problems should be balanced by estimating correlation integrals for various ϵ's and looking for a region where a reasonable linear relation between $\ln C(\epsilon)$ and $\ln \epsilon$ can be observed. Finally, confidence bounds for dimension estimates should be given.

In general, one should always exercise extreme caution when drawing conclusions from dimension estimation exercises, especially for economic and financial time series which can be particularly nonstationary and noisy. Literature in this area already contains enough notorious conclusions of presence of chaos and determinism in financial data, some of the finest examples of this can be found in e.g. book of Peters "Chaos and order in capital markets".

1.5.2 Takens estimator

Takens proposed an alternative approach of estimating the correlation dimension [29]. Assuming again that the exact scaling $C(r) = cr^{\alpha}$ for $r < r_0$ holds, Takens first considered estimating α from i.i.d. realisations $R_i = \| X_i - Y_i \|$ of the distance $\| X - Y \|$, where X_i and Y_i are independent each having the marginal distribution μ.

Applying ideas of Maximum Likelihood estimation, he suggested to estimate α by

$$\hat{\alpha} = -\frac{N}{\sum_{i=1}^{N} \log R_i}, \tag{1.19}$$

where N is a number of distances considered.

In general, independent realisations of the distances $\| X - Y \|$ will not be available (moreover, X_i's themselfs are, in most cases, not independent) and thus a modification of the estimator (1.19) may become necessary. Given a finite segment $X_1, ..., X_n$ of a stationary sequence of the reconstruction vectors, we can form $n(n-1)/2$ pairwise distances $\| X_i - X_j \|$. Takens suggested to use the estimator

$$\hat{\alpha}_n^T = -\left(\frac{2}{n(n-1)} \sum_{1 \le i < j \le n} \log \frac{\| X_i - X_j \|}{r_0} \right)^{-1}. \tag{1.20}$$

In fact, an estimator similar to (1.20) was first introduced by Hill [14] in the context of estimating the tail index of a distribution: suppose we have a random sample $T_1, T_2, ..., T_N$ from a distribution F which behaves at 0 as

$$F(x) \sim \text{const} \cdot x^{\alpha} \quad \text{as } x \longrightarrow 0,$$

and we want to estimate α without making assumptions about the form of F elsewhere. Assuming that actually $F(x) = Cx^{\alpha}$ when $x \le x_0$, for some known x_0 and some constant C, the conditional Maximum Likelihood estimator of α (Hill's estimator) is given by

$$\hat{\alpha}^H = -\left(\frac{1}{m} \sum_{i=1}^{m} \log \frac{T_{(i)}}{T_{(m+1)}} \right)^{-1}, \tag{1.21}$$

where $T_{(1)} \le T_{(2)} \le ... \le T_{(m)} \le T_{(m+1)} \le x_0$ are the order statistics which are below the threshold x_0. Note that the estimators (1.20) and (1.21) coincide up to a

scaling factor: in Hill's estimator observations are scaled by the last order statistics which is still below the threshold, and in Takens estimator by the threshold itself.

From the practical point of view, Takens estimator has pluses as well as minuses when compared to Grassberger-Proccacia estimator. It assumes exact scaling (which is impossible to check in practice) and the cutoff distance r_0 is needed. Also it can be quite sensitive to very small distances due to the logarithm. This problem however can be solved by introducing also a lower cutoff distance r_1 and considering distances only bigger than it. On the other hand, this estimator is computationally more efficient and does not require ad-hoc choice of linear scaling region.

2

Prediction

Prediction of future observations is an important problem in the analysis of time series. Given a time series $\{Y_n\}_{n\in\mathbf{N}}$ the question is to find a predictor for $Y_{n+s},\ s \geq 1$, as a function of a certain number of previous observations, i.e. we are looking for

$$\begin{aligned} \hat{Y}_{n+s} &= F(y_n, y_{n-1}, ..., y_{n-k+1}) \\ &:= \mathbf{E}(Y_{n+s}|Y_n = y_n, ..., Y_{n-k+1} = y_{n-k+1}). \end{aligned} \tag{2.1}$$

In traditional time series analysis, F is restricted to a parametric, usually linear form and a linear autoregression with Gaussian innovations (AR of ARMA) is applied to estimate F. However, as we argued in the introduction, for many time series the function F can be nonlinear for many time series. In that case linear predictions are not necessarily appropriate.

For example, if $f : \mathbf{R} \longrightarrow \mathbf{R}$ is a nonlinear chaotic map, then the time series obtained by $y_{n+1} = f(y_n)$, for some $y_0 \in \mathbf{R}$, satisfies the first order autoregression model with nonlinear autoregression function f. Usually there is noise present in the time series. What we actually observe is

$$\tilde{y}_{n+1} = y_{n+1} + \epsilon_{n+1} = f(y_n) + \epsilon_{n+1},$$

where ϵ_n are zero-mean errors with finite variance.

Theory of chaotic dynamical systems shows that even simple nonlinear dynamical systems can give rise to time series which exhibit highly erratic and seemingly random behaviour. Well-known examples, already mentioned above, are: the logistic map $f(x) = 4x(1-x)$, the Henon map, the Lorenz system of 3 coupled nonlinear differential equations.

Time series coming from a chaotic dynamical system in general satisfy a nonlinear autoregression model. This can be briefly explained by the following. Recall that the reconstruction vectors $X_1, X_2, ... \in \mathbf{R}^k$ are obtained from the observed time series $\{y_n\}_{n\in\mathbf{N}}$ by

$$X_i = (y_i, y_{i+1}, ..., y_{i+k-1}).$$

Takens reconstruction theorem implies that, in generic situations, if k is sufficiently high, the reconstruction vectors X_i are accumulating in $\mathbf{R}^k$ in the neighbourhood of an object, diffeomorphic to the original attractor, and, moreover, that there is dynamical map induced on the space of the reconstruction vectors. This map is also diffeomorphic to the original chaotic map T, i.e. in the absence of noise there exists a map $G : \mathbf{R}^k \longrightarrow \mathbf{R}^k$, diffeomorphic to T, such that for all i

$$X_{i+1} = G(X_i). \tag{2.2}$$

In terms of the original time series, the relation (2.2) means that there also exists a function $F : \mathbf{R}^k \longrightarrow \mathbf{R}$ such that

$$y_{i+k} = F(y_i, y_{i+1}, ..., y_{i+k-1}). \tag{2.3}$$

The original transformation T was assumed to be nonlinear, so F is also a nonlinear function. The relationship (2.3) implies that the time series $\{y_i\}_{i\in\mathbf{N}}$ satisfies the kth order nonlinear autoregression model, where the autoregression function F is some unknown nonlinear function not restricted to any parametric form. If there is noise present in the system, we shall for simplicity again assume that it is additive measurement noise, i.e. that we observe

$$\tilde{y}_{i+k} = y_{i+k} + \epsilon_{i+k} = F(y_i, y_{i+1}, ..., y_{i+k-1}) + \epsilon_{i+k}, \tag{2.4}$$

where ϵ_i are mean zero and finite variance errors. For clarity's sake we shall write $\{y_i\}_{i\in\mathbf{N}}$ for the observed time series, even when an additive noise is present.

Relation (2.4) expresses the functional dependence of the next observation in the time series $\{y_i\}_{i\in\mathbf{N}}$ on the previous k observations. In many practical situation we are interested in the relation between the observation $s > 1$ steps ahead and the previous k values, i.e. in the nonlinear function F^s:

$$y_{i+s} = F^s(y_i, y_{i-1}, ..., y_{i-k+1}). \tag{2.5}$$

If (2.3) holds, F^s can be expressed as

$$F^s(\cdot) = \underbrace{F(F...(F}_{s \text{ times}}(\cdot))...).$$

In the case of chaotic time series, as well as other nonlinear time series (not necessarily chaotic), one expects nonlinear or locally linear methods of prediction to have an advantage over traditional linear methods.

A method of prediction frequently mentioned in the literature on chaotic time series is the so-called (k, ϵ)-method (or locally linear predictors). It is based on the assumption that it is best to capture the underlying dynamics locally, to reduce the effects of nonlinearity. On the local scale the nonlinear function F, if it is smooth enough, can be successfully interpolated by a linear function. The method can be briefly described as follows: suppose we have observed k past values of the time series $\underline{y} := y'_1, ..., y'_k$, and want to predict y'_{k+s}. We collect all vectors of length k from the time series which are within distance ϵ from $\underline{y}$, as well as the corresponding observations s steps after, and then apply a linear regression to the collected data. The pair (k, ϵ) is selected by minimising some measure of prediction error. This method turns out to be quite successful if the time series comes from a low-dimensional chaotic dynamical system and enough data are available.

In the next two sections we shall concentrate on another method: the kernel regression smoothing. This method also has a local character, but is more flexible.

2.1 Kernel autoregression estimation for time series

Kernel smoothing is a method from nonparametric statistics for estimating an unknown regression function. First we describe it briefly in a more general setting.

Suppose that random pairs (X_i, Y_i), $i = 1, ..., n$, are coming from the same distribution as a random vector (X, Y), where $X, X_i \in \mathbf{R}^k$, $Y, Y_i \in \mathbf{R}$, and suppose that there exists a function $r : \mathbf{R}^k \longrightarrow \mathbf{R}$ such that

$$Y_i = r(X_i) + \epsilon_i, \tag{2.6}$$

where ϵ_i are i.i.d. zero-mean observation errors with fixed finite variance. Here r, called the *regression function*, is an unspecified function and it is not restricted to any parametric form. It can also be defined as the conditional expectation of Y given that $X = x$:

$$r(x) = \mathbf{E}(Y|X = x)$$

(it is well-defined if $\mathbf{E}|Y| < \infty$). The components of the random vector X are also called the *explanatory variables*, and the random variable Y is regarded as the *response.* One is interested in approximating the general relationship between X and Y, the function r, on the basis of the sample $\{(X_i, Y_i)\}_{i=1,...,n}$.

A *kernel estimator* of the function r is

$$\hat{r}_{n,h}(x) = \hat{r}_h(x) = \frac{\sum_{i=1}^n \mathbf{K}_h(x - X_i)Y_i}{\sum_{i=1}^n \mathbf{K}_h(x - X_i)}, \tag{2.7}$$

where $\mathbf{K}_h$ for $k > 1$ is defined via the so-called *kernel function* $K : \mathbf{R} \longrightarrow \mathbf{R}$ by

$$\mathbf{K}(\underline{x}) = \prod_{l=1}^{k} K(x_l/h),$$

where $\underline{x} = (x_1, ..., x_k)$. $\mathbf{K}_h$ is called the *product kernel.* For $k = 1$ one takes $\mathbf{K}_h(y) = K(y/h)$. Here $h = h_n$ is a sequence of scaling parameters, called the *bandwidth sequence.* The estimator (2.7) is usually referred to as the *Nadaraya-Watson estimator.*

The kernel function $K(y)$ is taken to be a continuous bounded and symmetric real-valued function which integrates to 1. Usually these functions are also taken to be unimodal with maximum at 0. The most commonly used kernel functions are:

Triangle $(1 - |y|) \cdot \mathbf{1}(|y| \leq 1)$;

Epaneshnikov $\frac{3}{4}(1 - y^2) \cdot \mathbf{1}(|y| \leq 1)$;

Gaussian $\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}y^2)$.

In the context of time series, we can denote $X_i = (y_i, y_{i-1}, ..., y_{i-k+1})$ and $Y_i = y_{i+s}$. Then the kernel estimator of the autoregression function F is

$$\hat{F}_h(\underline{x}) = \hat{r}_h(\underline{x}) = \frac{\sum_{i=k}^{n-s} \mathbf{K}_h(\underline{x} - X_i)\, Y_i}{\sum_{i=k}^{n-s} \mathbf{K}_h(\underline{x} - X_i)}. \tag{2.8}$$

The application of the kernel method to time series was studied by Collomb (1984), Delecroux (1987) (see [11] and the references therein), and it was shown (Delecroux (1987)) that the estimator (2.8) is consistent if the time series is a stationary ergodic process and the bandwidth sequence converges to 0 at some specified rate. (See also [2].)

In applications a very important question is how to select the bandwidth. Selecting a bandwidth that is too small leads to a higher variance of the kernel estimator (this is called *undersmoothing*), and choosing a bandwidth that is too large increases its bias (*oversmoothing*). In practice we have to balance these two factors. Consider the average squared error:

$$ASE[\hat{F}_h] = \frac{1}{n} \sum_{i=k}^{n} [\hat{F}_h(X_i) - F(X_i)]^2.$$

which is the sum of the variance and the squared bias components. A value of the bandwidth which minimises the ASE is desirable.

The consistency results of Delecroux, Bosq, mentioned above, give the theoretical optimal rate of convergence of the bandwidth in terms of n, which balances the variance and the bias of the kernel estimate. In most cases this rate is $h = h_n \sim 0(n^{-1/5})$. However, these results are of purely theoretical value and do not tell us how to choose the bandwidth in practice. A data-driven approach for bandwidth selection is therefore necessary. Such an approach is called *cross-validation* and amounts to the following. We estimate the ASE by the *cross-validation function*

$$CV(h) = \frac{1}{n}\sum_{i=k}^{n}[Y_i - \hat{F}_{h,i}(X_i)]^2,$$

where $\hat{F}_{h,i}(X_i)$ is the leave-i-out kernel estimate of $F(X_i)$

$$\hat{F}_{h,i}(X_i) = \frac{\sum_{i\neq j}\mathbf{K}_h(X_i - X_j)\,Y_j}{\sum_{i\neq j}\mathbf{K}_h(X_i - X_j)}.$$

The cross-validation function $CV(h)$ is an asymptotically unbiased estimator of the ASE and has the advantage that it can be computed directly from the data. Then we choose the value of the bandwidth $\hat{h}_{opt}$ which minimises $CV(h)$. The question then is whether $\hat{h}_{opt}$ also (asymptotically) minimises the ASE? Haerdle and Vieu [13] have shown that, if the sequence (X_i, Y_i) is strongly mixing and some additional conditions on K, F and on the distribution of (X_i, Y_i) are satisfied, then the cross-validation procedure is asymptotically optimal, i.e. that the bandwidth chosen by means of the cross-validation asymptotically minimises the ASE.

2.2 Variation of kernel smoothing method

One of the main disadvantages of the methods mentioned above (traditional kernel smoothing, (k, ϵ)-method) is the fixed choice of the autoregression order k, which results into taking entirely into account the previous k observations and completely disregarding the rest. Moreover, in applications k is usually unknown and has to be estimated in some way (traditionally via linear models).

In this section we suggest the variation of kernel autoregression smoothing which overcomes this disadvantage and is in general more flexible.

One way to apply the kernel smoothing method for prediction in the case $k > 1$ involves product kernels. Another way is to define in some suitable way a distance between two k-dimensional vectors $\underline{x}, \underline{y}$: $d(\underline{x}, \underline{y})$, and define the kernel regression estimator via a one-dimensional kernel function as

$$\hat{F}_h(\underline{x}) = \frac{\sum_{i=1}^{n} K[d(\underline{x}, X_i)/h]\,Y_i}{\sum_{i=1}^{n} K[d(\underline{x}, X_i)/h]}. \tag{2.9}$$

The distance $d(\cdot,\cdot)$ can be taken as the Euclidean or the maximum distance. However, the choice of these distances would be purely formal and would not exploit the fact that the vectors X_i's are parts of a time series, i.e. sequences of observations ordered in time. Here we suggest a variation of the estimator (2.9) and choose a

distance between the vectors which takes the specifics of the time series setting into account.

We define the distance between two vectors $\underline{x} = (x_1, x_2, ..., x_k)$ and $\underline{y} = (y_1, y_2, ..., y_k)$ as

$$d(\underline{x}, \underline{y}) = \sum_{i=1}^{k} (x_i - y_i)^2 \gamma_i, \tag{2.10}$$

where $\{\gamma_i\}_{i=1}^{k}$, $\gamma_i \in [0, 1]$, is a collection of weights which we put on each of the differences between the coordinates.

In general, we will chose decreasing weights $\gamma_1 \geq \gamma_2 \geq ... \geq \gamma_k$, expressing the idea that the influence of past observations on the prediction should be discounted as the time lag grows. More specifically, we suggest to take

$$\gamma_i = \gamma^i, \tag{2.11}$$

for some $\gamma \in (0, 1]$. One motivation for this is the exponential divergence of trajectories (whose starting points are close) in chaotic dynamical systems. However, this technique is appropriate for many time series, including those with stochastic properties. Thus, the more recent observations give more precise information about the present state of the system, and the information decreases exponentially as time passes.

Observations further in the past are less relevant for future predictions since they are most affected by noise. In chaotic dynamical systems this is expressed via the notion of the sensitive dependence on initial conditions. The exponential growth of errors, which are always present in a real-life systems, makes the chaotic evolution self-independent of its own past. Quantitatively, this is described by the Lyapunov exponents of a chaotic map. The mean exponential rate with which nearby orbits diverge with time is measured by the highest Lyapunov exponent, and the existence of at least one positive Lyapunov exponent is evidence for the sensitive dependence on initial conditions. In the context of chaotic time series we are more interested in backward divergence of orbits, which is responsible for decreasing influence of the past observations on the future ones. In that case the mean rate of divergence of backwards orbits is measured by the highest Lyapunov exponent of the inverse map T^{-1} (provided T is invertable). It is given by the inverse of the lowest Lyapunov exponent of the original map T.

For our applications this situation can be illustrated by the following example. If $\underline{x} = (x_1, ..., x_k)$ and $\underline{y} = (y_1, ..., y_k)$ are segments of two orbits of an invertable chaotic dynamical system, or two parts of the same orbit, and if $\| x_k - y_k \| = \delta$, then, due to the exponential divergence of backward trajectories, $\| x_{k-1} - y_{k-1} \| = \delta e^{\lambda}$, $\| x_{k-2} - y_{k-2} \| = \delta e^{2\lambda}$, etc., where λ is the highest Lyapunov exponent of the inverse of the underlying chaotic map. The distance between $\underline{x}$ and $\underline{y}$ is:

$$d(\underline{x}, \underline{y}) = \sum_{j=1}^{k} \| x_{k-j} - y_{k-j} \|^2 \gamma^j = \sum_{j=1}^{k} \delta e^{2j\lambda} \gamma^j \tag{2.12}$$

and it is completely determined by δ, while the factors $e^{j\lambda}$ indicate the natural divergence of trajectories with the rate λ. This reasoning suggests taking $\gamma = e^{-2\lambda}$ to "discount" for this exponential divergence. Due to the state space reconstruction, the above reasoning also applies when $\underline{x} = (x_1, ..., x_k)$ and $\underline{y} = (y_1, ..., y_k)$ are not parts of an orbit, but segments of a chaotic time series.

In practice we will choose γ in a different way, based on optimalisation of the value of γ in an experimental way. This method is explained in more detail below.

We believe our method is more flexible than that of locally linear predictors, as well as the other methods mentioned above. Rather than having sharp cut-off points (k, ϵ) in time we have a smoother way to assign the significance of past observations. The choice of the parameter γ in a way replaces the choice of the order of autoregression k. By taking the decreasing weights (2.11), the choice of k becomes less important, because the dependence on the previous observations is not cut off at the number k, but decreases smoothly with the decrease of γ_i.

The influence of the parameter γ on our kernel estimate is comparable to the bandwidth influence. Thus, the choice of γ as well as of h determines the quality of our predictions. At the same time the choice of γ and of h is not independent: the bigger the bandwidth h, the higher the value of γ we should choose. Again we use a cross-validation algorithm. But since the parameters h and γ are bound together, the selection of their optimal values should be carried out simultaneously, i.e. we choose the *optimal pair* $(h, \gamma)_{opt}$ as

$$(h, \gamma)_{opt} = \text{argmin}\{CV(h, \gamma)\},$$

where $CV(h, \gamma)$ is a cross-validation function of the estimator (2.9).

The consistency of our estimator (2.9) follows from standard results for the traditional kernel estimator (see [11], [2]). The asymptotic optimality of the double cross-validation procedure can possibly be seen using similar arguments to those found in [13].

Taking decreasing weights γ_i is not the only possible choice, and sometimes not the most efficient one. For some time series not the most recent observation(s), but those further in the past have more influence on future observations. Consider the following example.

Suppose that the time series $\{x_n\}_{n \in \mathbf{N}}$ is obtained by

$$x_{n+1} = 0.01 x_n + f(x_{n-t}) + \epsilon_n,$$

where $f : \mathbf{R} \longrightarrow \mathbf{R}$ is a nonlinear chaotic map (e.g. a logistic map) and ϵ_n are mean zero and finite variance errors. Here the value of the next observation depends much more on the value of the observation t time units in the past than on the previous one. Consequently, γ_t (and, possibly, also γ_{2t}, γ_{3t}, etc.) should be taken significantly bigger than γ_1 and other weights.

How can we recognise such a time series? It is usually not possible by just observing its plot, because the chaotic evolution f produces the time series which appears random. We generated the following time series:

$$x_n = 0.01 x_{n-1} + f(x_{n-4}) + \epsilon_n, \quad n = 1, ..., 100,$$

with $f(x) = 4xe^{-x^2}$, and the i.i.d. errors $\epsilon_i \sim \mathcal{N}(0, 0.05)$ (Fig.7.1).

The plot of this time series does not indicate that the essential dependence is on a delay of 4. The so-called *delay-s maps*, i.e. plotting x_i vs. x_{i+s}, can be helpful. On Fig.7.2 shows the delay-1 map, and we see that it does not have any structure. But the delay-4 map (Fig.7.3) immediately reveals the deterministic structure of the time series. Fig.7.4 shows the plot of the autocorrelations, and it is clear that the autocorrelations of lags that are a multiple of 4 are significantly larger than

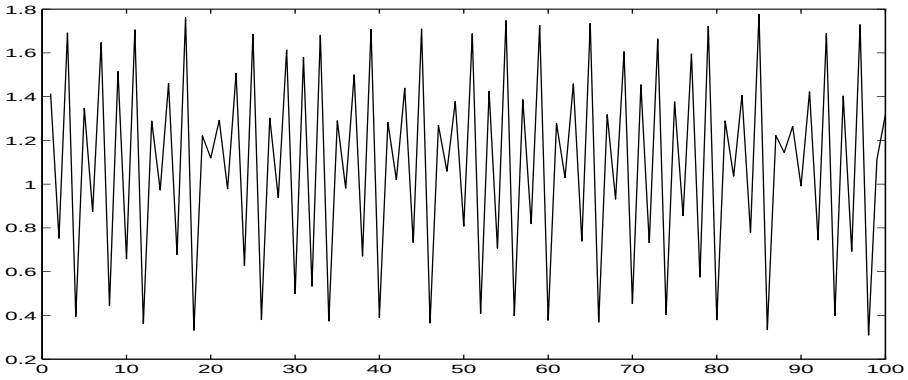


FIGURE 2.1. Time series x_n

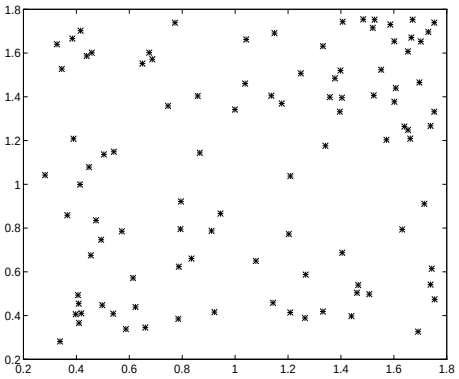


FIGURE 2.2. Delay-1 map

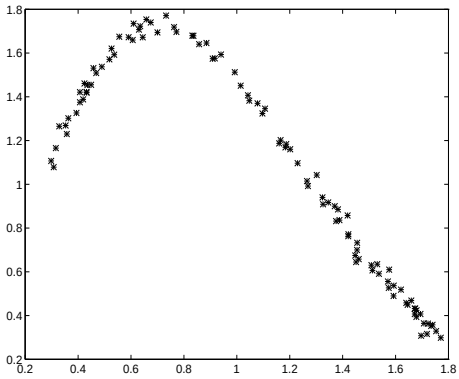


FIGURE 2.3. Delay-4 map

any other. An even more informative picture is obtained by examining the so-called *partial autocorrelations*. For the precise definition of the partial autocorrelation see Brockwell and Davis [4], pp. 98-102. The partial autocorrelation has the following meaning: its value at lag k indicates the amount of the additional information obtained from considering the linear autoregression model of order k instead of order $k-1$. Although the partial autocorrelations are defined in terms of a linear model, they certainly do show at which lags the dependence is most significant. The plot of partial autocorrelations for our working example is given in Fig.7.5.

The plot shows that the largest partial autocorrelations are at lags 4 and 12, with less significant ones at 1 and 3. This information can be used for determining the weights γ_i. For instance, γ_i's can be taken proportional to the partial autocorrelations. In terms of dynamical systems, the quantity analogous to the partial autocorrelation can be considered. This quantity is called the *mutual information*. Here we shall not go into detail as to how to define it, we shall only mention that this quantity with respect to an orbit of a dynamical system carries essentially the same meaning as the partial autocorrelation with respect to a time series. It can be also consistently estimated from a time series and, consecutively, used for determining the weights γ_i.

The example we just considered is more an exception than the rule. For most nonlinear time series geometrically decreasing weights $\gamma_i = \gamma^i$ is the most natural

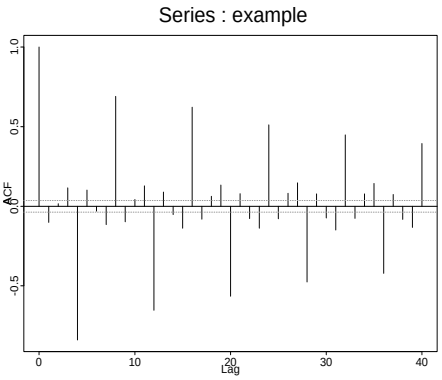


FIGURE 2.4. Autocorrelation function

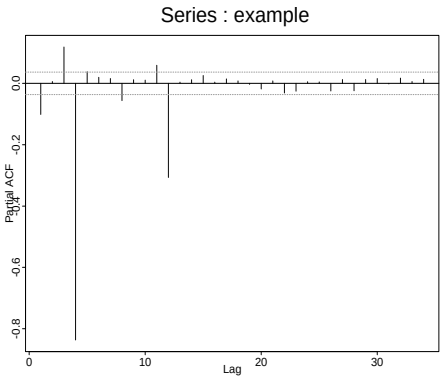


FIGURE 2.5. Partial autocorrelations

choice, since it reflects the idea of a decreasing influence of past observations on future ones. Moreover, this has the advantage that the choice of the autoregression parameter k becomes less essential, and also that the value of the parameter γ can in practice be selected together with the value of h by the cross-validation procedure.

Taking a distance of the form (2.12) with weights $\gamma_i = \gamma^i$ for application of the kernel smoothing method for prediction is also possible when high values of autocorrelations are observed at lags $\tau, 2\tau, ...$ for some $\tau > 1$. Then we consider the model

$$y_i = F(y_{i-\tau}, y_{i-2\tau}, ..., y_{i-k\tau}) + \epsilon_i$$

and estimate F by (2.9) with a distance as in (2.12) and weights $\gamma_i = \gamma^i,\ i = 1, ..., k$ for some $\gamma \in (0, 1)$. This is also useful when a time series is oversampled. In that case considering the autoregression on a number of all consecutive previous observations is not necessary and only leads to an oversized model. Then, by choosing a time delay τ and proceeding in the same way as above, we significantly reduce the model and, so, the computation time.

In general, the kernel autoregression estimate (2.9) with a distance as in (2.10) allows for broader flexibility of an assumed autoregression model and of assigning the influence of past observations on future ones.

2.3 Prediction by Neural Networks

Here we shall address briefly another nonlinear method of prediction which has become popular in the past few years: the application of neural networks.

2.3.1 Feedforward Neural Networks

Artificial neural networks (NN) originated as a mathematical model of the functioning of the human brain. Mathematically a neural network represents a directed graph, where the vertices, arranged in layers, are called the *neurons* and the directed edges the *synaptic connections*. Feedforward neural nets are those with connections leading only from neurons on the lth layer to the $l + 1$st layer. Each edge feeds the output of a neuron on the previous layer to the input of a neuron on the next layer, and has a synaptic weight assigned to it. The neurons which receive an input

from outside a network, i.e. those without input connections from other neurons, are called *input neurons*; those without output connections to other neurons are called *output neurons*. The layers in between are called *hidden layers*.

An output of a neuron j is multiplied by the synaptic weight w_{ij}, which corresponds to the connection from the jth to the ith neuron, and the result is received as an input of the ith neuron. Then, all inputs of the ith neuron and some threshold θ_i are summed up, and a function called the *activation* or *transfer function* σ is applied to the sum. This procedure is repeated till the output layer of the network is reached. If the output layer consists of more than 1 neuron, such a network can be considered as a superposition of several networks, so it makes sense to consider only neural networks with one output neuron.

The most commonly used neural networks are those with one hidden layer. A feedforward neural network with k input neurons, 1 hidden layer of n neurons and one output neuron can be viewed as a function $f = f_n : \mathbf{R}^k \longrightarrow \mathbf{R}$, and the total output of such a neural net is given by

$$f_n(\underline{x}) = \sum_{j=1}^{n} v_j \sigma(\sum_{i=1}^{k} (w_{ij} x_i - \theta_j)) = \sum_{j=1}^{n} v_j \sigma(\mathbf{w_j} \cdot \underline{x} - \theta_j), \tag{2.13}$$

where $\underline{x} = (x_1, ..., x_k)$ is the input of a network, w_{ij} are the weights assigned to the edges leading from the input layer to the hidden layer, σ is a transfer function, θ_i are the thresholds, and v_j are the weights from the hidden layer to the output neuron. Usually the sigmoidal function $\sigma(x) = 1/(1 + e^{-x})$ is taken.

It is assumed that the configuration of a neural net (a graph and a number of hidden neurons) and the transfer function are given, and weights and thresholds are adjusted so that the network can perform a given task. Adjustment of weights is called *training*, and it is done by some learning algorithm designed to minimise the mean square error between the desired and the actual output of the network. The most commonly used learning algorithm is the *error backpropagation* (BPL), based on the method of gradient descent.

Due to its rich connection structure, a neural network is supposed to learn how to perform a complex task from the examples, just like the human brain does, instead of being given a large set of rules in advance. The training process involves presenting to a network a set of known examples of inputs and corresponding outputs, and continuously adjusting weights until the network output maximally matches the desired output.

The major areas of application of artificial neural networks include a wide variety of image and pattern classification problems, function estimation and regression problems. In recent years, neural networks have been applied extensively to the prediction of future observations of a time series. Having observed a sufficiently long part of the time series, we can try to find the unknown functional relationship between the past and the future observations (2.1) by training a neural network to approximate the unknown autoregression function F.

The universal approximation capability of a one-hidden-layer feedforward neural network with sigmoidal activation function follows from the results of Cybenko [8]. He showed that the functions

$$\sum_{j=1}^{n} v_j \sigma(\mathbf{w_j} \cdot \underline{x} - \theta_j) \tag{2.14}$$

are dense in $L_2(\mu)$ (where μ is a probability measure concentrated on a bounded subset of $\mathbf{R}^k$), which makes them candidates to approximate any function in $L_2(\mu)$.

While the one-hidden-layer network with backpropagation learning algorithm provides a very powerful approximation tool, the training may be very slow and inefficient, especially for a large network. An important question is how to choose the optimal size and configuration of the network for a specific problem. The speed of approximation depends on the number of hidden neurons. Barron [1] proved that a sufficiently smooth function can be approximated by (2.14) in $L_2(\mu)$ at a rate $O(\frac{1}{\sqrt{n}})$. Hence, a network with a number of hidden neurons that is too small will not be able to reach a given error level, while a network that is too large requires too much training time. Training of a neural network involves minimisation of a function of $n(k+2)+1$ parameters, and so, a large network cannot be trained properly in reality.

The next section will deal with a constructive learning algorithm of Projection Pursuit Learning (PPL), which is inspired by the Projection Pursuit Regression (PPR). This learning algorithm builds a neural network by dynamically adding hidden neurons, in this way optimising the network size and decreasing learning time.

2.3.2 Projection Pursuit Learning

Projection pursuit is a nonparametric regression technique, known from nonparametric statistics, that allows the interpretation of high-dimensional data by considering well-chosen one-dimensional projections.

Let again X, X_i be the k-dimensional vectors of explanatory variables, Y, Y_i be the responses, $i = 1, ..., N$, and the (unknown) functional relation between X and Y is the regression surface r (2.6). In projection pursuit, r is approximated by the sum of empirically determined univariate functions g_j of linear combinations of explanatory variables, i.e.

$$\hat{r}_n(\underline{x}) = \sum_{j=1}^{n} g_j(\mathbf{a}_j^T \cdot \underline{x}), \tag{2.15}$$

where $\underline{x}$ is a vector of observed explanatory variables and $\mathbf{a}_j$ is a unit projection vector. The word "pursuit" refers to finding good projection directions by optimisation. The functions g_j (also called the ridge functions) are then estimated nonparametrically, e.g. by kernel or spline smoothing, or using a supersmoother. For a good review on projection pursuit regression see Huber [15].

Note that the structure of PPR (2.15) is similar to that of a neural network (2.13), where the activation functions g_j in each neuron have to be estimated, instead of being fixed in advance (e.g. the sigmoidal function). The algorithm of projection pursuit learning implements PPR into a one-hidden-layer neural network

$$f_n(\underline{x}) = \sum_{j=1}^{n} v_j g_j(\mathbf{a}_j^T \cdot \underline{x} - \theta_j), \tag{2.16}$$

where the weights $v_j, \mathbf{a}_j, \theta_j$ and the functions g_j are adjusted to minimise the mean-squared error between the network output and the desired output.

The following training scheme is considered: the weights and the ridge function corresponding to one hidden neuron are adjusted until there is no further improvement in error level. Then a new neuron is added and procedure is repeated.

Studies show that in general PPL performs better than BPL for model-free regression problems: PPL requires fewer neurons to achieve comparable accuracy and it is less sensitive to outliers. However, due to additional problem of ridge functions estimation, the total learning time is comparable to that of BPL, or even higher.

The learning time will decrease if the activation functions do not have to be estimated. For instance, we again can take the sigmoidal activation function and for the rest carry out PPL in the same way, i.e. adding the hidden neurons one by one. The following "relaxed" variant of projection pursuit regression does just that and, moreover, achieves the desirable approximation accuracy.

Suppose that f_n has been selected, then f_{n+1} is selected from the restricted class of functions of the form

$$(1-\alpha)f_n(\underline{x}) + \alpha v\sigma(\mathbf{w}_{n+1}\cdot\underline{x} - \theta_{n+1}),$$

where $\alpha, v, \mathbf{w}_{n+1}, \theta_{n+1}$ are the required parameters. Here the parameter α plays the role of a "relaxing" parameter. It allows more "space" for improvements of the approximation when f_{n+1} is being selected. In terms of a neural network this is the following iterative procedure: after having trained a network with n neurons, until some stopping criteria is reached, all the weights are kept fixed, one more neuron is added and the weights corresponding to this neuron are adjusted together with α. When there is no more improvement in terms of approximation error, one more neuron is added, etc., until the desirable error level is achieved.

The so-called "greedy approximation lemma" of Jones [16] establishes convergence of this procedure in $L_2(\mu)$ with the approximation rate of $O(\frac{1}{\sqrt{n}})$.

Adding hidden neurons with the sigmoidal activation function one by one substantially decreases learning time in comparison to BPL, where all weights must be adjusted simultaneously. This is confirmed by the numerical examples of the next section. However, replacing the unknown ridge functions by a single sigmoidal function is too restrictive. A more flexible approach would be to add more than one neuron at a time, in this way increasing the richness of the class of admissible ridge functions. We suggest adding neurons in pairs - two sigmoidal functions of the opposite signs form a unimodal function and can successfully approximate peaks and valleys of an unknown regression function. Comparison to a kernel smoothing methods provides an extra motivation for this suggestion.

In kernel smoothing the multivariate case is treated either by product kernels or by considering a distance between vectors. Instead, we may consider a projection pursuit kernel estimator:

$$\hat{r}_{N,n} = \sum_{j=1}^{n}\sum_{i=1}^{N}\frac{1}{h_j}K\left(\frac{\mathbf{a}_j^T\cdot\underline{x} - \mathbf{a}_j^T\cdot X_i}{h_j}\right)Y_i, \tag{2.17}$$

where $\mathbf{a}_j$ is a unit projection vector, h_j is a bandwidth in direction $\mathbf{a}_j$, and we suppose that the sum has been properly normalised. Here n is the number of hidden neurons and N is the sample size.

Note that we can rewrite an output of a one-hidden-layer neural network (2.13)

as

$$f_n(\underline{x}) = \sum_{j=1}^{n} v_j \sigma\left(\frac{\mathbf{a}_j^T \cdot \underline{x} - t_j}{s_j}\right), \tag{2.18}$$

where $|\mathbf{a}_j| = 1$. The similarity between (2.17) and (2.18) reveals some connections between these two approaches.

If we set $\sigma = K$, then both (2.17) and (2.18) perform data smoothing in each direction $\mathbf{a}_j$. The parameter s_j, which is chosen by the backpropagation algorithm, plays the role of the smoothing parameter h_j, which is chosen in kernel methods by the cross-validation method. The main difference between the two methods is that, while the kernel method performs explicit smoothing in all directions $\mathbf{a}_j$ using all the data, a neural network performs implicit smoothing, determining v_j (instead of Y_i/h_j) and t_j (instead of $\mathbf{a}_j^T \cdot X_i$) by some nonlinear optimisation procedure.

In a neural network the activation function σ is the sigmoidal function and does not have the form of a kernel. However, two sigmoidal function of the opposite signs approximate the typical unimodal bell-shape of a kernel function. Moreover, different bandwidths s_j are selected for each half of such an asymmetric kernel, in this way possibly improving approximation capabilities in comparison with symmetric kernel K with a single value of a bandwidth. These considerations support the suggestion of adding hidden neurons in PPL in pairs.

In the next section we shall apply PPL to predict a real-life time series and compare it with BPL in terms of the prediction error and the learning time.

2.4 Application to a fluidized bed time series

Here we will consider a time series of pressure measurements in a fluidized bed. We used this time series in the introduction as one of our examples of nonlinear time series. Part of the data set together with the delay map at lag 35 is shown again in Fig.7.6 and 7.7. The prediction step for this time series was chosen 35 measurements

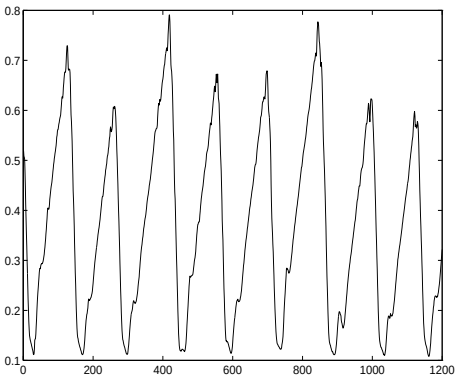


FIGURE 2.6. Pressure fluctuations in the fluidized bed

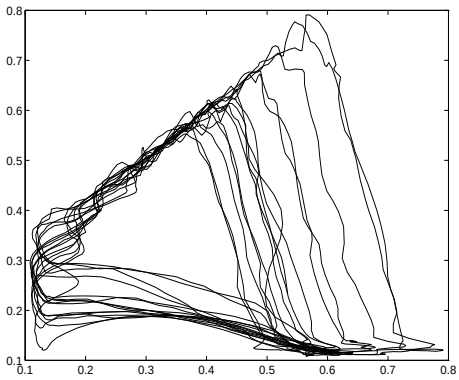


FIGURE 2.7. Delay map for the fluidized bed time series: X_n vs. X_{n+35}

ahead, and, since the time series is oversampled, we introduce a time delay $\tau = 35$

and look for a predictor of the form:

$$\hat{y}_{n+35} = F(y_n, y_{n-\tau}, ..., y_{n-5\tau}). \tag{2.19}$$

We used a larger part of the time series as the basis for predictions, and disjoint smaller parts as test sets. For this time series, as a measure of quality of our predictions, we take the *Average Absolute Error*:

$$AAE_N = \frac{1}{N} \sum_{j=1}^{N} \frac{|y_j^{(t)} - \hat{y}_j^{(t)}|}{R} \cdot 100\%,$$

where R is the range of values of the time series, $y_1^{(t)}, ..., y_N^{(t)}$ is a test set and $\hat{y}_1^{(t)}, ..., \hat{y}_N^{(t)}$ are the obtained predictions.

For application of our kernel method we took the Gaussian kernel function and geometrically decreasing weights $\gamma_i = \gamma^i$ $(i = 1, ..., 5)$ for computation of the distance (2.10). Selection of the bandwidth h and the parameter γ was done by simultaneous cross-validation. The cross-validation surface is shown in Fig.7.8, the optimal pair of parameters being $(h, \gamma) = (0.03, 0.6)$.

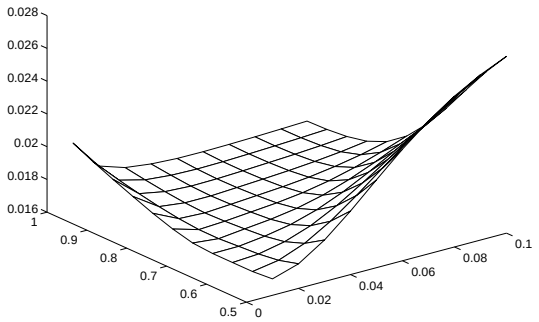
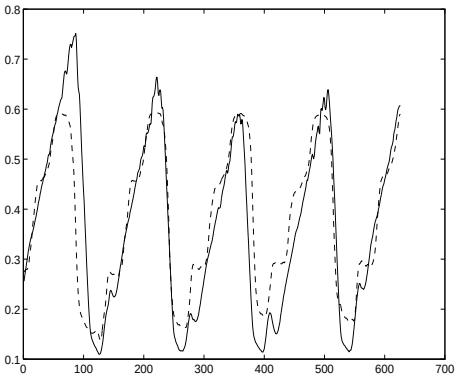


FIGURE 2.8. Cross-Validation Function $CV(h, \gamma)$

Fig.7.9 shows two test sets each consisting of 600 observations (solid line) together with predictions (dashed line). The value of the AAE is 5.75%. In general, the quality of prediction is quite good, and the dashed curve is rather smooth, but, as we could expect, on highs and lows, where the fluctuations are most noticeable, we get slightly worse predictions than on intermediate parts of the time series.

Next we compare the results with the performance of a neural net trained by the method of backpropagation. Looking again for a predictor of the form (2.19), we used the one-hidden-layer feedforward neural net with 5 inputs, 3 hidden neurons and 1 output, which was trained to be the prediction for a value 35 measurements ahead. We used 10^5 training iterations performed on the training set of length 3000 observations. Since we used a small neural net (5:3:1), this number of training iterations should be sufficient to train it properly.

The example of a test set together with neural net predictions is shown in Fig.7.10. The obtained AAE is 8.7%. Comparison of Fig.7.9 with Fig.7.10 shows that the



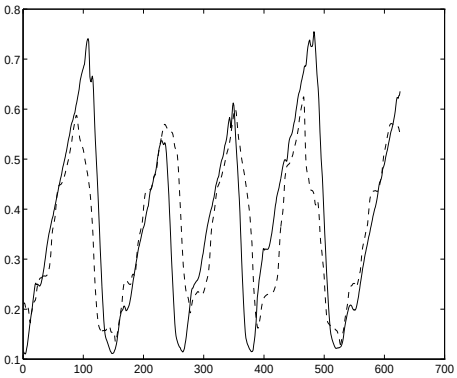


FIGURE 2.9. Fluidized bed t.s. with kernel method predictions

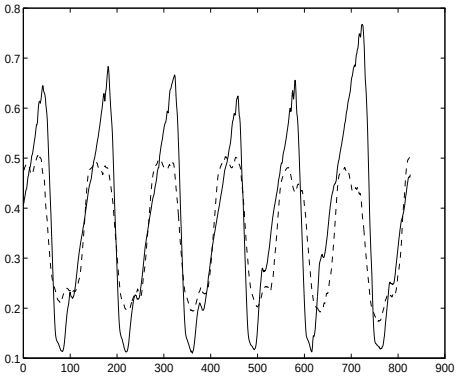


FIGURE 2.10. 5:3:1-NN predictions

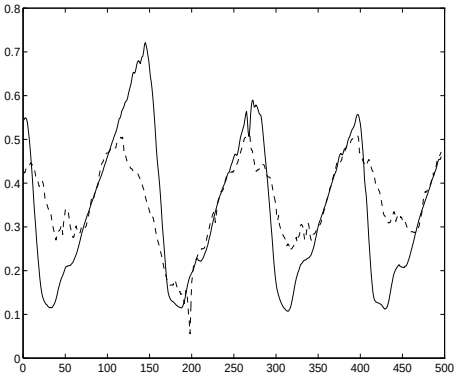


FIGURE 2.11. Local linear predictions

kernel method works better not only in terms of average error, but also has a definite advantage in predicting high and low values where the neural net failed.

We also apply local and global linear predictors to this data set. For the (k, ϵ)-method we take into account ϵ-close vectors of k past values, sampled with time delay 35, and we base the choice of the pair (k, ϵ) on the minimisation of the AAE. The optimal pair is $k = 3$, $\epsilon = 0.2$. A test set together with local linear predictions is shown in Fig.7.11. The AAE is in this case 8.4%, but note that again the predictions of high and low values are of lower quality.

Note, however, that the predictions obtained by the (k, ϵ)-method are rather undersmoothed (Fig. 7.11), and the method has a very local character. This is different from the smooth prediction curves obtained by the kernel method, where the parameters were tuned so that no under and oversmoothing occurs, and those obtained by the neural net, which is a more global approximation procedure.

For comparison we also fitted global linear autoregression of the past 5 values sampled with time delay $\tau = 35$ (as in (2.19), with F linear). This delivered the AAE of 13%, much higher than in applications of nonlinear and locally linear methods. This shows that strong nonlinear dependence in this data set is indeed best captured by applying nonlinear methods of predictions, and that linear methods perform quite poorly in this case.

2.5 References

[1] Barron, A.R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inform. Theory* **39**, No.3, 930-945.

[2] Bosq, D. (1995). *Nonparametric Statistics for Stochastic Processes.* In: *Lecture Notes in Statistics*, **110**. Springer.

[3] Brock, W.A., Dechert, W.A. (1988). Theorems on distinguishing deterministic from random systems. In: *Dynamic Economic Modelling*, Proc. of the 3d Int. Symp. on Economic Theory and Econometrics, Cambridge University Press.

[4] Brockwell, P.J., Davis, R.A. (1991). *Time Series: Theory and Methods.* Springer-Verlag.

[5] Casdagli, M. (1991) Chaos and Deterministic *vs.* Stochastic Nonlinear Modelling. *J. R. Statist. Soc. B*, **54**, No.2, 303-328.

[6] Chen, B., Tong, H. (1993). Nonparametric function estimation in noisy chaos. In: *Developments in time series analysis,* 183-206, Chapman & Hall, London.

[7] Cutler, C.D. (1991). Some results on the behaviour and estimation of fractal dimension of distributions on attractors. *J. Stat. Phys.* **62**, 651-708.

[8] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems* **2**, No.4, 303-314.

[9] Diks, C. (1999). Nonlinear time series analysis. Methods and applications. *Nonlinear Time Series and Chaos*, 4. World Scientific Publishing Co., Inc., River Edge, NJ.

[10] Grassberger, P., Procaccia, I. (1983). Characterization of strange attractors. *Phys. Rev. Lett.* **50**, 346-349.

[11] Gyorfi, L., Haerdle, W., Sarda, P., Vieu, P. (1989).*Nonparametric Curve Estimation from Time Series.* In: *Lecture Notes in Statistics*, **60**, Springer-Verlag.

[12] Haerdle, W. (1990). *Applied nonparametric regression.* In: *Econometric Society Monographs*, **19**, Cambridge University Press.

[13] Haerdle, W., Vieu, P. (1992). Kernel regression smoothing of time series. *J. Time Ser. Anal.* **13**, No.3, 209-232.

[14] Hill, B.M. (1975). A simple general approach to inference about the tail of a distribution. *Ann. Statist.* **3**, No.5, 1163-1174.

[15] Huber, P.J. (1985). Projection pursuit. With discussion. *Ann. Statist.* **13**, No.2, 435-525.

[16] Jones, Lee K. (1992). A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *Ann. Statist.* **20**, No.1, 608-613.

[17] Krylov, N., Bogolubov, N.(1937) *Ann. Math.* **38**, 65-113.

[18] Maechler, M., Martin, D., Schimert, J., Csoppenszky, M., Hwang, J.N. (1990). Projection pursuit learining networks for regression. In: *Proc. 2nd Int. IEEE Conf. Tools Artificial Intell.*, 350-358, Washington D.C.

[19] Milnor, J. (1985). On the concept of attractor. *Comm. Math. Phys.* **99**, 177-195.

[20] Nadaraya, E.A. (1964). On estimating regression. Theory Prob. Appl. **10**, 186-190.

[21] Osborne, A.R., Provenzale, A. (1989). Finite correlation dimension and generalized spectrum for dimensions. *Physica D* **35**, 357-381.

[22] Oseledec, V.I. (1968). A multiplicative ergodic theorem. Characteristic Lyapunov exponents of dynamical systems. (Russian) *Trudy Mosk. Mat. Obsc.* **19**, 179-210.

[23] Ruelle, D. (1989). *Chaotic evoilution and Strange Attractors.* Cambrige University Press.

[24] Ruelle, D. (1976). A measure associated with axiom-A attractors. *Amer. J. Math.* **98**, No.3, 619–654.

[25] Ruelle, D., Takens, F. (1971). On the nature of turbulence. *Comm. Math. Phys.* **20**, 167-192.

[26] Sauer, T., Yorke, J.A., Casdagli, M. (1991). Embedology. *J. Stat. Phys.* **65**, 579-616.

[27] Sugihara, G. (1994). Nonlinear forecasting for the classification of natiral time series. *Phil. Trans. R. Soc. Lond. A* **348**, 477-495.

[28] Takens, F. (1981). Detecting strange attractors in turbulence. *Dynamical systems and turbulence. Lecture Notes in Mathematics*, **898**, 336-381. Springer-Verlag.

[29] Takens, F. (1985). On the numerical determination of the dimension of the attractor. In: *Dynamical Systems and Bifurcations. Lecture Notes in Mathematics*, **1125**, 99-106. Springer-Verlag.

[30] Takens, F. (1991). Detecting Nonlinearities in Stationary Time Series. In: *Int. J. Bifrucation and Chaos* **3**, No.2, 241-256.

[31] Takens, F. (1996). Estimation of dimension and order of time series. In: *Nonlinear Dynamical Systems and Chaos. PNLDE*, **19**, 405-422. Ed. H.Broer, S.A.van Gils, I.Hoveijn, F.Takens. Birkhauser.