Rank-based predictability score: a new measure for determinism

NARO, DANIEL

Curs 2013-2014

Director: RALPH GREGOR ANDRZEJAK
GRAU EN ENGINYERIA EN INFORMÀTICA

Treball de Fi de Grau
Rank-based predictability score:
a new measure for determinism

Daniel Naro

TREBALL DE FINAL DE GRAU / ANY 2014

DIRECTOR DE LA TESI
Ralph Gregor Andrzejak Departament DTIC
Acknowledgement I thank my parents for making it possible for me to study the field I like, and to go so far in these studies. I also thank my girlfriend for supporting me in the challenge that this work was, and to understand that I needed sometimes extra time to work on it.

My work would not be now what it is without my supervisor Ralph Gregor Andrzejak: he introduced me into the field of signal analysis, proposed me the core idea of this project and accompanied its development with his advice and expertise. I also sincerely thank Petroula Laiou for her encouragements and useful comments.
Abstract

This work aims to contribute to signal analysis. We apply algorithms to signals extracted from systems which we want to understand better. More precisely, we measure the signal’s degree of predictability, which allows the distinction between deterministic and stochastic dynamics. As a new approach to this type of measure, we define and evaluate a rank-based nonlinear prediction score. The classical distance-based nonlinear prediction error algorithm serves as benchmark. Both methods are applied to known dynamics under controlled conditions with the aim to assess their sensitivity and specificity, and the influence the parameters have on both properties. The new rank-based nonlinear prediction score performs better or equally well as the distance-based measure. We derive that in application to white noise the rank-based measure follows a Gaussian distribution with an expected value of zero and known variance. According to our findings, the rank-based method should be preferred over the distance-based.

Aquest treball pretén contribuir a l’anàlisi de senyals. Apliquem algoritmes a senyals extretes de sistemes que volem entendre millor. Concretament, mesurem el grau de predictibilitat del senyal, el que permet la distinció entre dinàmiques deterministes i estocàstiques. Com a nova estratègia per aquesta mesura, definim i avaluem un mètode no lineal de puntuació de la predictibilitat basat en rangs. La nostra referència serà la mesura clàssica de l’error de predicció basat en distàncies. Apliquem ambdós mètodes en dinàmiques conegudes, amb l’objectiu d’avaluar la sensibilitat i especificitat en funció dels paràmeters. El nou mètode funciona millor o igual de bé que el mètode basat en distàncies. Demostrem que si el senyal és soroll blanc, els resultats del mètode basat en rangs segueixen una distribució gaussiana, amb mitja zero i variància coneguda. Segons els nostres resultats, el mètode basat en rangs hauria de ser el preferit.

Este trabajo pretende contribuir al análisis de señales. Usamos algoritmos que usan señales extraídos de sistemas que queremos entender mejor. Concretamente, medimos el grado de predictibilidad del señal, lo que permite la distinción entre dinámicas determinísticas y estocásticas. Como un nuevo enfoque para este tipo de mesura, definimos y evaluamos un método no lineal de puntuación de la predictibilidad basado en rangos. Usamos la medida clásica del error de predicción basada en distancias como referencia. Aplicamos los dos métodos en dinámicas conocidas, con el objetivo de evaluar la sensibilidad y especificidad en función de los parámetros. El nuevo método funciona mejor o igual de bien que el método basado en distancias. Demostramos que si el señal es ruido blanco, los resultados del método basado en rangos siguen una distribución gaussiana, con media cero, y varianza conocida. Según nuestros resultados, el método basado en rangos tendría que ser el preferido.
Summary

Index of figures x

1 INTRODUCTION 1

2 DEFINITIONS 5
   2.1 Dynamical systems 5
   2.2 Signals 8
   2.3 Delay reconstructions 12
   2.4 Distance matrices 12

3 NONLINEAR PREDICTION 17
   3.1 Rank-based nonlinear predictability score 19
   3.2 Distance-based nonlinear prediction error 26

4 SIMULATIONS 37
   4.1 Gaussian noise 37
   4.2 Autoregressive noise 47

5 CODE USED 61

6 CONCLUSIONS 63
List of Figures

2.1 A simple stochastic process: although we use the same initial conditions, we have three different signals. .................................................. 6
2.2 Memory of an autoregressive function: the higher the coefficient $a$, the stronger the autocorrelation of the system. ................................. 7
2.3 The Lorenz system: a three components deterministic dynamical system. 9
2.4 The Rössler system: another typical three components deterministic dynamical system. ................................................................. 10
2.5 The second component of the Lorenz system vs. time .................... 11
2.6 Delay coordinate reconstruction of the Lorenz system using the second component ($m = 3, \tau = 4$) ....................................................... 13
2.7 Strong similarity between distance matrices of original (A) and delay reconstruction (B) for the correct parameters ............................... 14
3.1 Comparison of a stochastic auto-regressive system with a deterministic Lorenz ................................................................. 18
3.2 In a chaotic deterministic system a small initial deviation imply extremely different results in time. ............................................. 19
3.3 Representation of nearest neighbours use, the average of the future of the most similar states as a prediction tool .............................. 20
3.4 The expected value behaves as expected. The variance of $R$ is proportional to the inverse of $k$ and $N$. ............................................. 27
3.5 If the independence of samples is reduced, the variance of the mean over samples is bigger. .................................................. 28
3.6 The actual value of $N_{\text{eff}}$ is the result of the values of $\eta$, $h$, $W$, and the time index. .................................................. 29
3.7 The derivation of the distribution of $\Omega_{i0}$ superimpose without errors on our simulations. .................................................. 32
3.8 The derivation of the distribution of the mean of squared simulated errors superimpose without estimates on our simulations. .......... 34
3.9 The expected value of $D$ behaves as expected, while its variance is close to our approximation. ............................................... 35
4.1 Because of the geometric sequence the deterioration of the signal due to the addition of noise start increasing rapidly when \( l > 68 \).  

4.2 Example of monotonic evolution of the two measures against noise amplitude.  

4.3 Example of distance curve based on the two measure with a better result for \( \mathcal{R} \).  

4.4 Example of multiple crossing of the threshold by the distance curve based on the result of \( D \): the last point above the threshold and the first below are far apart.  

4.5 Noise robustness for \( D \) in the case of Lorenz with white Gaussian noise.  

4.6 Noise robustness for \( \mathcal{R} \) in the case of Lorenz with white Gaussian noise.  

4.7 Performance comparison between \( D \) and \( \mathcal{R} \) in the case of Lorenz with Gaussian noise.  

4.8 Performance comparison between \( D \) and \( \mathcal{R} \) in the case of Lorenz with Gaussian noise cutting off extreme cases.  

4.9 Noise robustness for \( D \) in the case of Rössler with white Gaussian noise.  

4.10 Noise robustness for \( \mathcal{R} \) in the case of Rössler with white Gaussian noise.  

4.11 Performance comparison between \( D \) and \( \mathcal{R} \) in the case of Rössler with Gaussian noise.  

4.12 Example of monotonic evolution of the nonlinear Prediction Error against autoregressive coefficient: in this example both measures detect the determinism showing no specificity.  

4.13 Autoregressive noise: \( D \).  

4.14 Autoregressive noise: \( \mathcal{R} \).  

4.15 Autoregressive noise: difference between \( D \) and \( \mathcal{R} \).  

4.16 Example of non-monotonic evolution of the distance-based nonlinear predictability measures against noise amplitude (Rössler signal superimposed with autoregressive noise).  

4.17 Performance comparison between \( D \) and \( \mathcal{R} \) in the case of Rössler with Autoregressive noise.  

4.18 Performance comparison between \( D \) and \( \mathcal{R} \) in the case of Lorenz with Autoregressive noise.  

4.19 Three examples of degenerate case where \( D \) does not detect any determinism in the Lorenz signal with autoregressive noise.
Chapter 1

INTRODUCTION

When we want to study some dynamics, either a theoretical model or some real-life data, we have to make some measurements first. How we obtain data from the system depends on its nature. Regardless of the type of data, we can study the obtained signal. The results can then be linked back to properties of the system helping us to understand it.

One example of a system is the brain, from which we can record many different signals such as the electroencephalogram (EEG), functional magnetic resonance imaging (fMRI), or calcium imaging. These recordings can then be analysed to provide information to medical doctors for their diagnosis. As we described before, the results of the signal analysis give us some knowledge about the recorded system. An example of this was presented in the context of epilepsy for the localization of the seizure generating area [Andrzejak et al., 2012].

The measures extracted from the signals can be quite different. For example, we can compute the correlation between two components of the record. We can also measure how random or predictable a signal is. By predictable we mean how well we can use portions of the signal to predict its evolution. This result may then be used to quantify the amount of noise added to the system, for example.

There are different approaches on how to build a measure of predictability. One of them uses the concept of smoothness in the phase space [Salvino and Cawley, 1994]. The authors state that, because a smooth phase space is a sign of determinism, this is a useful measure for predictability. Within the phase space, they use local directional elements which are vectors connecting consecutive states. They propose to assess the degree of smoothness, by creating a grid and evaluating how aligned the local directional elements are within each cell. If the phase space is smooth, all directional elements point to the same direction. The problem is that the authors introduce a set of parameters which has to be tuned for each signal. These extra parameters carry two undesired consequences: first of all they reduce our ability to apply the method automatically. The second point is that we might be forcing the method to return the result we want to see,
what would lead to false conclusions. Moreover, more parameters make it harder to know the expected value of the measure in the null case, i.e. when the signal is pure noise. Knowing what to expect in the case where the signal is totally random is useful. When we use the measure on some unknown dynamics, we have to compare the results to the stochastic case in order to understand the results’ meaning. A similar method uses a grid of regular boxes where we compute the average direction that the dynamics takes across that box [Kaplan and Glass, 1992]. Consecutive states cross the different boxes. The direction of crossing is represented by a unit vector. The average direction is assessed by taking the mean over the unit vectors in one cell. The authors present an approximation for the direction average per box, for the noise signal case, in function of how many paths actually cross that given cell. The approximations get better when the number of crossings is higher. To increase this number we might need to increase the size of the cells, which leads to possible mistakes where the dynamics’ direction changes rapidly.

Another approach to detect determinism is to quantify the average error made when predicting the signal evolution on the basis of similar momentary states [Farmer and Sidorowich, 1987, Kantz and Schreiber, 2004, and references therein]. In such a measure, the lower the error is, the more predictable the signal is. It is not straightforward to derive the expected value of the stochastic case. However, in contrast to the approach of [Salvino and Cawley, 1994], in this measure and in [Kaplan and Glass, 1992] we do not have to tune extra parameters, we can run experiments on stochastic signals to estimate its expected value.

In order to have a more useful technique it is important to know the expected value for the null case. We could use trial cases in order to build a set of expected results, but this might be computationally intensive and could represent an overhead in data. Moreover, we might reach a point where we are interested in the probability distribution of the results. This is even harder than just to obtain the expected value and the variance.

A new approach was proposed for the detection of coupling between two signals using ranks [Chicharro and Andrzejak, 2009]. In this method we do not quantify our absolute error when we predict, we just focus on how precisely we are able to forecast which record in time is the closest to another, the second closest and so on. Because [Chicharro and Andrzejak, 2009] is about coupling, their method uses a signal $X$ to predict another signal $Y$. In order to apply the algorithm on just one signal we have to reformulate its definition. We are motivated by two facts: this method requires the same amount of free parameters as [Farmer and Sidorowich, 1987, Kantz and Schreiber, 2004, and references therein] or [Kaplan and Glass, 1992], and most importantly its definition leads to an easy derivation of the expected outcome for a stochastic signal. The normalization of the values is similar to the one used in [Kaplan and Glass, 1992]. However, in this coupling detection method the expected value is known and equal to zero, leading to a more precise normalization. This rank method is also more sensitive
and specific than a number of previous approaches to detect coupling [Chicharro and Andrzejak, 2009]. In this work we change the definition to transform this coupling detection algorithm into a rank-based predictability score, using the ideas in [Chicharro and Andrzejak, 2009]. Our working hypothesis is that we carry over its useful properties for coupling on predictability.

This adaptation to predictability results in a new rank-based nonlinear prediction score. We test its performance, compared to another, already existing method, the one presented in [Farmer and Sidorowich, 1987, Kantz and Schreiber, 2004, and references therein]. First of all we introduce the definition of both measures and we derive the expected behaviour for a pure noise signal for both of them. We then compare the sensitivity and specificity of both techniques. The specificity is assessed by analysing an autoregressive noise. For the sensitivity we add noise to a deterministic signal and observe at which point the interferences make us conclude that there is no determinism. It is important to note that we use known systems (in our case Lorenz and Rössler dynamics). We want to get clear evidences, and this is only possible if the only unknown element of the study is the measure we want to analyse.
Chapter 2

DEFINITIONS

2.1 Dynamical systems

We analyse different dynamical systems which are characterized by $d$ components evolving in continuous time $t$.

$$\mathbf{x}(t) = (x_1(t), x_2(t), \ldots, x_d(t))$$

Since we sample our continuous models, we work with time-discrete signals. The result of the sampling are time series denoted by $x(t_i)$, where $t_i = i \cdot \Delta t$.

We can distinguish stochastic systems and systems with a deterministic structure. The first type is based on a source of noise. An autoregressive system is an example of stochastic dynamics where the next state is determined by the current state and a random contribution:

$$x(t_i) = a \cdot x(t_{i-1}) + \xi(t_i)$$

(2.1)

where $\xi(t_i)$ are independent samples from Gaussian white noise with zero mean and unit variance. Due to the randomness, even with the same initial state, i.e. $x(t_0)$, the signal, i.e. the subsequent time steps of the signal, will always be different, as we can see in figure 2.1. If the coefficient $a$ is equal to 0, the signal is Gaussian white noise, since each time step is then equal to the $\xi(t_i)$ sample. If the value of $a$ is increased, the memory of the system is enhanced, as shown in figure 2.2. In other words, the value of a state $x(t_i)$ has more influence on the subsequent states $x(t_j)$ where $j > i$.

Deterministic dynamics are such that, for a given basic rule like an ordinary differential equation (ODE), the whole signal is completely defined by the initial conditions. As an example for deterministic dynamics we use the Lorenz system [Lorenz, 1963].

\[
\begin{align*}
\dot{x}_1(t) &= 10(x_2(t) - x_1(t)) \\
\dot{x}_2(t) &= 28x_1(t) - x_2(t) - x_1(t) \cdot x_3(t) \\
\dot{x}_3(t) &= x_1(t) \cdot x_2(t) - \frac{8}{3}x_3(t)
\end{align*}
\]
Figure 2.1: A simple stochastic process: although we use the same initial conditions, we have three different signals. The graph represents three independent realizations of an autoregressive process (equation 2.1 for $a=0.98$)
Figure 2.2: Memory of an autoregressive function: the higher the coefficient $a$, the stronger the autocorrelation of the system.
plotted in figure 2.3. This differential equation is solved with a Runge-Kutta integrator of 4th order. Because of the determinism, we always obtain the same result for the same initial conditions and precision of the integrator.

The Runge-Kutta integrator uses the concept of steps: we want to approximate the sampling of the continuous signal at discrete time points. These integration steps coincide with the $\Delta t$ used for discrete signals. Like for the Euler method, too big integration steps lead to mistakes: the signal is not approximated accurately, and in fact the shape could be totally destroyed. On the other hand, a too low value for the step gives an accurate sampling, with a high sampling frequency. However, for the same amount of sampled points we sample less time. One way to have both advantages is to sample with a high frequency more points, and then to downsample the signal. By downsampling we mean that we just store part of the signal, for example only the even time steps. We might be introducing aliasing, and therefore we might need a low-pass filter. However, this would appear only if we were to apply methods which are based on the frequency domain such as the Fast Fourier Transform. Moreover, the Nyquist frequency tells us we need at least two samples per oscillation for a correct reconstruction. With twenty samples per oscillation we are much higher then the minimum.

The Lorenz system is a common benchmark for nonlinear measures. Another example is the Rössler system [Rössler, 1976], which is defined by

$$
\begin{align*}
\dot{x}_1(t) &= -x_2(t) - x_3(t) \\
\dot{x}_2(t) &= x_1(t) + 0.15x_2(t) \\
\dot{x}_3(t) &= (x_1(t) - 10)x_3(t) + 0.2
\end{align*}
$$

(2.3)

The shape of this second deterministic system is shown in figure 2.4.

## 2.2 Signals

In an experiment we cannot always measure every dimension of the system. To simulate this, we use measurement functions, for example the function $g$:

$$
g(\bar{x}(t)) = x_2(t)
$$

(2.4)

The signals $x(t)$ are the output of the measurement functions, and the system is the input $(\bar{x}(t))$ as in:

$$
x(t) = g(\bar{x}(t))
$$

(2.5)

However, in doing so we loose information about the other components, as we can see in figure 2.5. Just looking at the figure we may not be able to imagine the shape of the full original system.
Figure 2.3: The Lorenz system: a three components deterministic dynamical system.
Figure 2.4: The Rössler system: another typical three components deterministic dynamical system.
Figure 2.5: The second component of the Lorenz system vs. time
2.3 Delay reconstructions

In order to recover some properties of the shape, at least in some way, we will use the delay coordinates according to Takens’ embedding theorem [Takens, 1981]. In this technique a reconstruction vector is defined for each time step \( i \), using a time delay \( \tau \), and an embedding dimension \( m \) as in:

\[
\vec{x}(t_i) = (x(t_i), x(t_{i-\tau}), ..., x(t_{i-(m-1)\tau})) \quad (2.6)
\]

Here \( i \) runs from \( \eta = (m - 1)\tau \) to \( N \). The embedding window \( \eta \) is necessary to avoid negative indices. Let us take one example of a short arbitrary signal with \( N = 9 \).

Table 2.1: Signal used for the examples

<table>
<thead>
<tr>
<th></th>
<th>3</th>
<th>4</th>
<th>7</th>
<th>10</th>
<th>1</th>
<th>5</th>
<th>2</th>
<th>11</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x(t_i) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x(t_{i-\tau}) )</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 2.2: Embedding vectors per time step for the example signal 2.3. Grey colour indicates the impossibility to create the embedding vector.

If we use delay coordinates to reconstruct the Lorenz system (figure 2.6) we can see that we recovered part of the shape: for example the two wings appear again.

2.4 Distance matrices

Given a set of vector \( x(t_i) \) we can define for each pair of time steps the Euclidean distance:

\[
D_{i,j} = \delta_{Euc}(\vec{x}(t_i), \vec{x}(t_j)) = \sqrt{\sum_{r=0}^{m-1} \left( x(t_{i-r\tau}) - x(t_{j-r\tau}) \right)^2} \quad (2.7)
\]

Across all pairs this result in a distance matrix \( D \).

This distance matrix obtained for the Lorenz system and for the reconstruction is shown in figure 2.7. What is interesting is the strong similarity between the two matrices.
Figure 2.6: Delay coordinate reconstruction of the Lorenz system using the second component ($m = 3, \tau = 4$)
A and B: close states in the original are close in the reconstruction, distant states in the original are distant in the reconstruction. The values are distributed alike in both images: same shape of multiples squares, same levels of grey in the same positions. This result should make us even more confident that we are able to make meaningful computations on reconstructed dynamics. Nevertheless, as always when we have to select the value of parameters, the quality of the reconstruction depends on the parameters.

If we take again the example introduced previously, we obtain the distance matrix (see table 2.4).

If we were to accept any state in the ordering of distances for the state \(i\), we would accept states really close in time to it (and therefore really similar in the phase space, too), maybe a state could even be in the embedding window of the other. In any case, these states do not tell us anything about the ability to use other portions of the signal to predict. To avoid the use of these states we use the Theiler window [Theiler, 1986] of
size \( W \): all the index \( j \) such that \(|i - j| \leq W\) will be ignored in the nearest neighbours matrix. In our simulations we use a Theiler window \( W = 10 \). We set the downsampling such that one oscillation is approximately twenty samples long. With this value of \( W \) we are sure that we accept in the embedding only values which are from other recurrences of the trajectory. In our example, and if \( W = 1 \), the distance matrix \( D \) is presented in table 2.4.

Table 2.3: Distance matrix based on the embeddings vectors 2.3.

Table 2.4: Distance matrix based on the embeddings vectors 2.3 taking into account the Theiler window \( W \).
Chapter 3

NONLINEAR PREDICTION

Depending on the type of the system, i.e. stochastic or deterministic, the trajectory may follow some determined path or be totally erratic. Let us illustrate this with figure 3.1. There is no similarity between the two plots. On the left side we can hardly predict from the position of one point what will be the next value. In contrast, on the right side with knowledge of the current state and of the equations of motion it is possible to predict the future state with no error. Of course the question arises of how well we can predict future states, if we do not have this knowledge of the system. What happens if we have just the points represented in the plot and we are asked to predict the future state of some coordinates? If we knew previously that we would not make any error, we now know that we will get some deviation. In order to predict the future of one state, we will look at the data that is available. We average the future of the most similar states to the one we want to predict. These $k$ most similar states are the $k$ nearest neighbours, i.e. the ones for which the distances measured with the distance function take the $k$ lowest values.

We should note that the Lorenz system is a chaotic deterministic dynamical system, where a small deviation at the beginning introduces significant changes at the end. We can represent that by superimposing two plots of the Lorenz system, where both will be defined with the same ordinary differential equations (ODE), but the second one will have a deviation added to the three components of the initial state, randomly selected from the interval $[-0.01, 0.01]$. The results are presented in figure 3.2, where for more clarity just the second component has been plotted. The conclusion to be drawn is that the difficulty of a correct prediction of the future states increases with time. Nevertheless at a short scale it is possible to approximate fairly well the future. For example, even a stochastic scenario, as an autoregressive signal, can be predicted: as we see in figure 2.2 with a high coefficient the system has enough memory so that the $\xi$ contribution does not imply great changes over a short time scale, which enables us to predict.

In order to predict the future of a state, we use the most similar states. But because the reconstruction is so similar to the original system, the closest points in the recon-
Figure 3.1: Left: trajectory of a three component stochastic auto-regressive system, Right: deterministic Lorenz System
Figure 3.2: In a chaotic deterministic system a small initial deviation imply extremely different results in time. Comparing a Lorenz signal, with a variation of itself where a small deviation was introduced.

struction are also the closest in the state space of the original system. We can draw this approach, as we do in figure 3.3: the blue point represent where we want to predict the future, the red points examples of close points, the green the future of these used to make our prediction, and the black the point we try to approximate with our prediction.

3.1 Rank-based nonlinear predictability score

[Chicharro and Andrzejak, 2009] introduce a rank-based coupling detection measure L which shows better results than other coupling detection methods. This is the motivation to apply a similar strategy in the case of the nonlinear predicton measures. The main concept is that we want to predict indices of the $k$ most similar states.

Given a certain reference point $x(t_i)$, at first we determine the $k$ nearest neighbours for each state. To do that, we define the nearest neighbours matrix $C'$, for which each...
Figure 3.3: Representation of nearest neighbours use, the average of the future of the most similar states as a prediction tool.
row $i$ will store the indices of the $j$-th state such that $c_{i,j}$ is the index of the $j$-th closest to $i$ state of the signal. In other words, $C$ is the result of ordering independently each row of the distance matrix $D$, but instead of storing the list of values, we use the list of indices.

In our example with the signal shown in table 2.3, the matrix $C$ looks like the one shown in table 3.1. Later on we explain why we have to leave those time steps which index is bigger then $N - h = 9 - 1 = 8$, where $h$ is the horizon, the number of time steps we look into the future.

<table>
<thead>
<tr>
<th>time index</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>6</td>
<td>8</td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>8</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: $C$ matrix based on the example signal 2.3.

Let us define the rank matrix $G$ and its cells $g_{i,j}$: for a time index $i$, we know that the time index $j$ is the $g_{i,j}$-th nearest neighbour, but in this case the distance is defined only as $|x(t_i) - x(t_j)|$, i.e. without embedding. Nevertheless we use the indexes indicated by matrix $C$ to read the matrix $G$. This is why we take into account $\eta$, i.e. for $G$ we do not take into account those indexes for which we could not have an embedding window. Because we have ordered the matrix, we do not expect to have symmetry in $G$ as we have in $D$. The later is a distance matrix, and our distances are the same if we compare time step $a$ with time step $b$, or time step $b$ with time step $a$, and therefore it is symmetric. The fact that time step $b$ is the closest to $a$ does not mean that time step $a$ is the closest to $b$, therefore matrix $G$ is not symmetric.

In our signal example table 2.3 the distance matrix without embedding window and with $W = 1$ is shown in table 3.1.

The resulting matrix of ranks $G$ is shown in table 3.1.

Using the previously defined matrix $g_{i,j}$, we define the average rank selection $G_i$ as in:
In this definition we read the indices of $k$ closest time steps according to the matrix $C$. In our example, we use $k = 2$, and we will add to these indices $h = 1$. The result of this is shown in table 3.1.

As stated in equation 3.1 we add the value of the horizon $h$ to the indices obtained from $C$, and read the ranks obtained in $G$, when reading $h$ rows ahead. The values for $G_i$ for our example are shown in the table 3.1 if the signal was perfectly predictable, looking the horizon distance $h$ forward for each of the $k$ nearest indices (adding $h$ to the top $k$ $w_{i,j}$), we should recover the top-$k$ for the time step $i + h$. The formula for the perfect match case $G^1$ is:

$$G^1_i = \frac{1}{k} \sum_{j=1}^{k} g_{i+h,c_{i,r}+h}$$

(3.1)

If the selection of the indices was totally random because the signal was so stochastic, the selection of the indices forming the prediction would be the result of $k$-sampling uniformly and without replacement the space $[1, N]$. We should be careful with the value of $N$, but for the moment we simplify using this value.

The average $G_i(X)$ of $k$-sampling uniformly and without replacement the interval $[1, N]$ is the expected value for this type of distributions:

$$G^0_i = \frac{N + 1}{2}$$

(3.3)
Table 3.3: The rank matrix $G$ for our example signal 2.3.

<table>
<thead>
<tr>
<th>time index</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.4: The selection of the $k = 2$ closest neighbours according matrix $C$ for the example signal 2.3.

<table>
<thead>
<tr>
<th>selected index</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r = 1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r = k = 2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The final function $R$ is the distance between the stochastic case and the actual limit, which is normalized using the difference of the stochastic and the perfectly predictable limit:

$$R(X) = \frac{1}{N} \sum_{j=1}^{N} \frac{G_j - G_i^0}{G_i^1 - G_i^0}$$

(3.4)

As we said previously, if the signal is unpredictable we can anticipate the behaviour of $G_i$. As we can see in equation 3.1, the result would then be the average across sampling $k$ values of a discrete uniform distribution with range $[1, N]$. We can draw two conclusions from this, based on our knowledge of this type of distributions. The first one is the limit value $G_i$ we reach when the signal is stochastic:

$$E(G_i) = \frac{N + 1}{2}$$

(3.5)
Table 3.5: Readings of the rank, and values of $G_i$ for the example signal 2.3.

From this we can derive the expected value of $\mathcal{R}$ when the signal is stochastic.

$$E(\mathcal{R}(X)) = E \left( \frac{1}{N} \sum_{j=1}^{N} \frac{G_j - G_i^0}{G_i^1 - G_i^0} \right)$$

$$= \frac{1}{N} E \left( \sum_{j=1}^{N} \frac{G_j - G_i^0}{G_i^1 - G_i^0} \right)$$

$$= \frac{1}{N G_i^1 - G_i^0} \sum_{j=1}^{N} E \left( \frac{G_j - G_i^0}{N - \frac{X}{2}} - \frac{N + \frac{X}{2}}{2} \right)$$

$$E(\mathcal{R}(X)) = 0$$

The second one is the variance of $G_i$. Because we use an average across $k$ in equation 3.1, we divide the variance of the sampling of a uniform distribution by $k$. By using the equation of the variance for a uniform distribution with replacement we introduce some error in the computation.

$$\text{Var}(G_i) = \frac{(N - 1)^2}{12k}$$

We have neglected that we sample without replacement because $k$ is significantly smaller than $N$. We know that for variances we have two basic rules.

$$\text{Var}(a + X) = \text{Var}(X)$$

$$\text{Var}(aX) = a^2 \text{Var}(X)$$

Let us apply this to equation 3.4. We should also recall that $N$ is usually three orders
of magnitude bigger than $k$, therefore we approximate $N - k \approx N$.

$$\text{Var}(\mathcal{R}(X)) = \text{Var} \left( \frac{1}{N - h - \eta} \sum_{j=\eta}^{N-h} \frac{G_i - G_i^0}{G_i^1 - G_i^0} \right)$$

$$= \text{Var} \left( \frac{1}{N - h - \eta} \sum_{j=\eta}^{N-h} \frac{G_i}{G_i^1 - G_i^0} \right)$$

$$= \text{Var} \left( \frac{1}{N - h - \eta} \sum_{j=\eta}^{N-h} \frac{G_i}{G_i^1 - G_i^0} \right)$$

$$= \frac{1}{(G^1_i - G^0_i)^2} \text{Var} \left( \frac{1}{N - h - \eta} \sum_{j=\eta}^{N-h} G_i \right)$$

$$= \frac{1}{(N+1)^2 - \frac{k+1}{2}} \frac{(N - 1)^2}{12k \cdot N}$$

$$= \frac{4}{(N + 1 - k - 1)^2} \frac{(N - 1)^2}{12k \cdot (N - h - \eta)}$$

$$\text{Var}(\mathcal{R}(X)) \approx \frac{(N - 1)^2}{3(N - k)^2 k (N - h - \eta)}$$

$$\approx \frac{(N - 1)^2}{3(N - k)^2 k \cdot (N - h - \eta)}$$

$$\approx \frac{(N - 1)^2}{3(N - k)^2 k \cdot (N - h - \eta)}$$

$$\text{Var}(\mathcal{R}(X)) \approx \frac{1}{3(N - k)k}$$

We compare this analytical approximation and the one for the mean to actual measures: 2500 results of $\mathcal{R}$ using Gaussian white noise of different length as signal ($N = 256, 512, 1024, 2048$). The parameters used are $m = 2, \tau = 1, h = 1$ and $W = 0$. The differences between the variance and expected value of the simulations and the analytical approximations versus $N$ and versus $k$ can be seen in figure 3.4. In the case of the expected value we see the strong similarities between the simulations and the analytical values. In the case of the variance, we see that both curves have the same shape: the analytical approximation is proportional to the observed results. The mismatch is mainly the consequence of the non-independence of the operations which appear in the computation of $\mathcal{R}$. If two time steps are really alike in the reconstruction, than both will have almost the same set of $k$-nearest neighbours, and it is very likely that there
are mutual nearest neighbours. Therefore, there will be many repetitions through all the values of $G_i$ (equation 3.1). Let us understand the effects of repetitions on the variance. For this we design an experiment. First of all, we choose a distribution, we make simulations to obtain $M_{\text{sims}}$ averages over $M_{\text{samples}}$ of the distribution. Because we know the distribution and $M_{\text{samples}}$, we have a theoretical result for the distribution with which the simulation should match. We then redo the same experiment, but instead of using $M_{\text{samples}}$ we duplicate some values, i.e. we increase the dependency. At last we observe the difference between the distributions.

Let us apply this experiment to the continuous uniform distribution. The results are plotted in figure 3.5. We can see that for a higher dependency or more duplicated data, the value of the variance is bigger. Because of the repetitions we are losing degrees of freedom for the central limit theorem (CLT). Because the degrees of freedom decrease in the CLT, the variance is higher. We said before that the nearest neighbours portion of the measure introduces more or less similar data. Based on what we have seen in the experiment, it is normal that our analytical approach, which takes the independence for granted, expects a variance value which is lower than the one we can observe in our experiments.

As we said, we should be careful with $N$. We should use instead $N_{\text{eff}}$ for the effective $N$. With this, we take into account the fact that we cannot select some indices due to $\eta$, $h$ and $W$. We can see the influence of each element in figure 3.6. At the top of the figure we see the influence of $\eta$, at the bottom the one of $h$, and the thickness of the diagonal is the result of the value of $W$. In the general case, where there is no overlapping, we can use neither the points where the embedding window cannot be constructed (i.e. indices lower than $\eta$) nor those where we cannot read an unavailable future (i.e. indices higher than $N - h$), nor the points within the Theiler window of the time step we want to predict (this represents $2W + 1$ indices). In the end, from the $N$ steps the signal has, we can just use $N_{\text{eff}} = N - 2W - h - \eta - 1$. The value of $N_{\text{eff}}$ depends on the fact that the diagonal may or may not overlap with another unavailable region. If there is overlap, the value of $N_{\text{eff}}$ increases.

If we apply equation 3.4 on our signal we obtain that $R = -0.0952$. To understand the intermediate result, we have first to first find out the values of $N_{\text{eff}}$ for this signal, and then deduce the other values. These steps are shown in table 3.1.

### 3.2 Distance-based nonlinear prediction error

In our second approach we measure the predictability in the domain of the amplitude of the signal. For each time step, we want to make a prediction for the evolution of the reference point $x(t_i)$ an interval of $h$ time steps ahead. For each of the time indices of the $k$ nearest neighbours (using the previously defined matrix $c_{i,j}$) we go $h$ steps ahead.
Figure 3.4: The expected value behaves as expected. The variance of $\mathcal{R}$ is proportional to the inverse of $k$ and $N$. Although the analytical approximation gives us a good sense for the value of the variance, the numerical results are higher because of the dependence.
Figure 3.5: If the independence of samples is reduced, the variance of the mean over samples is bigger. There is a match between our expectations for the distributions of means, and what we observed. The more duplicated samples, the higher the variance.
Figure 3.6: The actual value of $N_{eff}$ is the result of the values of $\eta$, $h$, $W$, and the time index. The value of $N_{eff}$ remains the same when the Theiler’s window does not overlap neither with the embedding window nor the end. Black cells correspond to unavailable indices, white to usable indices.
Table 3.6: Intermediate results which leads to the result of $R$ for example signal table 2.3.

and average across the $k$ values. The error that we make is defined by the equation:

$$
\epsilon_{i_0} = \left( x(t_{i_0+h}) - \frac{1}{k} \sum_{r=1}^{k} x(t_{ci_0,r+h}) \right)^2
$$

(3.10)

The distance-based nonlinear prediction error [Farmer and Sidorowich, 1987; Kantz and Schreiber, 2004, and references therein] is defined as the root mean error across all time steps: we obtain

$$
D = \sqrt{\frac{1}{N-h-\eta} \sum_{i=\eta}^{N-h} \epsilon_i}
$$

(3.11)

In this equation $N$ is the size of the signal. We cannot read the values of time steps greater than $N$. This is why the sum has $N-h$ as its upper limit.

Let us derive the distribution of $\epsilon_{i_0}$, given that the signal is a Gaussian distribution whose mean is $\mu = 0$ and variance $\sigma^2 = 1$. In order to clarify the notations we will introduce $\Omega_{i_0}$ as

$$
\Omega_{i_0} = \left( x(t_i) - \frac{1}{k} \sum_{r=1}^{k} x(t_{i,r+h}) \right)
$$

(3.12)

We know that $x(t_i)$ is the sample of the standard normal distribution. If we look at the definition of $A = \frac{1}{k} \sum_{r=1}^{k} x(t_{i,r+h})$ we see that it is an average $A$ of identically distributed independent samples, therefore and according to the Central Limit Theorem, the corresponding distribution is normal with mean $\mu_A = 0$ and variance $\sigma_A^2 = \frac{1}{k}$. $\Omega_{i_0}$ is the difference of two normal distributions. We know that it follows a new normal distribution whose mean will be the difference of the means of the terms, and the variance the sum of the terms’ variances (as stated on page 309 in DeGroot’s textbook [DeGroot and Schervish, 2011]). This means that $\Omega_{i_0}$ has a mean $\mu_{\Omega_{i_0}} = 0$ and a variance...
$$\sigma_{\Omega_{\alpha_0}}^2 = 1 + \frac{1}{k}.$$ We validate this result by simulating the distribution and superimposing our analytical approximation (see figure 3.7). Because the two plots match very well, we know that we can approximate the distribution of $\epsilon_{\alpha_0}$.

$\epsilon_{\alpha_0}$ is the square of the Gaussian distributed $\Omega_{\alpha_0}$. We know that we can find the expected value by using moments ([Weisstein, 2014]) and the values $\mu_{\Omega_{\alpha_0}} = 0$ and $\sigma_{\Omega_{\alpha_0}}^2 = 1 + \frac{1}{k}$:

$$E(\epsilon_{\alpha_0}) = \mu_{\Omega_{\alpha_0}}^2 + \sigma_{\Omega_{\alpha_0}}^2$$
$$= \mu_{\Omega_{\alpha_0}}^2 + 0$$
$$= 1 + \frac{1}{k} \quad (3.13)$$

In order to approximate the value of the variance of $\epsilon_{\alpha_0}$, we use again $\mu_{\Omega_{\alpha_0}} = 0$ and $\sigma_{\Omega_{\alpha_0}}^2 = 1 + \frac{1}{k}$:

$$\text{Var}(\epsilon_{\alpha_0}) = E(\epsilon_{\alpha_0}^2) - E(\epsilon_{\alpha_0})^2$$
$$= E(\Omega_{\alpha_0}^4) - \left(\mu_{\Omega_{\alpha_0}}^2 + \sigma_{\Omega_{\alpha_0}}^2 \right)^2$$
$$= \mu_{\Omega_{\alpha_0}}^4 + 6\mu_{\Omega_{\alpha_0}}^2 \sigma_{\Omega_{\alpha_0}}^2 - \mu_{\Omega_{\alpha_0}}^4$$
$$= 3\sigma_{\Omega_{\alpha_0}}^4$$
$$= 2\sigma_{\Omega_{\alpha_0}}^4 \quad (3.14)$$

To verify our result we run an experiment: again with $2^{22}$ simulations and $k = 5$. We compare our prediction for $E(\epsilon_{\alpha_0})$ and $\text{Var}(\epsilon_{\alpha_0})$ to the values observed when sampling $\epsilon_{\alpha_0}$ out of samples of Gaussian noise (results not displayed in any figure). We see that the difference is very small: the error is in the order of magnitude of $10^{-4}$ for the expected value, and of $10^{-3}$ for the variance.

We do not know very much about the distribution of $\epsilon_{\alpha_0}$. It should be similar to a chi-squared distribution with one degree of freedom, but since it does not have a unit variance, we cannot use this as a property to come a step further in our derivation. Instead we can use the fact that each $\epsilon_{\alpha_0}$ is independent because we are taking a Gaussian noise as signal in our derivation. Therefore, we can apply the Central Limit Theorem to approximate the distribution of $B = \frac{1}{N-h-n} \sum_{i=n}^{N-h} \epsilon_i$. Applying this approach we expect this mean to evolve as a Gaussian distribution with mean $\mu_B = E(\epsilon_{\alpha_0})$ and variance $\sigma_B^2 = \frac{\text{Var}(\epsilon_{\alpha_0})}{N}$. The bigger the value of $N$, the better this approximation will be. We compare once more our analytical conclusions to simulations (figure 3.8).

The definition of $D$ includes a square root as final step. We have to take this into account to approximate the measure’s expected value and variance (Taylor expansions,
Figure 3.7: The derivation of the distribution of $\Omega_{i_0}$ superimpose without errors on our simulations. We used $k = 5$ for the $2^{22}$ simulations where we sampled values of $\Omega_{i_0}$ based on Gaussian white noise.
We know how to approximate this because we have an approximation for the distribution of the previous step, and we have the following properties (where \( f(x) \) is a function applied on the random variable \( x \) and where \( p(x) \) is the probability density function of the random variable \( x \)):

\[
E(f(x)) = \int f(x)p(x)dx \tag{3.15}
\]

\[
\text{Var}(f(x)) \approx (f'(x))^2 \text{Var}(x) \tag{3.16}
\]

In our case \( f(B) = \sqrt{B} \) and \( p(B) \) is the p.d.f. of a Normal Distribution with mean \( \mu_B = 1 + \frac{1}{k} \) and variance \( \sigma_B^2 = \frac{2}{N} \left(1 + \frac{1}{k}\right)^2 \). Resolving the integral in equation 3.15 can be hard, therefore we just approximate the value numerically with the rectangle method.

As we saw in equation 3.16, we have a way to approximate the value of the variance.

\[
\text{Var}(f(B)) \approx (f'(E(B)))^2 \text{Var}(B) \]

\[
= \left( \frac{1}{2\sqrt{E(B)}} \right)^2 \frac{2}{N} \left(1 + \frac{1}{k}\right)^2
\]

\[
= \left( \frac{1}{2\sqrt{E(\epsilon_{i0})}} \right)^2 \frac{2}{N} \left(1 + \frac{1}{k}\right)^2
\]

\[
= \left( \frac{1}{2\sqrt{1 + \frac{1}{k}}} \right)^2 \frac{2}{N} \left(1 + \frac{1}{k}\right)^2
\]

\[
= \frac{1}{4(1 + \frac{1}{k})^2} \frac{2}{N} \left(1 + \frac{1}{k}\right)^2
\]

\[
= \frac{1}{2N} \left(1 + \frac{1}{k}\right)
\]

We can observe the difference between the measures and our analytical predictions in figure 3.9. We see that our analytical approximation for the expected value is extremely close to our measures. We also see that an increasing \( N \) takes us closer to the observed value. It seems that our derivation for the mean of the squared distance is not accurate for small signal sizes \( N \), but it can be improved by increasing this parameter. The variance, however, is below our prediction. This does not correspond to what we have seen as expected behaviour for non independent signal. Despite intensive consideration and discussions of this point we cannot resolve resolve this mismatch and leave it for future investigation.
Figure 3.8: The derivation of the distribution of the mean of squared simulated errors superimpose without estimates on our simulations. We used $k = 5$, $N = 2048$, for the $2^{16}$ simulations based on Gaussian noise as.
Figure 3.9: The expected value of $D$ behaves as expected, while its variance is close to our approximation. An increasing $N$ makes the approximation better, which means that the approximation of the properties of the mean of squared distances needs more samples to converge as expected. ($m = 2, \tau = 1, h = 1$).
Chapter 4

SIMULATIONS

We want to compare the sensitivity and specificity of both the distance-based prediction error $D$ and rank-based prediction score $R$. We call a measure sensitive if it detects the presence of a deterministic signal even if we add noise to the input. Measurement noise is added to the signal after the integration. We call a measure specific regarding determinism if the value indicating determinism only appears in the case of a real deterministic signal. We use the Lorenz and Rössler signals (Equations 2.2 and 2.3, respectively) superimposed with Gaussian white measurement noise or autoregressive measurement noise. The two measures have four parameters: the embedding dimension $m$, the time delay $\tau$, the prediction horizon $h$, the Theiler window $W$ and the number of nearest neighbours $k$. In order to evaluate the parameter dependence of the results and in order to prove that our conclusions are not a special case for one setting of the parameters, we use the sets:

$$m \in \{2, 4, 6, 8, 10\}$$
$$\tau \in \{1, 5, 10, 20\}$$
$$h \in \{1, 5, 10, 20\}$$
$$k \in \{1, 5, 10, 20\}$$

As in other works, we use a fixed $W$ [Andrzejak et al., 2012]. In our case $W = 10$.

4.1 Gaussian noise

Before adding noise, the signal is normalized to zero mean and unit variance. We down-sample the signal in order to have approximately 20 points per oscillation. This value gives a compromise between precision and the cost in memory which this represents for the computer. It is also comparable to other experiments in the literature ([Wayland et al., 1993] or 17 points per cycle in [Salvino and Cawley, 1994]). We then add one sample of a Gaussian white noise to each time step. The mean of the noise is set to zero.
Figure 4.1: Because of the geometric sequence the deterioration of the signal due to the addition of noise start increasing rapidly when $l > 68$. Until this point the shape of the signal is pretty much intact. From $l = 70$ upwards it is even hard to recognize the oscillations any more.

and the standard deviation $\sigma_{\text{noise}}$ is set in the following way:

$$\sigma_{\text{noise}} = 10^{-3} \cdot r^l$$

where $l = 1, \ldots, 100$ and $r = 1.1$. For each noise amplitude we run 500 or 1000 trials to make the results more reliable (cf. chapter 5). In order to give an idea of how altered is the signal after adding the noise we show four different examples obtained for four different values for the index $l$.

The mean of these trials against the noise amplitude should evolve from the value for high determinism to high unpredictability, i.e. for the distance-based measure $D$ from a low result to a high result and the opposite for the rank-based measure $R$. We can see the resulting S-shaped curves in figure 4.2. We can observe that the range of the result increases with the amplitude of the noise: we need a measure which takes
Figure 4.2: Example of monotonic evolution of the two measures against noise amplitude. We show the mean (solid line) and range across realizations (dashed lines). Parameters: $m = 2, \tau = 1, h = 1, k = 1$.

this into account when comparing the results. The other problem is that the expected value of the distance-based measure for the highest noise amplitude case depends on the parameters.

We use the Wilcoxon Statistic [Hanley and McNeil, 1982], which returns the probability $d$ that a random element sampled from a distribution A is bigger than one randomly sampled from a distribution B. As explained in [Hanley and McNeil, 1982], the straightforward implementation of the statistic is based on scoring the results of comparing the elements of both distributions. If A’s sample is bigger than B’s, we score with 1, if equal with $\frac{1}{2}$ and if smaller with 0. The actual output $d$ of the Wilcoxon Statistic is the average score obtained through all comparisons. A value of $d = 0.5$ implies that on average all tests show that the A and B share the same median. For $d < 0.5$, A has lower values than B, and for $d > 0.5$ A has higher values than B. The reference distribution could be parameter A or parameter B.
We use the maximal noise level as reference. We compare each noise level with the maximal noise level as reference (obtaining for each noise level a distance $L_{\text{max. noise}}$ towards the reference) in order to find the first noise level where we detect the determinism. When computing $L_{\text{max. noise}}$, the order of the parameters is important. One input would return a result higher than 0.5, the other one lower. For our use, both results carry the same meaning since we see in both the distance to the reference. $D$ and $R$ evolve in two different ways: $D$ decreases when there is determinism, while $R$ increases. We will set the reference as the parameter A or B in order to always get a $L_{\text{max. noise}}$ closer to one for determinism. Let us observe the data contained in the S-Surve 4.2, but plotting $L_{\text{max. noise}}$(figure 4.3). As we said the the $L_{\text{max. noise}}$ values should be greater than 0.5. If the prediction error detects a greater determinism, the result of $D$ is closer to 1. We expect high values for low noise values, but the question is at which point we will start having them. The sooner when read from right to left (i.e. the higher the values of the noise) the distance $L_{\text{max. noise}}$ gets bigger, the more sensitive the measure appears to be.

We want to evaluate the results automatically, for all parameter settings. Therefore, we define a score for each case, that we can then compare. We define a threshold where we are confident that we detected the determinism at $\Delta = 0.95$. We want to know the index $l$ of the noise amplitude (see equation 4.2) at which for the first time (from left to right) we were below this threshold. The values $l_{\text{critical,}D}$ and $l_{\text{critical,}R}$ of the index $l$ where $D$ and $R$ cross the threshold respectively, gives us an indication of the robustness against noise of the measure. The higher $l_{\text{critical,}D}$ and $l_{\text{critical,}R}$, the more noise can be added and still $D$ and $R$ detect the predictability respectively. Therefore the higher the values, the more sensitive the measure. Searching for the first below and not the last above (which could be seen as equivalent) we are stricter with the oscillations of $L_{\text{max. noise}}$ as we give the worst possible grade to that result. If there are fluctuations of $L_{\text{max. noise}}$ in the result around the threshold value, there will be some difference in the index $l$ between the first point under the threshold and the last over. Depending on the parameter set and the input, the difference can be huge as in figure 4.4.

We know study the dependence of $l$ on the parameters. We see in figure 4.5 that an increasing $\tau$ and $m$ lead to a decrease of $l_{\text{critical,}D}$ up to the point where there is no detection any more ($l_{\text{critical,}D} = 0$). In order to understand this result we should recall that the signals are downsampled so that we have approximately 20 points per oscillation. This means for example for $m = 4$ and $\tau = 10$, the total time covered by the embedding vector already exceeds one oscillation period. This makes the prediction harder as compared to small embedding windows. As we can see in figure 4.6 the rank-based measure $R$ decrease the sensitivity when $m$ and $\tau$ increases, too. Both measures also share a decrease in sensitivity when $h$ is increased. In the case of $D$ $k$ does not have great effects, in the case of $R$ it increases slightly the sensitivity.

Now that we have both $l_{\text{critical,}D}$ and $l_{\text{critical,}R}$ we can define the difference $\Delta_l = l_{\text{critical,}D} - l_{\text{critical,}R}$. Positive and negative values of $\Delta_l$ indicate higher sensitivity of $D$ and $R$,
Figure 4.3: Example of distance curve based on the two measure with a better result for $\mathcal{R}$. Both behave similarly, but $\mathcal{R}$ reaches higher $L_{\text{max, noise}}$ value for higher noise amplitudes. Parameters: $m = 2, \tau = 10, h = 20, k = 1$
Figure 4.4: Example of multiple crossing of the threshold (shown by the darked black line) by the distance curve based on the result of $D$: the last point above the threshold ($\triangle$) and the first below (□) are far apart. In this example ($m = 5$, $k = 1$, $\tau = 3$, $h = 3$), the last point above the threshold and the first below are 59 indexes apart.
Figure 4.5: Noise robustness for $D$ in the case of Lorenz with white Gaussian noise. $l_{\text{critical},D}$: minimal noise level at which $D$ does not detect the determinism any more (the higher to more robust is $D$). The noise level is expressed as the exponent $l$ in the equation 4.2.
Figure 4.6: Same as figure 4.5 but for $\mathcal{R}$.
Figure 4.7: Performance comparison between $D$ and $R$ in the case of Lorenz with Gaussian noise. Difference of the index of the noise step at which $D$ and $R$ do not detect the determinism any more. The higher the value the more sensitive is $D$ compared to $R$, respectively. As we saw in figure 4.5, for a broad set of parameters $D$ does not detect the determinism at all. Therefore, we see a clear preference for $R$, i.e. negative values of high magnitude of $\Delta_t$, in figure 4.7. To better resolve the details we cut out the most extreme results by rescaling the colours and obtain figure 4.8. We can observe that $R$ is overall better, and the advantage over $D$ is increasing with the horizon $h$.

The results for the Rössler signal (equation 2.3) superimposed with white noise are obtained using the same strategy. The use of $l_{\text{critical},D}$, $l_{\text{critical},R}$, and $\Delta_t$ is the same. We can again observe in figure 4.9 and 4.10 the same dependence on $h$. The main difference in this case is that an increasing $k$ increases the sensitivity for both measures. This may be induced by an intrinsic difference between the Lorenz and the Rössler signal. On the other hand, as introduced in chapter 5, there are differences in the way the simulation were run. At this point we start using signals with a length $N = 2048$ instead of $N = 1024$. As we still use around 20 points per cycle, the number of cycle rises, making it more probable to find more near neighbours. By increasing the value of
Figure 4.8: Same as 4.7 but cutting out the most extreme results.
Figure 4.9: Analog to 4.5 but for a Rössler signal.

$k$, $D$ and $R$ can take advantage of this effect, improving the sensitivity.

The difference of both (figure 4.11) gives a slight advantage for $D$: the mean difference between $l_{\text{critical}, D}$ and $l_{\text{critical}, R}$ is 0.125. Since the common ratio $r = 1.1$, we have that the distance-based $D$ still detects the determinism with a noise level 1.0120 higher.

### 4.2 Autoregressive noise

As stated in equation 2.1, when using autoregressive noise we have to understand the influence of the autoregressive coefficient $a$. We know that with a higher value of $a$, the system has more memory. This memory implies that the signal is more predictable. Nevertheless, because an autoregressive signal is stochastic, we do not want a nonlinear predictability measure to detect determinism. This would be a sign of lack of specificity. Nevertheless, if what we are interested in is detecting predictability, it is a sign of sensitivity.

We compare both $D$ and $R$ on different autoregressive signals with coefficients $a$
Figure 4.10: Analog to 4.6 but for a Rössler signal.
Figure 4.11: Same as 4.7 but for a Rössler signal
defined by:

\[ a = 1 - 0.0001 \cdot 1.202264^l \text{ where } l \in [1, 50] \]  

(4.3)

For this range of values, when \( l \) is high, the \( a \) is close to 0: we should there expect the nonlinear predictability measure to take values close to those of total randomness. As we move towards lower values of \( l \), the memory of the autoregressive process causes both nonlinear predictability measures to indicate at least some predictability. In the end, the resulting evolution over \( a \) resembles the shape of a nonlinear predictability measure against noise amplitude (cf. figure 4.2) as we can see in figure 4.12. Because of this similarity, we can apply the same strategy and analyse the index at which we supposedly detect a determinism: see figures 4.13 and 4.14, which are analogous to 4.5 and 4.6. In the case of a deterministic signal, a high index for \( l_{\text{critical},D} \) and \( l_{\text{critical},R} \) was a sign for sensitivity. In the case of a stochastic signal, a low value of \( l_{\text{critical},D} \) and \( l_{\text{critical},R} \), or no detection at all is a sign for specificity, when we narrow our interest to determinism.

The correlation between samples decreases with an increasing temporal distance. Trying to predict more time steps ahead is then harder, and the nonlinear predictability measure results will detect less determinism. This explains why the increasing horizon helps gaining specificity (figures 4.13 and 4.14). A higher \( m \) and a higher \( \tau \) reduce the values of \( l_{\text{critical},D} \) and \( l_{\text{critical},R} \): we increase the specificity. As we have seen previously, making the embedding window bigger makes it harder to predict correctly. As we saw previously, an increasing value of \( h \) decreases \( l_{\text{critical},D} \) and \( l_{\text{critical},R} \). However in this case, a higher value of \( k \) reduces \( l_{\text{critical},D} \) and \( l_{\text{critical},R} \). In this experiment this difficulty translates to a higher specificity. We compare both results by taking the difference \( \Delta l \) equal to \( l_{\text{critical},D} - l_{\text{critical},R} \). If \( R \) shows more specificity, \( l_{\text{critical},R} \) is smaller than \( l_{\text{critical},D} \), and \( \Delta l \) is therefore positive. If \( D \) is more specific, then \( \Delta l \) is negative. We see the results for the different parameters set in figure 4.15, which is analogous to figures 4.7 and 4.11. We see, as for the Rössler contaminated with Gaussian noise, a slight advantage for \( D \). This measure shows specificity 0.2781 steps before. Nevertheless, if we use these methods just to detect predictability, then \( R \) is more sensitive.

We now have reference \( L_{\text{max, noise}} \) results for a pure autoregressive signal. We change the role of the autoregressive signal. We use it as noise which is superimposed to our usual systems. We start with the Rössler system. We should mention that we now have a new parameter: the autoregressive coefficient \( a \). For practical reasons (i.e. reducing the computation time) we limit ourselves to \( a = 0.98 \). As for the case of the contamination with white noise, we first look at the \( L_{\text{max, noise}} \) results (figure 4.16). The most interesting fact is that there is a maximum or minimum at the intermediate noise level. For low noise amplitudes, we know that we have an almost pure Rössler signal, leading to a nonlinear predictability measure result reflecting high determinism. For high noise amplitudes, we know that we have an almost pure autoregressive signal where \( a = 0.98 \). This leads to a value similar to the one we saw before when no deterministic signal was
Figure 4.12: Example of monotonic evolution of the nonlinear Prediction Error against autoregressive coefficient: in this example both measures detect the determinism showing no specificity
Figure 4.13: Autoregressive noise: $\mathcal{D}$. Index of the noise step at which the $\mathcal{D}$ starts returning a value of non-predictability. If we restrict ourselves to detect only determinism then a lower value is a sign of specificity. If we want to detect predictability then a higher value is a sign of sensitivity.
Figure 4.14: Same as figure 4.13 but for $\mathcal{R}$.
Figure 4.15: Autoregressive noise: difference between $\mathcal{D}$ and $\mathcal{R}$. Difference of the index of the noise step at which the distance-based and rank-based measure start returning a value of non-predictability. If we restrict ourselves to detect only determinism then a lower value is a sign that $\mathcal{D}$ is more specific. If we want to detect predictability then a higher value is a sign of more sensitivity for $\mathcal{D}$. 
involved. For intermediate noise levels, it is much harder to guess the value returned by the nonlinear predictability measures. From this example we see that there is a bounce towards non-predictability.

We need to change our metrics accordingly: we cannot compare any more with the highest noise amplitude result, since it is not the result showing minimum predictability. To avoid the disturbance originated by the rebound, we use the point of the lowest noise amplitude as reference. We then ask the following question: which is the first point which shows significantly more unpredictability? We use the values $l_{\text{critical, } D}$ and $l_{\text{critical, } R}$ for the index $l$ where $D$ and $R$ cross the threshold, respectively. The more sensitive a measure is, the more noise added to the signal it will tolerate. In other words, the higher the values of $l_{\text{critical, } D}$ and $l_{\text{critical, } R}$ are, the more sensitive $D$ and $R$ are, respectively. With this new metric we can again compare the indices of sensitivity using $\Delta_l = l_{\text{critical, } D} - l_{\text{critical, } R}$ (figure 4.17). The way the difference $\Delta_l$ is defined, a positive value means that

![Figure 4.16: Example of non-monotonic evolution of the distance-based nonlinear predictability measures against noise amplitude (Rössler signal superimposed with autoregressive noise).](image)
$\mathcal{D}$ is more robust against noise, whereas a negative value translates to the case where $\mathcal{R}$ is more sensitive. We see, as in the case of Rössler with Gaussian noise, a slight advantage in favour of $\mathcal{D}$: on average we detect the determinism 0.5719 steps later with $\mathcal{D}$, which is equivalent to a noise amplitude 1.1173 times bigger.

We observe strong similarities with the combination of Lorenz with Gaussian noise, when the signal is instead superimposed with an autoregressive signal (figure 4.18). As in the case of Gaussian white noise, the most extreme differences between $\mathcal{D}$ and $\mathcal{R}$ are due to those cases where we do not find a point which crosses the threshold for $\mathcal{D}$. When we had no rebound we took the maximal noise case as reference knowing that it would show no determinism. Now that we take the reference where the noise is minimal, we have to understand the meaning of an empty result. The more likely explanation is that we start in a situation where we do not detect the determinism at the lowest level of noise. Perhaps the method could always detect the determinism or the rebound could affect us in some unexpected way. In order to be sure, we look into the raw results. For three different settings, randomly chosen from the problematic situations (figure 4.19) we see that the results for the lowest noise level are even higher than for an autoregressive signal. This proves that our metric is correct in considering those as degenerated cases. As in section 4.1, the problems with $\mathcal{D}$ appear when both $m$ and $\tau$ take high values. The value of $h$ and $k$ also have an importance since the number of degenerated cases increases with those two parameters. The best result for $\mathcal{D}$ is when it stops detecting 6 steps after $\mathcal{R}$, i.e. with a noise level 3.2012 higher. Nevertheless, due to all the cases where $\mathcal{D}$ fails, $\mathcal{R}$ is again clearly more sensitive.
Figure 4.17: Performance comparison between $\mathcal{D}$ and $\mathcal{R}$ in the case of Rössler with Autoregressive noise. Difference of the index of the noise step at which $\mathcal{D}$ and $\mathcal{R}$ do not detect the determinism any more. The higher the value the more sensitive is $\mathcal{D}$ compared to $\mathcal{R}$. 
Figure 4.18: Same as 4.17 but with Lorenz as signal.
Figure 4.19: Three examples of degenerate case where $\mathcal{D}$ does not detect any determinism in the Lorenz signal with Autoregressive noise. In the three examples, we already start in an unpredictable state and decrease towards the value for the autoregressive signal.
Chapter 5

CODE USED

The experiments were conducted using Matlab on different computers. The code to compute the distance matrix is an adaptation of the code shared in the article [Chicharro and Andrzejak, 2009] (ntsa.upf.edu). Both the code for the distance-based prediction error and for the rank-based prediction score was carefully revised and checked for consistency against other versions provided by the supervisor. The code was also analysed form a statistically point of view (e.g. test that $R$ really outputs a normal distribution with zero as expected value when the input is Gaussian white noise).

Some of the lightest computations, as running some statistical test could be done on a laptop, but due to the high amount of different possible settings, we need more resources. Luckily each setting is independent of the others and we do not need inter-process communication of any kind, except recollecting the results. This makes the parallelisation of the simulations really easy. They were at first run on a server of 24 cores using a Matlab session provided with 8 pool-workers of the Parallelisation Toolbox (i.e. we were able to run the experiments over one of the parameters in different processes, dividing the time needed by around 8). When the DTIC cluster become available the simulations were migrated there, and increased drastically in speed. Instead of using pool-workers, we split the simulations in jobs of a job-array. For each instance, a Matlab session is started and is told what the index in the array is. With this information it can deduce the parameters setting and the operations to be run on them.

In order to improve even more the results, some modifications were applied to the code: we reuse the distance matrix for all the combinations of $k$ and $h$ given some $m$ and $\tau$ instead of using one new for each new parameter setting. The consequence of this is that we do not have to recompute the distance matrix, reducing once again the computation time.

With all this enhancement we were able to add more combinations to the parameters, double the length of the signal from 1024 to 2048, and make 1000 realisations instead of 500. Because the improvement were done over the course of the thesis, the simulations do not share all characteristics.
Chapter 6

CONCLUSIONS

We have analysed the behaviour of two different measures, the distance-based nonlinear prediction error and the rank-based nonlinear prediction score. We focused on the sensitivity and the specificity of both strategies. We used as signal two well-known dynamics, the Lorenz and the Rössler systems, frequently used as test cases, and two types of noise, Gaussian noise and autoregressive noise.

Our analytical derivations proved that we were able to characterize the qualitative behaviour correctly for both algorithms when the input was sampled from uncorrelated Gaussian noise. In the case of the rank-based method $R$, the derivations were much easier and we were even able to make correct conclusions about the complete distribution of the results. More importantly, $R$ has the advantage to have an expected value of 0 regardless of the parameters. Both methods have a variance which is proportional to the inverse of the signal’s size $N$ and the number of nearest neighbours $k$. This is conceptually easy to understand (we take mean values across $k$ and $N$ in the definitions). Nevertheless, in the case of $R$ the derivation of the variance does not need a final approximation.

In the case of the Lorenz dynamics, both for Gaussian and autoregressive noise, there are parameter settings where $D$ does not detect determinism at all. In contrast, $R$ is able to detect there the predictability of the signal, leading to a much higher sensitivity. The distance-based method $D$ was never able to obtain a clearly better result than $R$. When we apply the method to a Rössler signal, the results are slightly better for $D$, and this is also the final result of our analysis for the specificity of both methods.

The better results obtained by $R$ for the Lorenz signal and its parameter-independent expected value should be a reason to consider it as the preferred method. We cannot claim that $R$ is clearly better, mainly because we apparently lose specificity when we restrict ourselves to the detection of determinism. This study could be further enhanced by evaluating how well both methods take advantage of surrogates to increase the specificity ([Theiler et al., 1992]; [Schreiber and Schmitz, 2000]). When we want to test a system against a null hypothesis such as ”the system was recorded from an uncorre-
lated linear stochastic, stationary, Gaussian amplitude distributed process”, the method of surrogates is a useful technique. It provides us with extra evidence which enhances the criterion of the method.

It would also be interesting to test both methods with experimental data and to evaluate how often they match the discrimination of an expert. As we said in the introduction, we can use for example EEG recordings as real data. The use of surrogates may also be useful in this application. In fact they are already used in this domain (e.g. [Andrzejak et al., 2012]).

Finally, a third way to further improve the study of both methods is to see how well both of them can be translated to programming technologies such as OpenCL, CUDA, or even the Parallel Computing Toolbox of Matlab. The three technologies proposed allow the use of General-Purpose Computing on Graphics Processing Units (GPGPU). Since Graphic Processing Units (GPU) are able to run operations on matrices efficiently, their use may be helpful to increase the speed when computing the distance matrix. We generally use analog-to-digital converters to obtain real-life signals. Since these devices have limited resolution, we do not need double precision floats. Therefore, we can use GPUs for videogames instead of models for professional use such as Quadro or Firepro. This means that both methods could run faster on cheaper computers, so that more use case scenarios would become possible.
Bibliography


