Chaos Detection in Noisy Discrete Dynamical Systems using Lyapunov Exponent

An Application to Financial Time Series

Francesco Bacchiocchi, Gianmarco Genalti

Project for the course of Discrete Dynamical Models
Politecnico di Milano, Fall 2020

Contents

1 Introduction

The presence of Chaos in financial time series has been discussed for a long time and the debate about it is not still closed despite many literature did not find evidence of it. Since financial time series with daily granularity are an example of discrete dynamical system, we will introduce a mathematical definition of Chaos based on the notion of **Lyapunov Exponent** for a discrete time dynamical system. In particular, we will introduce a methodology to estimate it based on a **Feed Forward Neural Network**, such an architecture is able to fit any n-dimensional function properly dealing with the intrinsic noise of the financial environment. We will implement and then report some results coming from real world data, with the aim to detect chaotic behaviors.

2 Chaotic Behaviour and Lyapunov exponent

2.1 Preliminary definitions

We firstly introduce the definitions of Chaos and Lyapunov exponent related to a discrete dynamical system.

Definition 1. *A Discrete dynamical system $f : I \to I$ (with $I \subseteq \mathbb{R}$) is said **Chaotic** if:*

- *The periodic orbits are dense (i.e. any interval $(a, b) \subseteq I$ contains at least a point belonging to a periodic orbit)*
- *f is topologically transitive: for any $x, y \in I$ and for any $\epsilon > 0$ there exists $z \in I$ and $k \in N$ such that:*
$$|z - x| < \epsilon \qquad |f^k(z) - y| < \epsilon,$$
- *The system exhibits a sensitive dependence on initial conditions, that is: there exists $\delta > 0$ such that, for any $x \in I$ and for any $\epsilon > 0$ there exists $z \in I$ and $k \in N$ such that:*
$$|z - x| < \epsilon \qquad |f^k(x) - f^k(z)| > \delta,$$

Before providing a formal definition of Lyapunov exponent we will present a simple example related to the well known map $x_{n+1} = f(x_n) = cx_n$ where $n \in \mathbb{N}$, $f : \mathbb{R} \to \mathbb{R}$ and $c > 1$ is a real constant. Solutions of previous equation are given by $x_n = c^n x_0$ with $x_0 \in \mathbb{R}$. For this type of system we are interested in seeing how an initial error $E_0 > 0$ on x_0 is propagated after n iterations. Let' s call $|E_n| = |f^n(x_0) - f^n(x_0 + E_0)|$ the error after n iterations, in our case this value becomes: $|E_n| = |c^n x_0 - c^n(x_0 + E_0)| = c^n E_0$. If we consider now the ratio $|E_n/E_0|$ we get:

$$\left|\frac{E_n}{E_0}\right| = c^n$$

Clearly the parameter c provides the rate of growth of the error per iteration and so bigger values of c are associated to a an higher growth of the initial error E_0. Since the ratio $|E_n/E_0|$ may become too large as $n \to \infty$ we can take the natural logarithm of previous equation getting:

$$\frac{1}{n} \ln\left|\frac{E_n}{E_0}\right| = \ln c$$

We want now to define a similar quantity for a generic discrete dynamical system.

Definition 2. *Given a discrete dynamical system $f : I \to I$. Given $x_0 \in I$ $I \subseteq \mathbb{R}$, the Lyapunov exponent $\lambda(x_0)$ of the system is defined as follow:*

$$\lambda(x_0) = \lim_{n\to\infty} \lim_{E_0\to 0} \frac{1}{n} \ln\left|\frac{E_n}{E_0}\right| \tag{1}$$

where: $E_k = f^k(x_0) - f^k(x_0 + E_0) = f(x_{k-1}) - f(x_{k-1} + E_{k-1})$ for $k \geq 1$

From definition (1), we can immediately understand that positive values of Lyapunov exponent are associate to those system showing sensitive dependence on initial conditions indeed for large values of n we can relay on the following approximation:

$$|E_n| \approx e^{\lambda(x_0)n}|E_0| \tag{2}$$

Thanks to formula (2) we can see that two close trajectories in phase space starting respectively in x_0 and in $x_0 + E_0$ will be exponentially far apart as the number of iterations n increases.

Remark 1. *The previous definition can be rewritten in a more compact form, consider:*

$$\frac{1}{n}\ln\left|\frac{E_n}{E_0}\right| = \frac{1}{n}\ln\left|\frac{E_n}{E_{n-1}}\frac{E_{n-1}}{E_{n-2}}\cdots\frac{E_1}{E_0}\right|$$

thanks to the properties of logarithms we get:

$$\frac{1}{n}\ln\left|\frac{E_n}{E_0}\right| = \frac{1}{n}\sum_{k=1}^{n}\ln\left|\frac{E_k}{E_{k-1}}\right| \tag{3}$$

Finally assuming $f \in C^1(I)$, applying the chain rule in cascade we have:

$$\lim_{E_0\to 0}\left|\frac{E_n}{E_0}\right| = \lim_{E_0\to 0}\left|\frac{f^n(x_0+E_0) - f^n(x_0)}{E_0}\right| = |f'(x_{n-1})\ldots f'(x_0)|$$

Substituting it in (1) we finally obtain:

$$\lambda(x_0) = \lim_{n\to\infty}\lim_{E_0\to 0}\frac{1}{n}\ln\left|\frac{E_n}{E_0}\right| = \lim_{n\to\infty}\sum_{k=1}^{n}\ln|f'(x_{k-1})| \tag{4}$$

Since the the ratio $|E_k/E_{k-1}|$ represent the error propagation from iteration $k-1$ to iteration k, thanks to (3) we can have an another intuition of the meaning of Lyapunov exponent which can be actually considered as an average of the logarithm of the error propagation at each iteration. Formula (4) is useful for a matter of numeric computation since the value of the ratio $|E_n/E_0|$ may become too large as $n \to \infty$ and cause problems of overflow.

2.2 Examples

In this section we will present two remarkable examples of sensitive dependence on initial conditions related to the well known logistic map: $h_a(x) = ax(1-x)$ for a discrete dynamical system in both the cases $a = 4$ and $a > 4$.

Example 1. Let's consider firstly the discrete dynamical system given by the Logistic map: $h_4(x) = 4x(1-x)$ where $h_4 : [0,1] \to [0,1]$. ($i.e$ $x_{n+1} = h_4(x_n)$ $n \in \mathbb{N}$). To better understand the meaning of Lyapunov exponent we iterate $n = 9$ times both the initial value of $x_0 = 0.202$ and the same when an error equal to $E_0 = 0.001$ is added computing the previous introduced ratio $|E_n/E_0|$. In this case we get: $|E_n/E_0| = 256.2$. Has we can clearly see we are delaying with a great amplification of the initial perturbation E_0 and we expect that the Lyapunov exponent in this case would be greater than zero. Figure (1) shows how in $n = 9$ iterations the initial error is propagated. As argued before we expect a positive value of Lyapunov exponent but before to present its exact value we will to try to estimate it with some numerical simulations, in particular given $x_0 \in [0,1]$ we can rely on approximation (2) rewritten as follow:

$$\frac{1}{n}\ln\frac{E_n}{E_0} \approx \lambda(x_0) \tag{5}$$

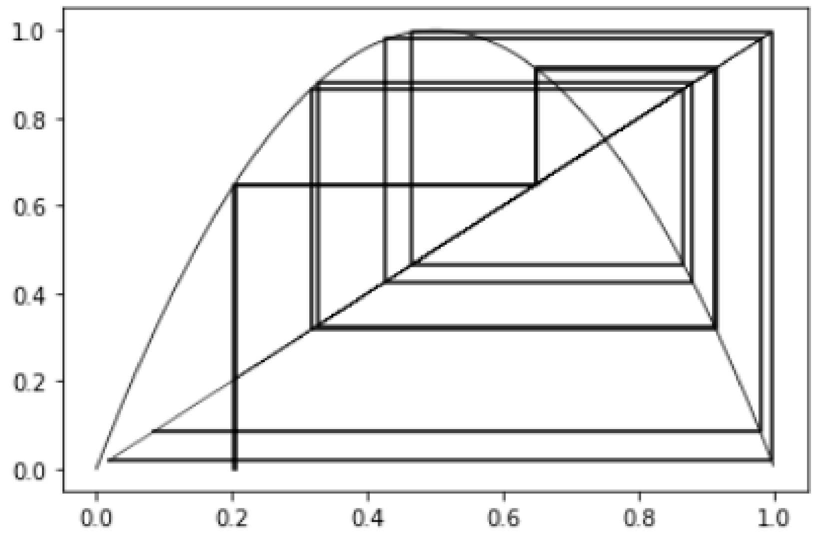

Figure 1: Iterations of the logistic map with initial value of x_0 and $x_0 + E_0$

which results meaningful when the number of iterations n is sufficiently large and E_0 is sufficiently small. Note that in this case we will not provide any information about the accuracy of the estimation but we will just provide for different values of x_0 and different initial perturbation E_0 the respective approximated Lyapunov exponent. In particular in table (1) we have reported all the different estimates with a number of iterations n leading to an error $|E_n| > 0.1$.

x_0	Error E_0	Steps n	Error E_n	Exponent
0.202	0.001	9	0.2562	0.6162
0.202	0.0001	11	-0.1235	0.6472
0.202	0.00001	15	0.2573	0.6770
0.347	0.001	7	-0.1233	0.6878
0.347	0.0001	11	-0.1856	0.6842
0.347	0.00001	15	0.3139	0.6903
0.869	0.001	8	0.2507	0.6905
0.869	0.0001	10	0.1404	0.7249
0.869	0.00001	13	0.1143	0.7188

Table 1: Estimate of Lyapunov exponent for different values of x_0 and E_0

Thanks to table 1 we have an intuition that $\lambda(x_0) = \ln 2 \approx 0693$ for all $x_0 \in [0, 1]$. Our intuition is confirmed by the following theorem with the introduction of a little limitation which will be properly discussed.

Theorem 1. *Consider the discrete dynamical system $h_4 : [0, 1] \to [0, 1]$. The Lyapunov exponent in this case results equal to $\lambda(x_0) = \ln 2$ for almost any $x_0 \in [0, 1]$*

A possible proof of the Theorem is Provided by **Wolf et al.**[9] To better understand the meaning of almost any point $x_0 \in [0, 1]$ in the previous theorem we have to observe the following.

Remark 2. *A particular computation of $\lambda(x_0)$ holds when x_0 belongs to an orbit of period m (i.e. $x_m = f^m(x_0) = x_0$) indeed we get:*

$$\lambda(x_0) = \frac{1}{m} \sum_{k=1}^{m} \ln |f'(x_{k-1})|$$

In particular if x_0 is a fixed point we have: $\lambda(x_0) = \ln|f'(x_0)|$

This formula is useful for the logistic map previous introduced since $x_0 = 0$ is a fixed point and provides this value of $\lambda(0) = |f'(0)| = \ln 4 > 0$ which also shows that $x_0 = 0$ is an unstable fixed point. So what we have affirmed before that is: $\lambda(x_0) = \ln 2$ for all $x_0 \in [0, 1]$ has now to be re formulated in more formal contest. Indeed we observe that $\lambda(x_0) = \ln 4$ for those points whose orbit eventually ends in the fixed point 0 (for example $x_0 = 1$). Note that this is a dense subset of $[0, 1]$. Finally we observe that in this example we are delaying with a system showing a positive value of Lyapunov exponent (and so sensitive dependence on initial conditions) which is also chaotic. Note that sensitive dependence on initial condition is only a necessary condition for a system to be chaotic as we will see in next example.

Example 2. Let's now consider the logistic map when the parameter $a > 4$ i.e. $h_a(x) = ax(1-x)$ and $h_a : \mathbb{R} \to \mathbb{R}$. In this case the system clearly shows sensitive dependence on initial conditions and its Lyapunov would be greater than zero. Nevertheless we cannot affirm that this system has a chaotic behaviour since periodic orbits are not dense in $\mathbb{R}$ indeed it is always possible to find an $I \subset \mathbb{R}$ which does not contain any point belonging to a periodic orbit. In this case it is interesting to note that there exists a Cantor-like set R defined as :

$$R = \{x \in [0, 1], \;\; h_a^k(x) \in [0, 1] \; \forall k \in \mathbb{N}\}.$$

This set is invariant by definition and repulsive (i.e. ${h_a}^k(x) = -\infty$ as $k \to \infty$) and the dynamic of h_a restricted to R is chaotic.

2.3 The Maximum Lyapunov Exponent

Given a vector valued discrete dynamical system of the form:

$$\mathbf{X}_{t+1} = F(\mathbf{X}_t) \tag{6}$$

where $t \in \mathbb{N}$, $F : \mathbb{R}^m \to \mathbb{R}^m$ is a nonlinear map and $F \in C^1(\mathbb{R}^m)$ and
$m > 1$. Analogously to the scalar case we can consider a generic $\mathbf{X}_0 \in \mathbb{R}^m$ and a perturbation on it equal to $\mathbf{E}_0 \in \mathbb{R}^m$. We will indicate Jacobian matrix of the map F evaluated in $\mathbf{X}_t$ as $\mathbf{J}_t = DF(\mathbf{X}_t)$. Applying a first order approximation and the chain rule to the difference $F^t(\mathbf{X}_0 + \mathbf{E}_0) - F^t(\mathbf{X}_0)$ we get after some computation:

$$\|\mathbf{E}_t\| = \left\|F^t(\mathbf{X}_0 + \mathbf{E}_0) - F^t(\mathbf{X}_0)\right\| = \|\mathbf{T}_t\mathbf{E}_0\| + o(\|\mathbf{E}_0\|) \tag{7}$$

where we have defined:

$$\mathbf{J}_{t-1} \ldots \mathbf{J}_0 = \mathbf{T}_t$$

Calling $\mathbf{E}_{0i}$ the eigenvector corresponding to the i-th largest eigenvalue of matrix $\mathbf{T}_t'\mathbf{T}_t$, which will be indicate as $\nu_i(\mathbf{T}_t'\mathbf{T}_t)$, we can write:

$$\|\mathbf{E}_t\|^2 = (\mathbf{T}_t\,\mathbf{E}_{0i})'(\mathbf{T}_t\,\mathbf{E}_{0i}) = |\nu_i(\mathbf{T}_t'\mathbf{T}_t)|\,\|\mathbf{E}_{0i}\|^2 \quad as\;\|\mathbf{E}_{0i}\| \to 0 \tag{8}$$

Analogously to the scalar case we can define the logarithm of the ratio between the error at time t and the error at initial time getting:

$$\frac{1}{t}\ln\frac{\|\mathbf{E}_t\|}{\|\mathbf{E}_{0i}\|} = \frac{1}{2t}\ln\frac{\|\mathbf{E}_t\|^2}{\|\mathbf{E}_{0i}\|^2} = \frac{1}{2t}\ln|\nu_i(\mathbf{T}_t'\mathbf{T}_t)| \quad as\;\|\mathbf{E}_{0i}\| \to 0$$

Finally we can provide the following definition of the set of Lyapunov exponents

Definition 3. *Given a a vector valued discrete dynamical system as presented in (4). The set of Lyapunov Exponents* $\{\lambda_1(\boldsymbol{X}_0)\ldots\lambda_m(\boldsymbol{X}_0)\}$ *is given by:*

$$\lambda_i(\boldsymbol{X}_0) = \lim_{t\to\infty} \frac{1}{2t} \ln(|\nu_i(\boldsymbol{T}_t'\boldsymbol{T}_t)|) \quad 1 \le i \le m \tag{9}$$

The Largest eigenvalue $\lambda_1(\boldsymbol{X}_0)$ *is called* ***Maximum Lyapunov Exponent*** *and will be shortly indicated as* ***MLE***

Similarly to the scalar case positive values of Lyapunov Exponents will be associated to that directions (in phase space) where two arbitrarily close trajectories at initial time will diverge, on the contrary negative exponents will be related to directions of contraction. We focus our attention on maximum Lyapunov exponent since, if positive, shows that that the analyzed system will present a sensitive dependence on initial conditions along at least one direction.

3 MLE for noisy Discrete Dynamical systems

In previous section we have considered only pure deterministic Discrete dynamical system. From now on we will present discrete dynamical system affected by some random noise and we will discuss for this type of systems the previous introduced notion of **Maximum Lyapunov exponent**. In particular let's consider:

$$x_t = f(x_{t-1}, .., x_{t-m}) + u_t \tag{10}$$

with $t \ge m$, $f\colon \mathbb{R}^m \to \mathbb{R}$ is a non linear and smooth map (*i.e.* $f \in C^\infty(\mathbb{R}^m)$), $\{u_t\}_{t=m}^{\infty}$ is a sequence of *i.i.d* (independent and identically distributed) random variables such that $\mathrm{Var}[u_t] = \sigma^2$ and $\mathbb{E}[u_t]= 0$ $\forall t \ge m$.
We introduce $F\colon \mathbb{R}^m \to \mathbb{R}^m$ such that (10) can be rewritten as follow:

$$\mathbf{X}_t = F(\mathbf{X}_{t-1}) + \mathbf{U}_t \tag{11}$$

where $\mathbf{X}_t = (x_t, ..., x_{t-m+1})' \in \mathbb{R}^m$, $\mathbf{X}_{t-1} = (x_{t-1}, ..., x_{t-m})' \in \mathbb{R}^m$ and $\mathbf{U}_t = (u_t, 0, .., 0)' \in \mathbb{R}^m$. We can define the Jacobian of the map F in (10) evaluated at $\mathbf{X}_t$ as $\mathbf{J}_t = DF(\mathbf{X}_t)$. Specifically, we have:

$$\mathbf{J}_t = \begin{pmatrix} \frac{\partial f}{\partial x_{t-1}} & \frac{\partial f}{\partial x_{t-2}} & \cdots & \frac{\partial f}{\partial x_{t-(m-1)}} & \frac{\partial f}{\partial x_{t-m}} \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix} \tag{12}$$

Also for systems presented in (11) we can relay on the previous definition of MLE which we will report below:

$$\lambda_1(\mathbf{X}_0) = \lim_{t\to\infty} \frac{1}{2t} \ln(|\nu_1(\mathbf{T}_t'\mathbf{T}_t)|) \tag{13}$$

Calling $\mathcal{L} = \lim_{t\to\infty}(\mathbf{T}_t'\mathbf{T}_t)^{1/2t}$ we can observe that $\nu_1(\mathcal{L}) = \lambda_1(\mathbf{X}_0)$. Note that the eigenvectors of matrix $\mathcal{L}$ will depend on the particular realization of random vectors $\{\mathbf{U}_t\}_{t=m}^{\infty}$ and so will be different for different realization of the system even though the eigenvalues are constant and remain always the same **Eckman e Ruelle**[3]. To ensure the existence almost surely[1] of the MLE in definition (13) to a constant value and its independence from the initial value[2] $\mathbf{X}_0$ we have to introduce this set of assumptions related to system (11).

Assumption 1. $\{\boldsymbol{U}_t\}_{t=m}^{\infty}$ *is a set of i.i.d. random vectors as previous defined.*

[1]Note that in this case the limit in definition (13) is a limit of random variables
[2]$\lambda_1(\mathbf{X}_0) = \lambda_1 \ \forall \mathbf{X}_0 \in \mathbb{R}^m$

Assumption 2. *There exists an invariant set $\mathcal{A}$ and a unique Borel probability measure μ on $(\mathbb{R}^m, \mathcal{B})$ such that*

$$\mu(B) = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} I_B(\boldsymbol{X}_t)$$

for all $B \in \mathcal{B}$ and $\boldsymbol{X}_0 \in \mathcal{A}$. (Note that $I_B(x) = 1$ if $x \in B$, otherwise $I_B(x) = 0$)

Assumption 3. *X_0 is randomly sampled from the distribution μ*

Assumption 4. $\displaystyle\int_A \ln(max(||DF(X)||, 0)) < \infty$

A detailed analysis on this set of assumptions is presented by **Nychka et al.**[6] We just note that the the set of previous assumptions generate a time series which is **Ergodic** and **Stationary**.

3.1 A weaker definition of Chaos

In next sections we will fit an autoregressive model affected by some random noise to a finite time series (i.e the financial one). Note that the presence of noise in this type of contest results fundamental to properly describe the market's daily fluctuation and a pure deterministic dynamical system would result too restrictive. Moreover since we will not know a priori the map generating the data (actually we would reconstruct it trough a feed forward neural network) checking topological property of such a function would result almost impossible. Due to that we will rely on a broad definition of Chaos in which the first and second requirement of the previous one (density of periodic orbits and topologically transitivity) are relaxed. This definition of Chaos, firstly introduced by Ruelle, can be reformulated in our contest as follow:

Definition 4. *A noisy vector discrete dynamical system $F : I^m \subseteq \mathbb{R}^m \to I^m \subseteq \mathbb{R}^m$ is said to be* ***Chaotic*** *if it shows sensitive dependence on initial conditions (i.e. its Maximum Lyapunov exponent $\lambda_1(\boldsymbol{X}_0) > 0 \ \forall \boldsymbol{X}_0 \in I^m$)*

Note that this type of definition holds also for system affected by noise once the definition of Maximum Lyapunov exponent is well posed. So from now on we will assume that the dynamical system will generate a time series such that assumptions 1), 2), 3) holds.

4 Chaos detection Methodology

In this section we will consider a finite sequence of real numbers assuming to be generated by an autoregressive non linear model affected by some random noise (*i.e* the financial time series) as presented in (10) with the introduction of a parameter L taking account of the time granularity. Then we will reconstruct the unknown map generating the data trough a Feed-Forward Neural Network to finally provide the estimate of Maximum Lyapunov exponent.

4.1 Estimate of the MLE

Let $\{x_t\}_{t=1}^{T}$ be a random scalar sequence generated by the following:

$$x_t = f(x_{t-L}, .., x_{t-mL}) + u_t \tag{14}$$

Which is actually the same presented in (10) with the introduction of the time granularity $L \in \mathbb{N}_+$ and $mL \le t \le T$. As seen before we can introduce $F \colon \mathbb{R}^m \to \mathbb{R}^m$ and rewrite (14) as follow:

$$\mathbf{X}_t = F(\mathbf{X}_{t-L}) + \mathbf{U}_t \tag{15}$$

where $\mathbf{X}_t = (x_t, ..., x_{t-(m-1)L})' \in \mathbb{R}^m$, $\mathbf{X}_{t-L} = (x_{t-L}, ..., x_{t-mL})' \in \mathbb{R}^m$, $\mathbf{U}_t = (u_t, 0, .., 0)' \in \mathbb{R}^m$. A crucial point is how to estimate the unknown map f in (14) from the observed data. To do so, we rely on a single hidden-layer feed-forward neural network with single output. We use the estimated $\hat{f}$ provided by the Neural Network in Equation (12) in order to get the following estimate of the Jacobian Matrix.

$$\hat{\mathbf{J}}_t = \begin{pmatrix} \frac{\partial \hat{f}}{\partial x_{t-L}} & \frac{\partial \hat{f}}{\partial x_{t-2L}} & \dots & \frac{\partial \hat{f}}{\partial x_{t-(m-1)L}} & \frac{\partial \hat{f}}{\partial x_{t-mL}} \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix} \tag{16}$$

To ensure the existence almost surely of maximum Lyapunov exponent we will assume that the previous introduced assumptions 1), 2), 3) hold for system (15) and, based on definition (13), we will provide the following estimate of Maximum Lyapunov exponent:

$$\hat{\lambda}_M = \frac{1}{2M} \ln \nu_1((\hat{\mathbf{T}}_M \mathbf{U}_0)(\hat{\mathbf{T}}_M \mathbf{U}_0)') \tag{17}$$

Where M is the number of observation we consider to provide the estimation and it is usually smaller than the sample size T introduced before. In this work we will adopt $M = \lfloor T^{2/3} \rfloor$ (i.e. the inferior integer part of $T^{2/3}$) as proposed by **BenSaida and Litimi**[2], since it is a good trade-off between the accuracy of the result and computational effort: clearly, taking $M = T$ yelds a better accuracy but requires more computations, while with a too small value of M we would provide a very poor estimation. The matrix $\hat{\mathbf{T}}_M$ is defined as $\hat{\mathbf{T}}_M = \prod_{t=1}^{M} \hat{\mathbf{J}}_{M-t}$, while $\nu_1((\hat{\mathbf{T}}_M \mathbf{U}_0)(\hat{\mathbf{T}}_M \mathbf{U}_0)')$ represent the largest eigenvalue of the matrix $(\hat{\mathbf{T}}_M \mathbf{U}_0)(\hat{\mathbf{T}}_M \mathbf{U}_0)'$, finally $\mathbf{U}_0 = (1, 0, ..., 0)'$ is a unit vector used to reduce the bias during the computation. In appendix A we have reported the asymptotic properties of the estimator $\hat{\lambda}_M$.

4.2 Neural Network learning process of the unknown map

To estimate the Jacobian matrix in Equation (16), it is first necessary to estimate f in Equation (14). To do so we adopted a Feed Forward Neural Network with one hidden layer and one single output. The extension to a more depth Neural Network is still an open problem and no literature have yet analyzed it, this is due the definition and the computation of the Jacobian matrix, which would otherwise result too complicated. Roughly speaking, the functioning of a FFNN is a sequence of matrix and functions multiplications, in which the units called *neurons* are linked among them a *layering fashion*, where the neurons in a lower layer receives inputs from the neurons in the previous layer. Each unit contains a function called *activation function* that will produce the output of such neuron by evaluating the output of the previous layer, in the form of **linear combination** of the outputs of the previous layer's neurons, and summing a *bias* term. Basically, the FFNN will train itself by calibrating the weights of such combinations using optimization algorithm such as gradient descent. A brief scheme of the FFNN presented in Figure (2): at first we do have a layer dedicated to receive the m-dimensional input vector, said $(x_{t-L}, ...x_{t-mL})$, this input vector is propagated using the weights $\beta_{i,j}$ with $i = 1, \dots, m$ and $j = 1, \dots, q$ through a q-dimensional hidden layer in which each neuron has an activation function equal to Ψ (typically non linear and S shaped) and bias equal to $\beta_{0,j}$ with $j = 1, \dots, q$. The output neuron uses only the identity map in order to preserve the superposition effect coming from the hidden layer. Finally the analytical form of the output of the Neural Network is given by :

$$x_t = \hat{f}(x_{t-L}, .., x_{t-mL}, \theta) = \alpha_0 + \sum_{j=1}^{q} \alpha_j \Psi(\beta_{0j} + \sum_{i=1}^{m} \beta_{ij} x_{t-iL}) \tag{18}$$

The vector θ contains the FFNN weights, which are estimated minimizing the **Squared Noise**:

$$S(\theta) = \sum_{t=mL+1}^{T} (x_t - \hat{f}(x_{t-L}, .., x_{t-mL}, \theta))^2 = \sum_{t=mL+1}^{T} \hat{u_t}^2 \tag{19}$$

Due to the non linearity of activation function Ψ, also the mean square minimization process presented in Equation (19) results non linear.

5 Experimental Results

We implemented **by scratch** the methodology discussed in the previous section: we started retrieving historical data of time series from *Yahoo! Finace*, then we scaled the data and fed them to our Feed-Forward Neural Network. Intuitively, how are the data fed to the architecture? First of all we granularize them with granularity of L, then we split them in sequential windows of data of dimension m, using q neurons to perform the forecast of the value immediately after a given data window: due to this kind of data processing, the triplet (L, m, q) assumes a crucial role in our tractation and we will need to evaluate the methodology for different choices of it. We used the **Tensorflow** package of the programming language **Python** to code our FFNN, with one single hidden-layer with q neurons. We used $\Psi(x) = tanh(x)$ as activation function: thanks to this choice together with the result of **Kidger and Lyons**[8], we have a theoretical guarantee about the goodness of the approximation that we will obtain from the FFNN. In Figure 3 is reported a simple scheme of how our code works[3] , we will explore multiple possibilities for the parameters (L, m, q), estimating multiple values of the MLE for a given time serie, as provided by **BenSaida**[1].

5.1 Chaos Detection in Commodities' stock price

In order to provide an example of chaotic behaviour in financial time series, we decided to bring our results regarding two of the most important commodities in the world: **gold (stock code GC=F) and crude oil (stock code CL=F)**. On both of them we performed the procedure aforementioned, cycled over multiple choices of the (L, m, q) parameters.

FFNN goodness of fit

We start anayzing **how good** is the estimation of the FFNN. The first, intuitive way, in which we can do this is to have a glance at the true data in comparison with the FFNN fit of the data:

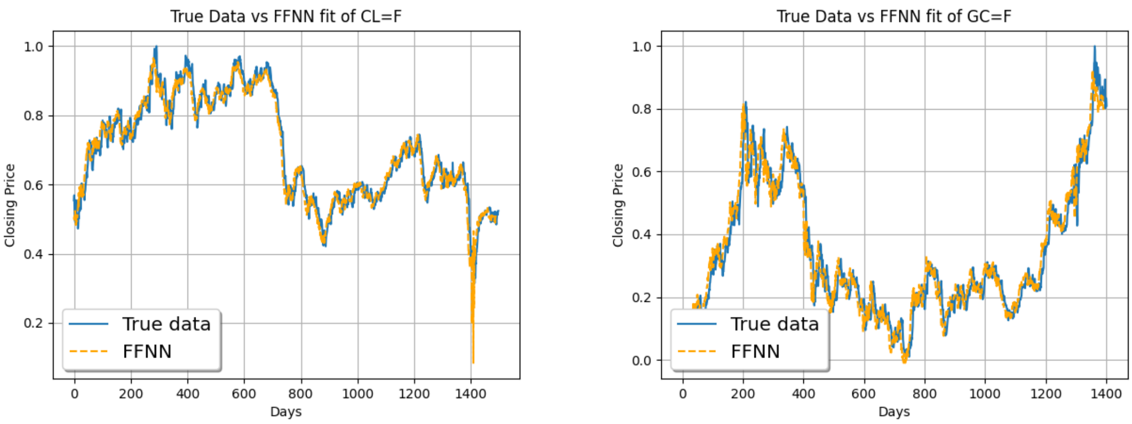

Figure 3: FFNN fit on the true data

[3]Our code is publicly available at https://bit.ly/lyapunov

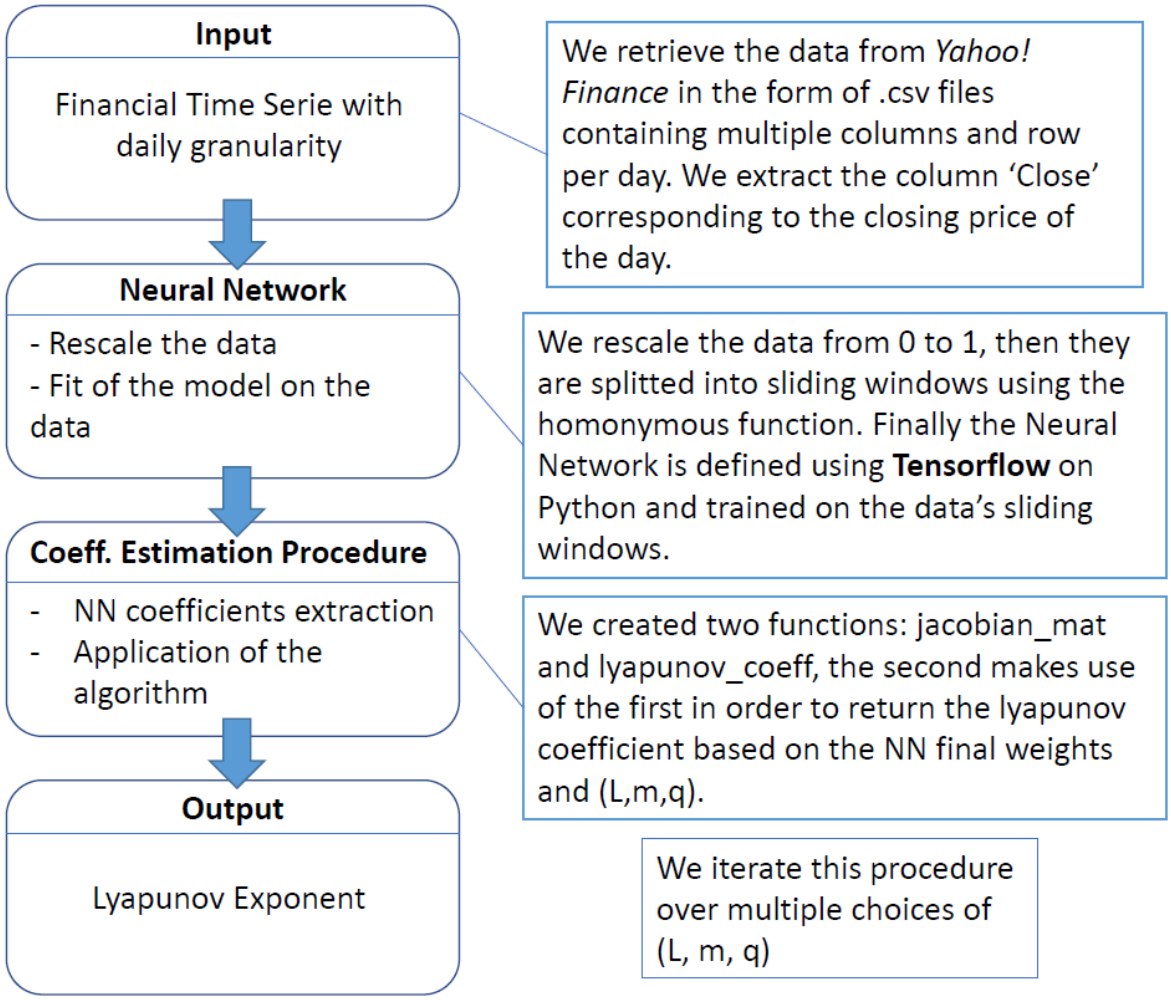

Figure 2: Scheme of our implementation

In Figure 4, we can observe how the orange line which represents the prediction of the FFNN fits very closely the blue line, nominally the true data on which we are working (closing prices of our two commodities).

But how can we quantify this goodness of fit? We mentioned the *Squared Noise* as the quantity to minimize, we can easily obtain the plots of it along the possible choices of the dimensions (L, m, q):

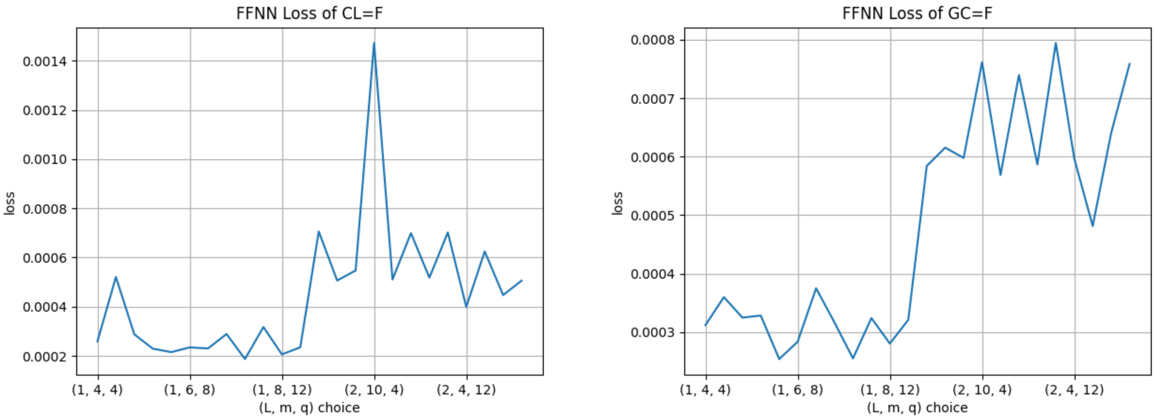

Figure 4: FFNN final losses for multiple choices of (L,m,q)

As we can see in Figure 5, the *loss* of our FFNN is always below 0.0015 for Crude Oil while it is even always below 0.0008 for Gold. Such small values of the losses, evaluated for many possibilities of the triplet (L, m, q) are the best index to quantify how good the FFNN is fitting the right weights on its neurons, we are now pretty confident about the weights that we are gonna plug into the MLE estimation methodology.

MLE estimation

The package Tensorflow of Python let us extract the estimated weights from our Neural Network. With a simple command we can extract the coefficients mentioned in Section 4.2, we can use them to evaluate the equation (18) and obtain an approximated analytical form of the function f introduced in (14). We defined two functions: one is able to compute the Jacobian Matrix in the form (16), while the other makes use of the Jacobian matrices computed by the latter in order to produce the numerical estimation of the MLE of the time serie.

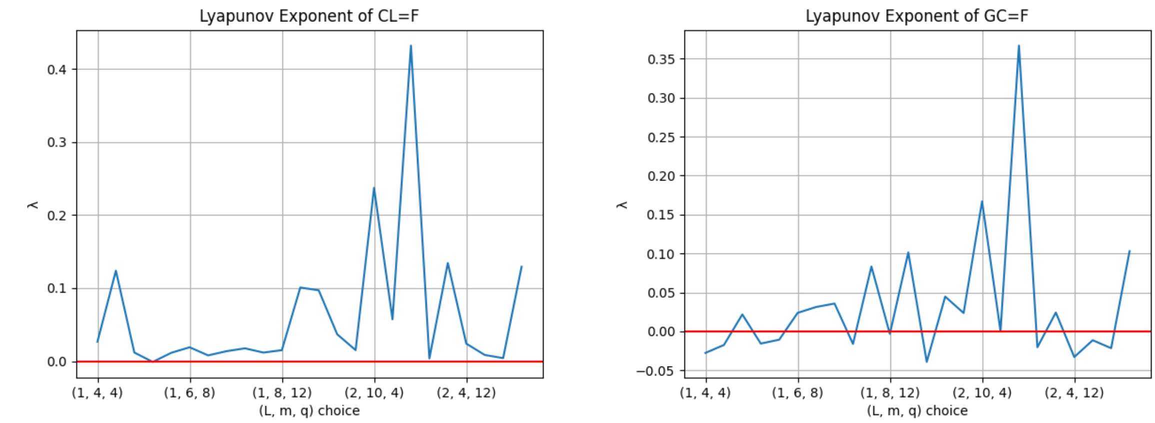

Figure 5: Estimated Lyapunov Exponent for multiple choices of (L,m,q)

As we can see in Figure 6, the MLE of the two commodities present different behaviours: while the one of Crude Oil is greater or equal than 0 for all the choices of the triplet (L,m,q), the one of Gold is sometimes negative. How can we explain this behaviour and classify this serie as chaotic or not? The authors of [1] suggest to verify if at least one Lyapunov Exponent is greater than 0, this would be a sufficient condition for a chaotic behaviour. Since in our two cases most of the exponents are positive, we can conclude that both the commodities show a chaotic behaviour. We observe that in some cases this conclusion would result more difficult,

BenSaida [1] provides some **statistical test** aimed to verify the chaotic behavior hypothesis, we reported this auxiliary methodology in Appendix A.

6 Summary

We first introduced the financial time series as discrete dynamical systems, providing the related notions of Chaos and Lyapunov Exponent. Then, we reorganized the methodology proposed in [1], highlighting the mathematical properties behind it and replicating it in Python code starting from scratch. We tested this approach on the stock prices of two of the main commodities in the financial market, this is an alternative test case from the one proposed by the authors of [1] related to *Volatility Indices*, in the end we found a chaotic behavior in them both. The code related to this project is yet modularized and *"ready to use"*, in the sense that it is sufficient to provide an input time serie and the code will automatically produce the Lyapunov Exponents for multiple choices of (L, m, q), in line with what was proposed in the original work. Thinking about possible generalizations and further developments of this methodology, a natural extension could be i.e. the case of *portfolio analysis*, in which multiple stocks are bought together and we aim to explore the chaotic behavior of the economic return of the correspondent portfolio. We also believe that many chaos detection techniques will make use of Artificial Neural Networks in future due to their approximation properties and analytical form.

Appendices

A Asymptotic distribution of MLE estimator

The estimation of the Maximum Lyapunov Exponent will not clearly correspond to the true value of it, we can only try to minimize the noise around this estimation. This means that the true value of this exponent will be somewhere around our estimation, how can we quantify the distribution of probability of this? We will present the main asymptotic results related to the estimate of MLE at time M (i.e. the previous introduced $\hat{\lambda}_M$) and how to test the chaotic hypotheses for a noisy and finite time series as presented before.

Theorem 2. *The random variable $\hat{\lambda}_M$,presented in (17), has the following asymptotic distribution:*

$$\sqrt{M}(\lambda - \hat{\lambda}_M) \sim \mathcal{N}(0, \Sigma) \quad \text{as } M \to \infty \tag{20}$$

where Σ is the Heteroskedasticity and Autocorrelation Consistent (HAC) matrix and it is defined in assumption 2

Assumption 1. *For some $\psi > 0$ and $1 \leq i \leq m$:*

$$\max |F_{i,t-1}(J_{M-1}, \ldots, J_o)| = O(M^{\psi}) \quad 1 \leq t \leq M$$

$$F_{i,t-1}(J_{M-1}, \ldots, J_o) = \frac{\partial \ln(\nu_i(\boldsymbol{T}_t' \boldsymbol{T}_t))}{\partial F(\boldsymbol{X}_{t-1})}$$

Assumption 2. *The following limit exists and it is positive and finite*

$$\Sigma = \lim_{M \to \infty} Var\left[\frac{1}{\sqrt{M}} \sum_{t=1}^{M} \eta_t\right]$$

where:

$$\eta_t = \xi_t - \lambda_1 \qquad \xi_t = \frac{1}{2} ln\left(\frac{\nu_1(\hat{\boldsymbol{T}}_t' \hat{\boldsymbol{T}}_t)}{\nu_1(\hat{\boldsymbol{T}}_{t-1}' \hat{\boldsymbol{T}}_{t-1})}\right)$$

For the proof of Theorem 1 and a propose of a consistent estimate of Σ see **Shintani et al. (2003)** [4]. Thanks to Theorem 1 and an appropriate estimate of the covariance matrix Σ we can define the statistics $\hat{Z}$ as:

$$\hat{Z} = \frac{\hat{\lambda}_M}{\sqrt{\frac{\hat{\Sigma}}{M}}} \sim \mathcal{N}(0,1) \quad as\ M \to \infty \tag{21}$$

Note that it is possible to provide some statistical test to verify or not the chaotic hypotesis specifically we will test: $H_0 : \lambda < 0$ against $H_1 : \lambda > 0$. We finally note that to ensure the validity of Thereom 1 other technical assumption are required both on the Neural network and the financial time series, the reader is left to [4].

References

[1] Litimi, Houda BenSaïda, Ahmed Belkacem, Lotfi Abdallah, Oussama. (2019). Chaotic behavior in financial market volatility. Journal of Risk. 21. 27-53. 10.21314/JOR.2018.400.

[2] BenSaïda, A. and Litimi, H. (2013). High level chaos in the exchangeand index markets. Chaos, Solitons Fractals, 54:90–95.

[3] Eckmann, J .P. and Ruelle, D. (1985) Ergodic theory of chaos and strange attractors. Rev. Mod. Phys., 57, 617-656.

[4] Shintani, M. and Linton, O. (2004). Nonparametric neural network estimation of Lyapunov exponents and a direct test for chaos. Journal of Econometrics, 120(1):1–33.

[5] Tomarelli, F. Salinelli, E. (2013). Discrete Dynamical Models. Springer Verlag.

[6] Nychka, D., Ellner, S., McCaffrey, D., and Gallant, A. R. (1992). Finding chaos in noisy systems. Journal of the Royal Statistical Society. Series B, 54:399–426.

[7] P. Kidger, T. Lyons (2020). Universal Approximation with Deep Narrow Networks. COLT 2020

[8] Chaos and Fractals (2004). Peitgen, H. Jürgens, H. Saupe, D. Springer.

[9] Wolf, A., Swift, J. B., Swinney, H. L., and Vastano, J. A. (1985). Determining Lyapunov exponents from a time series. Physica D: Nonlinear Phenomena, 16(3):285–317.