Chapter 4
Determinism and predictability

In this chapter we will discuss the notion of the predictability of a system evolving over time or, strictly speaking, of a signal emitted by such a system. Forecasting future values of some quantity is a classical problem in time series analysis but the conceptual importance of the prediction problem is not limited to those who want to get rich by knowing tomorrow's exchange rates. Even if, instead, you are interested in describing, understanding or classifying signals, stay with us for a few pages.

In this book we are concerned with the detection and quantification of possibly complicated structures in a signal. We want to be able to convince others that the structures we find are real and not just fluctuations. The most convincing argument for the presence of some pattern is if it can be used to give an improved prediction. It is a necessary condition for a theory to be compatible with the known data but it is not sufficient. In order to become accepted, a theory must successfully predict something which can be verified subsequently. In time series analysis, we can take this requirement of predictive quality quite literally.

Most concepts, which we will introduce later in order to describe time series data, can be interpreted to some extent as indirect measures of predictability. Due to their indirect nature, some conclusions will remain controversial, especially if the structures are rather faint. The statistically significant ability to predict the signal better than other techniques do will then be a more convincing affirmation of nonlinear and deterministic structure than several dubious digits of the fractal dimension.

Readers whose primary interest is forecasting should see this chapter only as an introduction to concepts that are further elaborated in the chapter on modelling and forecasting, Chapter 12.

4.1 Sources of predictability

A signal which does not change is trivial to predict: the last observation is a perfect forecast for the next one. Even if the signal changes, this can be a reasonable
4.1. Sources of predictability

The predictability of a system evolving over time involves a method of forecasting,\(^1\) and a signal for which this holds is called *persistent*. A system which changes periodically over time is also easy once you have observed one full cycle. Independent random numbers are easy as well: you do not have to work hard since working hard does not help anyway. The best prediction is just the mean value. Interesting signals are something in between; they are not periodic but they contain some kind of structure which can be exploited to obtain better predictions.

Before we mention the most common structures which can be exploited for predictions, let us state how we quantify the success of the predictions we make. The error measure that is most commonly used is the *root mean squared (rms) prediction error*. If we predict the outcome of measurements at times \(n\) to be \(\hat{s}_n\), while the actual measurements are \(s_n\), then the rms prediction error is

\[
e = \sqrt{\langle (\hat{s}_n - s_n)^2 \rangle},
\]

(4.1)

where \(\langle \cdot \rangle\) denotes the average over all the trials we have made. You can easily verify that for independent random numbers this error is minimised by \(\hat{s}_n = \text{const.} = (s_n)\). Other error measures are the *mean absolute error* \(\langle |\hat{s}_n - s_n| \rangle\), or the *logarithmic error* \(\langle \ln |\hat{s}_n - s_n| \rangle\). Error or cost functions can be much more complicated, in particular when money is involved. If you travelled to a country where credit cards are unknown and predicted the necessary amount of cash $100 short, then you would be in trouble: you would have to cable your bank for more cash, you would miss your flight home and end up in prison before the money was there. $100 too much in your pocket are harmless, the maximal cost being $100 if you lose it.

If for some reason you use a fancy cost function, the mean is not necessarily the best prediction even for independent quantities. Predictability is increased by knowledge of the distribution of the values that the predicted quantity takes.

The mean and the probability distribution are static characteristics of the data and do not involve any correlations in time. Depending on the strength and type of correlations, we can improve predictions considerably.

The most common and best-studied sources of predictability are linear correlations in time. Assume a fully random process with a slowly decaying autocorrelation function. The randomness prevents us from making any precise prediction, even with an absolute knowledge of the present state. The strong correlation assures us that the next observation will be given by a linear combination of the preceding observations, with an uncertainty of zero mean,

\[
\hat{s}_{n+1} = \bar{s} + \sum_{j=1}^{m} a_j (s_{n-j} - \bar{s}),
\]

(4.2)

\(^1\) It was claimed that weather forecasts were not much better than this until recently.
where the mean value $\bar{s}$ of the time series is conveniently split off. As we depicted in Section 2.3, the autocorrelation function and the power spectrum are intimately related by the Wiener–Khinchin theorem (see Section 2.3), which are prominent objects of linear statistical inference. Their information can be exhaustively exploited by linear predictors. However, due to the assumed random nature of the data, the underlying auto-regressive models and moving average models are stochastic.

Another kind of temporal correlation is present in nonlinear deterministic dynamical systems. Recall that pure dynamical systems are described either by discrete time maps,

$$x_{n+1} = F(x_n), \quad n \in \mathbb{Z}, \quad (4.3)$$

or by first-order ordinary differential equations,

$$\frac{d}{dt} x(t) = f(x(t)), \quad t \in \mathbb{R}. \quad (4.4)$$

Both variants are mathematical descriptions of the fact that all future states of such a system are unambiguously determined by specifying its present state at some time $n$ (respectively $t$). This implies that there also exists a deterministic forecasting function. However, as we will see later, any inaccuracy in our knowledge of the present state will also evolve over time and, in the case of a chaotic system, will grow exponentially. In the latter case, we can no longer use the time evolution to calculate states arbitrarily far in the future since we can never know a physical quantity with zero uncertainty. However, even for a chaotic system the uncertainty is amplified only at a finite rate, and we can still hope to make reasonable short-term forecasts exploiting the deterministic evolution law. In contrast to correlations reflected by the autocorrelation function, those imposed by nonlinear deterministic dynamics may be visible only by using nonlinear statistics. They are usually called nonlinear correlations, and one has to employ new techniques in order to exploit them for predictions. There are of course lots of data which are neither deterministic chaos nor linear stochastic processes. They do not follow any of these two paradigms, and only for certain cases we will be able to do anything with them. See the discussion in Section 12.1.

### 4.2 Simple nonlinear prediction algorithm

Let us now be more specific and develop a simple prediction algorithm which exploits deterministic structures in the signal. As with most algorithms in this field, we will proceed by assuming that the data originates in an underlying dynamical system. Once we have established the algorithm, we will have to study its behaviour in more realistic cases.
4.2. Simple nonlinear prediction algorithm

A deterministic vector valued data set sampled at discrete times in its state space is perfectly described by Eq. (4.3). Unfortunately, only if we know the mapping $F$ we can use this as a prediction scheme. Knowing $F$ is an unrealistic expectation when working with real world data. With $F$ unknown, we have to make some assumptions about its properties. Using the minimal assumption that the mapping $F$ is continuous\footnote{We can usually allow the map to be composed of a few continuous pieces.} we can construct a very simple prediction scheme. In order to predict the future state $\mathbf{x}_{N+1}$, given the present one $\mathbf{x}_N$, we search a list of all past states $\mathbf{x}_n$ with $n < N$ for the one closest to $\mathbf{x}_N$ with respect to some norm. If the state at time $n_0$ was similar to the present (and thus close in phase space), continuity of $F$ guarantees that $\mathbf{x}_{n_0+1}$ will also be close to $\mathbf{x}_{N+1}$. Since by assumption $n_0 < N$, also $n_0 + 1 \leq N$, and hence we can read $\mathbf{x}_{n_0+1}$ from the data record.

If we have observed the system for a very long time, there will be states in the past which are arbitrarily close to the present state, and our prediction, $\mathbf{x}_{N+1} = \mathbf{x}_{n_0+1}$, will be arbitrarily close to the truth. Thus, in theory we have a very nice prediction algorithm. It is usually referred to as the “Lorenz’s method of analogues” because it was proposed as a forecasting method by Lorenz (1969).

Let us now face reality one step at a time. Even if the assumption of an underlying deterministic system is correct, we usually do not measure the actual states $\mathbf{x}_n$. Instead, we observe one (or a few) quantities which functionally depend on these states. Most commonly we have scalar measurements

$$ s_n = s(\mathbf{x}_n), \quad n = 1, \ldots, N, $$

in which, more often than not, the measurement function $s$ is as unknown as $F$. Obviously we cannot invert $s$ but we can use a delay reconstruction (see Section 3.2) to obtain vectors equivalent to the original ones:

$$ \mathbf{s}_n = (s_{n-(m-1)\tau}, s_{n-(m-2)\tau}, \ldots, s_{n-\tau}, s_n). $$

This procedure will introduce two adjustable parameters into the prediction method (which is in principle parameter free): the delay time $\tau$ and the embedding dimension $m$. The resulting method is still very simple. For all measurements $s_1, \ldots, s_N$ available so far, construct the corresponding delay vectors $\mathbf{s}_{(m-1)\tau+1}, \ldots, \mathbf{s}_N$. In order to predict a future measurement $s_{N+\Delta n}$, find the embedding vector $\mathbf{s}_{n_0}$ closest to $\mathbf{s}_N$ and use $\mathbf{s}_{n_0+\Delta n}$ as a predictor. This scheme has been used in tests for determinism by Kennel & Isabelle (1992), for example. To our knowledge, the idea goes back to Pikovsky (1986). The method used by Sugihara & May (1990) is closely related.

As a next step towards reality, we no longer ignore the fact that every measurement of a continuous quantity is only valid up to some finite resolution. This resolution is
limited mainly by fluctuations in the measurement equipment, which are hopefully random, and by the fact that the results are eventually stored in some discretised form. In any case the finite resolution, call its typical size \( \sigma \), implies that looking for the single closest state in the past is no longer the best we can do since inter-point distances are contaminated with an uncertainty of the order of \( \sigma \). All points within a region in phase space of radius \( \sigma \) have to be considered to be equally good predictors \textit{a priori}. Instead of choosing one of them arbitrarily, we propose to take the arithmetic mean of the individual predictions based on these values. The estimated resolution of the measurements introduces the third parameter into our prediction scheme.

Now we are ready to propose the following prediction algorithm. Given a scalar time series \( s_1, \ldots, s_N \), choose a delay time \( \tau \) and an embedding dimension \( m \) and form delay vectors \( s_{(m-1)\tau+1}, \ldots, s_N \) in \( \mathbb{R}^m \). In order to predict a time \( \Delta n \) ahead of \( N \), choose the parameter \( \epsilon \) of the order of the resolution of the measurements and form a neighbourhood \( \mathcal{U}_\epsilon(s_N) \) of radius \( \epsilon \) around the point \( s_N \). For all points \( s_n \in \mathcal{U}_\epsilon(s_N) \), that is, all points closer than \( \epsilon \) to \( s_N \), look up the individual "predictions" \( s_{n+\Delta n} \). The prediction is then the average of all these individual predictions:

\[
\hat{s}_{n+\Delta n} = \frac{1}{|\mathcal{U}_\epsilon(s_N)|} \sum_{s_n \in \mathcal{U}_\epsilon(s_N)} s_{n+\Delta n}.
\]

(4.7)

Here \( |\mathcal{U}_\epsilon(s_N)| \) denotes the number of elements of the neighbourhood \( \mathcal{U}_\epsilon(s_N) \). In the case where we do not find any neighbours closer than \( \epsilon \) we might just increase the value of \( \epsilon \) until we find some. A simple but effective implementation of this scheme is the program \texttt{zerot} included in the TISEAN package and discussed in Section A.4.1. Later in this book we will describe more refined ways of prediction and modelling. In this more general setting, it will turn out that the scheme described here is based on a zeroth-order approximation of the dynamics, i.e. it is a locally constant predictor as opposed to a locally linear or even a global nonlinear model. Even more interestingly, it will be shown that for certain nonlinear stochastic processes, namely Markov chains, this is also the optimal predictor.

If more than a few predictions are needed, for instance because the average prediction error needs to be determined accurately, this implementation is very inefficient. For each prediction, \textit{all} vectors in the embedding space are considered (excluding only those in the temporal vicinity of the one to be predicted in order to avoid causality conflicts). They are tested for closeness, and most of them are rejected as too far away. It is therefore wise to use an efficient method to find nearest neighbours in phase space, see Section A.1.1.

3 For simplicity, we propose the use of the maximum norm: A point belongs to \( \mathcal{U}_\epsilon(s_N) \) if none of its coordinates differs by more than \( \epsilon \) from the corresponding coordinate of \( s_N \).
4.3 Verification of successful prediction

If you made a prediction which came true, your success might have been just a coincidence. For significance, you have to study the errors of many predictions starting from independent actual states. This approach can be very tedious. Imagine that you can predict on 1 November whether there will be snow on Christmas Eve (let us say in Copenhagen – it is too easy to predict in Montreal). Only after a couple of years will it be statistically significant whether your prediction is useful. In practice, you would prefer to test your method on a record of the last one hundred years or so, but there is a problem: the “future” events you are predicting are already known, and you will have to make sure you do not use any existing information on the outcomes you want to predict. If you use the past to optimise your prediction method, you will eventually account for all the funny coincidences that happened over the last century. However, it is no longer appropriate to quote the performance of the predictor on this record as the prediction error. Such an error would be called an in-sample prediction error. If somebody tries to convince you of something by giving an impressive prediction error, always make sure it is an out-of-sample prediction error! (Otherwise one should rather speak of a “post-diction”.)

Out-of-sample error means that the given data set is divided into a training set and a test set. The parameters of the predictor (e.g., embedding dimension, time lag, neighbourhood size in Eq. (4.7)), are optimised by minimisation of the prediction error on the training set. The number which you use to access its performance should then be the prediction error on the test set. The predictor in Eq. (4.7) contains one ambiguity with this respect: Should the neighbourhood $U_e(s_n)$ be taken from the training set or from the test set? The concept of causality allows us to search for neighbours in the entire history of the current state $s_N$, ignoring the splitting into training and test set. However, history means that also the “futures” of the neighbours, the elements $s_{n+\Delta n}$, have to be in the past of the current state, i.e., $n + \Delta n < N$. This is important for long term predictions of highly sampled data with large $\Delta n$.

The fact that in-sample errors often are much smaller than out-of-sample errors is more relevant the more parameters are adjusted to optimise the performance on the training set. The simple nonlinear predictor, Eq. (4.7), can almost be called a parameter free model, the few parameters being fixed initially. Nevertheless we will quote out-of-sample errors in the examples. They have been estimated from data which have been withheld while building the predictor. A notable exception where the use of in-sample prediction errors may be justified are tests for nonlinearity where the errors are not evaluated on their own, but are compared to surrogate data sets. A more thorough discussion on the statistical relevance of quantities such as prediction errors will be taken up in Chapter 7.
4. Determinism and predictability

Figure 4.1 As an example for a simple nonlinear prediction, an enlarged region of a phase portrait of the NMR laser data is shown. (See Fig. 3.6 for a full portrait.) For an exemplary phase space point we highlighted all its neighbours closer than $e = 50$ (the cluster of squares in the lower part) and their images after two time steps. The cluster of the images is more spread out, but the centre of mass still yields a reasonable prediction.

Often, the data base is too small to be split into training and test sets. Then, a variant of this, called leave-one-out statistics, or complete cross-validation, is employed: For every single prediction, the predictor uses information from the full data set, excluding only the information which is temporally correlated to the actual prediction to be made. In the simple predictor Eq. (4.7) this means that those delay vectors whose time indices obey $|n - N| < n_{\text{corr}}$ are excluded from the neighbourhood $\mathcal{U}_c(s_N)$, where $n_{\text{corr}}$ has to be fixed in accordance with the prediction horizon $\Delta n$ and the correlations in the data set.

Example 4.1 (Prediction of NMR laser data). Let us test the proposed algorithm on an experimental but highly deterministic data set. In the introduction, we mentioned the NMR laser experiment carried out at the Institute for Physics at the University of Zürich. The data set we used to produce Fig. 1.2 consisted of 38 000 points, exactly one point per driving cycle (see Appendix B.2 for more information). We split the data set into two parts. The first 37 900 points form the data base for the prediction algorithm and the last 100 points form the test set. On each of the latter, we performed predictions for $\Delta n = 1, \ldots, 35$ time steps ahead. For each prediction horizon $\Delta n$, we computed the root mean squared prediction error $e_{\Delta n} = \sqrt{\langle (s_{n+\Delta n} - \bar{s}_{n+\Delta n})^2 \rangle}$, which measures the average deviation of the predictions from the subsequently measured actual values. This prediction error is shown as diamonds in Fig. 1.2 (see also Fig. 4.1). We used the TISEAN
4.3 Verification of successful prediction

Figure 4.2 Prediction errors versus prediction time using a linear (diamonds) and the simple nonlinear (crosses) predictor for the human breath rate data from Example 1.2. For an explanation see Example 4.2.

program zeroth (see Appendix A.4.1) with embedding dimension $m = 3$, delay time $\tau = 1$ and neighbourhood size $\epsilon = 50$, which is about twice the amplitude of the measurement error in this data set. Note that for the small amplitude regime, the increase in the error is well described by an exponential $e_{\Delta n} \approx e^{0.3\Delta n}$, which suggests that the prediction error is dominated by the chaotic divergence of initially close trajectories. Later we will (in Chapter 5) estimate the maximal Lyapunov exponent of this data set to be $\lambda = 0.3$. However, we do not recommend the use of the increase in prediction error as a method for estimating the Lyapunov exponent since the averaging is done in the wrong way.

Example 4.2 (Prediction of human breath rate). We can also try to predict data which are not as clearly deterministic as the NMR laser data in Example 4.1. However, we must be careful when it comes to interpreting the results. If linear correlations are present in the data, the nonlinear predictor will pick up this information to some extent. Thus, if an algorithm based on chaos theory is able to predict better than chance, this alone cannot be taken as a signature of deterministic chaos in the data. In Chapter 7, we will discuss how to make more significant statements in this respect. Now let us try to forecast future values of the human breath rate data set mentioned in the introduction (Example 1.2, Fig. 1.4; see also Appendix B.5). Again we split the data set into two parts, this time with a database containing 1500 points. For each prediction horizon, the prediction error is estimated by an average of 100 trials. Figure 4.2 shows the root mean squared (rms) prediction error for the best linear predictor we could find (diamonds), and the simple nonlinear predictor introduced in this section (crosses). The linear predictor has five coefficients, the nonlinear one uses neighbourhoods of $\epsilon = 1000$ in two
4. Determinism and predictability

dimensions. The results for both does not depend much on the parameters chosen, and consequently no systematic optimisation of these parameters was performed. Thus, the prediction errors can be regarded as out-of-sample errors. We observe that the nonlinear method is able to make slightly better prediction over short times (up to three time units), while the linear method is marginally better from then on. Note the much higher relative prediction error (that is, error as a fraction of the standard deviation of the signal) than in the NMR laser example. There is a much stronger effectively random component in this data set. We will revisit this example in Chapter 7, where we will investigate it with respect to possible nonlinear structures.

The quality of a nonlinear prediction can only be assessed by comparison with some benchmark given by standard techniques. Let us draw your attention to a common way of presenting the results of prediction which we consider not to be very instructive and sometimes misleading. You will often see plots showing a time series together with one- (or a few) step-ahead predictions. If the time series is autocorrelated (as is typical for a continuous time process), it is not particularly hard to predict a few steps ahead. As an extreme example, it is easy to predict the weather ten minutes from now! In analogy to this, we are not surprised that locally linear predictions in Abarbanel (1996) of the volume of the Great Salt Lake in Utah appear to be quite convincing. However, in this case we could verify that the prediction horizon compared to the autocorrelation time is so short that the one-step prediction error can easily be beaten by the linear model $s_{n+1} = 2s_n - s_{n-1}$, which just extrapolates the trend of the last two measurements.

More sophisticated prediction schemes and many issues related to modelling and prediction will be discussed in Chapter 12.

4.4 Cross-prediction errors: probing stationarity

It turns out that the locally constant prediction scheme described in Section 4.2 is one of the most robust indicators of nonlinear structure in a time series and is particularly sensitive to deterministic behaviour. We will thus use it later in Section 7.1 as a discriminating statistic in a test for nonlinearity. Indeed, this simple scheme works with much less data than, e.g., local linear predictors (see Section 12.3.2) and, unlike global nonlinear fits (see Section 12.4), no smart choice of basis functions for each new data set is necessary. These properties make the algorithm well suitable as a crude, but nevertheless sensitive, probe whenever we are looking for (nonlinear) predictability. As an example of the many situations where such a probe can be useful, let us use it to study non-stationarity in a long data set.
4.4. Cross-prediction errors: probing stationarity

The idea is that, in a system where different dynamics are at work at different times, data records from different episodes are not equally useful as data bases for predictions. Let us break the available data into $N_i$ segments $S_i$, $i = 1, \ldots, I$, such that each segment is just long enough to make simple nonlinear predictions, say $N_i = N/I \approx 500$. Now for each two segments $S_i$ and $S_j$, compute the rms prediction error using $S_i$ as the training set (i.e. find neighbours in $S_i$) but use $S_j$ as the test set (i.e. on which predictions are performed). The cross-prediction error as a function of $i$ and $j$ then reveals which segments differ in their dynamics. If a prediction error for a couple of segments $S_i$, $S_j$ is larger than the average, the data in $S_i$ obviously provide a bad model for the data in $S_j$. We expect the diagonal entries $i = j$ to be systematically smaller: they represent in-sample errors since training set and test set are identical. The use of mutual predictions as a test for stationarity is developed in Schreiber (1997). The TISEAN package has a program called nstatz that does most of the work.

Example 4.3 (Non-stationarity of breath rate). The data set introduced in Appendix B.5 contains a recording of the breath rate of a human patient during almost a whole night (about 5 hr). Of course, conditions have probably changed during that time. As a first approach to the question of which episodes of the signal show similar dynamics, we split the recording into 34 non-overlapping segments $S_i$ of 1000 points (500 s) each. Now we use every segment in turn as a data base for predictions with zeroth. We use unit delay and three dimensional embeddings. The size of the neighbourhoods is set to one quarter of the variance of the values in the data base. For each of the 34 data bases we compute 34 prediction errors, predicting values in segment $S_j$. In Fig. 4.3 we show the rms prediction error as a surface over the $i-j$-plane. Except for the expected lower errors for $i = j$, we see that there is a transition around the middle of the recording: segments from the first half are useless for predictions in the second half, and vice versa. □

Figure 4.3 Mutual prediction errors for segments of a long, non-stationary recording of the breath rate of a human. See text of Example 4.3.
4. Determinism and predictability

The idea of breaking a data set into parts and considering the fluctuations of a certain statistic computed on the single parts is widespread, and the only question is what statistic to use. We have chosen nonlinear predictability, since it is very appropriate in the case where data possess strong nonlinear correlations, and it yields meaningful results even for data from stochastic processes. The main advantage of our choice, however, lies in the fact that it allows the computation of cross-correlations as in Fig. 4.3. The variations of the cross-prediction errors are much larger and hence more significant than the fluctuations of the in-sample errors, which appear on the diagonal in Fig. 4.3. We will come back to this in Chapter 13.

4.5 Simple nonlinear noise reduction

A close relative of forecasting is noise reduction. Rather than predicting future values, we now want to replace noisy measurements by better values which contain less noise. For prediction, we had no information about the quantity to be forecasted other than the preceding measurements. For noise reduction, we are better off in this respect. We have a noisy measurement to start with, and, unless we have to process data items immediately in real time, we also have the future values. On the other hand this provides a benchmark: the replacements we propose have to contain errors which are on average less than the initial amplitude of the noise, otherwise the whole procedure is worthless.

Noise reduction means that one tries to decompose every single time series value into two components, one of which supposedly contains the signal and the other one contains random fluctuations. Thus, we always assume that the data can be thought of as an additive superposition of two different components which have to be distinguishable by some objective criterion. The classical statistical tool for obtaining this distinction is the power spectrum. Random noise has a flat, or at least a broad spectrum, whereas periodic or quasi-periodic signals have sharp spectral lines. After both components have been identified in the spectrum, a Wiener filter (Section 2.4) can be used to separate the time series accordingly.

This approach fails for deterministic chaotic dynamics because the output of such systems usually leads to broad band spectra itself and thus possesses spectral properties generally attributed to random noise. Even if parts of the spectrum can be clearly associated with the signal, a separation into signal and noise fails for most parts of the frequency domain. Chaotic deterministic systems are of particular interest because the determinism yields an alternative criterion to distinguish the signal and the noise (which is, of course, not deterministic).

The way we will exploit the deterministic structure will closely follow what we have done for nonlinear predictions. (Again we have to refer the reader who needs
4.5. Simple nonlinear noise reduction

Considering the fluctuations of a widespread, and the only questionable linear predictability, since it allows the computation of the cross-prediction errors of the in-sample fluctuations of the in-sample We will come back to this in

Simple noise reduction

Rather than predicting future values by better values which contain exactly the quantity to be forecasted with, and, unless we have to have the future values. On the facts we propose have to contain the magnitude of the noise, otherwise every single time series value contains the signal and the other piece of the data we assume that the data can be decomposed into components which have the classical statistical tool for noise filtering, a Wiener filter and for the spectrum, a Wiener filter takes this into account.

Economics because the output of the filter and thus possesses spectral properties if parts of the spectrum can go to signal and noise fail for logistic systems are of particular importance. The criterion to distinguish the

will closely follow what we refer the reader who needs

optimal results to a later chapter, Chapter 10 in Part Two.) Let us, for convenience, state the problem in delay coordinates right away. Let the time evolution of the signal be deterministic with a map \( f \) which is not known to us. All that we have knowledge of are noisy measurements of this signal:

\[
s_n = x_n + \eta_n, \quad x_n = f(x_{n-m}, \ldots, x_{n-1}).
\]

(4.8)

Here \( \eta_n \) is supposed to be random noise with at least fast decaying autocorrelations and no correlations with the signal \( x_n \). We will refer to \( \sigma = \sqrt{\langle \eta^2 \rangle} \) as the noise amplitude or absolute noise level.

How can we recover the signal \( x_n \) given only the noisy sequence \( s_n \)? After what we have learned about predictions, at a first glance the solution seems obvious: in order to clean a particular value of the time series, throw away the measurements and replace them by a prediction \( \tilde{x}_n \) based on previous measurements, say

\[
\tilde{x}_n = \tilde{f}(s_{n-m}, \ldots, s_{n-1}).
\]

(4.9)

If you are a more conservative character, you may prefer to use some linear combination of the noisy measurements and their predictions. But even then, this scheme will fail for chaotic systems. Remember that the previous measurements are noisy as well, and the errors they contain will be amplified by the chaotic dynamics. The proposed replacement will on average be much worse than the noisy measurements we had. (How much worse depends on the average local expansion rate of the dynamics — expect a factor of 1.5 — 3.0 for typical maps! See Section 10.2 for a more quantitative discussion.)

The next, smarter, suggestion would be to reverse time and “predict” the desired value from the subsequent measurements. The idea is that a reversed expansion is after all a contraction. Here, the catch is that, in bounded systems, not all errors can be expanded in all directions, and those directions in which errors are contracted become expanding under time reversal. Thus a version of Eq. (4.9), solved for \( \tilde{x}_{n-m} = s_{n-m} \), will fail again since the expansion is the dominant effect.

What finally works is to solve the implicit relation

\[
x_n - f(x_{n-m}, \ldots, x_{n-1}) = 0
\]

(4.10)

for one of the coordinates in the middle, say \( x_{n-m/2} \), for even \( n \). This stabilises the replacement in both the stable and the unstable directions. This is illustrated in Fig. 4.4. Of course we do not know \( f \). Even if we did, we could not in general solve it for one of its arguments. Therefore, we choose a very crude approximation for now and replace the function \( f \) (solved for one argument) by a locally constant function. This is exactly what we did for the simple nonlinear prediction method given above. In other words, in order to obtain an estimate \( \tilde{x}_{n-m/2} \) for the value of \( x_{n-m/2} \), we form delay vectors \( s_n = (s_{n-m+1}, \ldots, s_n) \) and determine those
which are close to \( s_{n_0} \). The average value of \( s_{n-m/2} \) is then used as a cleaned value \( \hat{s}_{n_0-m/2} \):

\[
\hat{s}_{n_0-m/2} = \frac{1}{|\mathcal{U}_c(s_{n_0})|} \sum_{s_{n_0} \in \mathcal{U}_c(s_{n_0})} s_{n-m/2} .
\] (4.11)

Again, for numerical case we use the maximum norm. Only when the embedding dimension is really large (say, \( m > 20 \)), and we want to treat data with a high noise amplitude, then a Euclidean norm is more useful for the identification of those neighbours in the noisy data set that would also be good neighbours for the noise free data. Due to the assumed independence of signal and noise,

\[
(s_{n} - s_{k})^2 = \sum_{l=0}^{m-1} (s_{n+l} - s_{k+l})^2 = \sum_{l=0}^{m-1} (\alpha_{n+l} + \eta_{n+l} - \alpha_{k+l} - \eta_{k+l})^2 \\
\approx (\alpha_{n} - \alpha_{k})^2 + 2m\sigma^2 ,
\]

where the approximation becomes the better the larger \( m \). Hence, the squared Euclidean distance of two noisy delay vectors is identical to the squared Euclidean
distance of the noise free vectors plus a constant. The noisy vectors closest to \( s_n \) are therefore in good approximation exactly those whose noise free counterparts would also be the closest vectors to the noise free version of \( s_n \), and these are the ones we need for successful noise reduction.

Not surprisingly, Eq. (4.11) is very similar to Eq. (4.7). As opposed to the prediction case, here \( \mathcal{U}_\epsilon(s_{n_0}) \) is never empty, no matter how small a value of \( \epsilon \) we choose: it always contains at least \( s_{n_0} \). This is good to know since we have to make some choice of \( \epsilon \) when we use this algorithm. It is guaranteed that if we choose \( \epsilon \) too small, the worst thing that can happen is that the only neighbour found is \( s_{n_0} \) itself. This, however, yields the estimate \( \hat{s}_{n_0-m/2} = s_{n_0-m/2} \), which just means that no correction is made at all.

We have just embarked on an important discussion. How can we make sure that all we do is to reduce noise without distorting the signal? Since we do not know the clean signal, all we can ask for in practice is that the effect of any possible distortion is more acceptable than the effect of the noise before. In most cases, this will mean that the error due to distortion must be considerably smaller than the noise amplitude.

Obviously, it will be necessary to gain experience with nonlinear noise reduction using examples where the true signal is known. Since this is usually the case only with numerical data generated on the computer, such data are predominantly used in the original literature. These studies have demonstrated that the simple nonlinear scheme, as well as more advanced schemes, which we will describe in Section 10.3, are in fact able to reduce noise in deterministic data without causing undue distortions. Before cleaning experimental data, the more sceptical reader may wish to try the algorithm on artificial data of similar properties in order to gain confidence. Moreover, we recommend as a minimal test for the reliability of the result to study the statistical properties of what the algorithm has eliminated as noise, i.e., of the time series of differences between noisy signal and signal after noise reduction, \( \hat{s}_n = s_n - \hat{x}_n \). Its autocorrelation function should decay fast, there should be no significant cross-correlations between \( \hat{s}_n \) and the signal \( x_n \), and its marginal distribution should satisfy one's expectations of what the measurement noise looks like (in particular, its standard deviation should not exceed the assumed noise amplitude).

Studies with known true signals suggest that a good choice for the size of the neighbourhoods is given by 2–3 times the noise amplitude. While in some cases it is easy to guess the approximate noise level from a plot of the data, we can also estimate it using the correlation sum, as will be discussed in Section 6.7. If we are in doubt about the noise level, we should underestimate it. As we said previously, this will at worst weaken the performance but will not introduce artefacts of the filtering.
For the embedding dimension $m$ it seems reasonable to use a value such that the dynamics are deterministic both in the future and in the past coordinates. Here, the conservative choice is a larger value of $m$ by which we get rid of some of the more dubious neighbours. Within the TISEAN package, the scheme described here is implemented as the program 1azy, which is discussed in Appendix A.8.

**Example 4.4 (Simple nonlinear noise reduction on NMR laser data).** As a tribute to those very reasonable experimental scientists who refuse to be convinced just because a method works on the Hénon map, the data set we choose as our first example is at least half real. The “clean” signal is taken to be 5000 values of the NMR laser data we showed before (see Appendix B.2 for details). They contain about 1.1% measurement noise. Now we artificially add “measurement noise” consisting of random numbers which have been filtered to have the same power spectrum as the data themselves. The rms amplitude of the noise is chosen to be 10% of the amplitude of the data. This kind of noise is called in-band noise and is completely indistinguishable from the signal for any spectral filter. Now we use the simple nonlinear noise reduction scheme to remove the measurement noise again. The result is shown in Fig. 4.5. Of course, our algorithm will not be able to distinguish the real and the additional noise which limits the evaluation of its performance. Still we can reduce the amplitude of the added random component by a factor of about 1.7, using values of $\epsilon$ corresponding to about 2–3 times the noise amplitude. □

**Example 4.5 (Simple nonlinear noise reduction on far infrared laser data).** Let us now proceed to a really real example. In Fig. 4.6 (left hand panel) we show an enlargement by a linear factor of two of a phase portrait of data set A of the Santa
4.5. Simple nonlinear noise reduction

![Figure 4.6](image)

Data set A of the Santa Fe Institute time series competition, Weigend & Gershenfeld (1993), before and after noise reduction with the simple algorithm.

Fe Institute time series competition; see Weigend & Gershenfeld (1993). The data (Appendix B.1) represent the fluctuating intensity of a far infrared laser and can be approximately described by three coupled ordinary differential equations [Hübner et al. (1993)]. Since the raw data are 8-bit integer numbers, the measurement error is at least the discretisation error, which can be assumed to be uniformly distributed in the interval $[-0.5, 0.5]$. Noise reduction would be worth while even if we could just remove this error and thus increase the dynamical range of the data. We apply the noise reduction algorithm assuming an (absolute) noise amplitude of one unit and choose $\epsilon = 2$, i.e. define as neighbours the points which are not further apart than two units in each of the seven directions of embedding space. The resulting data set is shown in Fig. 4.6 (right hand panel) with the same magnification as the original data. All we can say at this stage is that the structure in the data appears enhanced after noise reduction. This is of course an unsatisfactory statement, but we will need more powerful tools in order to verify and quantify noise reduction on real data. In particular, we will need an estimate of the noise level before and after noise reduction. Such an estimate is possible on the basis of the correlation integral (Chapter 6, and Section 10.2).

Let us make a remark about the practical implementation of this simple nonlinear noise reduction method. Straightforward encoding of Eq. (4.11) would result in a time consuming algorithm since we want to clean all points and not just a few. The main computational burden is to form $\epsilon$-neighbourhoods for each of the points we want to correct. We can save orders of magnitude of computer time if we apply a fast neighbour search algorithm instead of trying all possible pairs and rejecting those which are not close enough. In fact, we reduce the asymptotic operation count from $O(N^2)$ to $O(N \ln N)$ if we use a binary tree, or even to $O(N)$ if we use a box-assisted method. Schreiber (1995) describes in detail how this is done. The
TISEAN package uses box-assisted neighbour search methods whenever it is appropriate, where you find implementations of this and more refined noise reduction schemes.

**Further reading**

General sources for the problem of time series prediction are the proceedings volumes by Casdagli & Eubank (1992) and Weigend & Gershenfeld (1993). Classical articles in the field include Farmer & Sidorowich (1987) and (1988), Casdagli (1989), and Sugihara & May (1990). More specific references to advanced prediction methods will be given in Chapter 12. The simple noise reduction method of Section 4.5 has been proposed by Schreiber (1993).

**Exercises**

4.1 Try zeroth (see Section A.4.1) on some artificial data sets. Start out with the two linearly uncorrelated data sets created in Exercise 2.2. How do the prediction errors depend on the prediction time in the two cases?

4.2 Try the above program on 5000 iterates of the Hénon map (Example 3.2), adding different amounts of noise. Compare the prediction error to the (artificial) case where you use the Hénon map itself for predictions. For reassurance in case you find the result puzzling: we obtain a mean prediction error of about 0.1 for Gaussian noise of amplitude 0.05 if we choose neighbourhoods of radius between 0.05 and 0.15.