Contents

1. Entropy-Complexity

Given the time series data $Y = (y_1, ..., y_N)$, one can estimate the associated PDF as $P = (p_1, ..., p_M)$, where M is the amount of unique states the system can have. This allows the notion of Shannon entropy to be introduced:

$$H(P) = -\sum_{j=1}^{M} p_j \log(p_j).$$

Then, the statistical complexity measure can also be calculated using the following rule:

$$C(P) = Q_{\mathcal{J}}(P, P_U) S(P),$$

where $Q_{\mathcal{J}}(P, P_U)$ is normalized Jensen-Shannon divergence between P and the PDF of a uniform distribution

$$Q_{\mathcal{J}}(P, P_U) = \frac{\mathcal{J}(P, P_U)}{\mathcal{J}_{\max}}$$

and $S(P)$ is normalized Shannon entropy given by:

$$S(P) = \frac{H(P)}{H(P_U)} = \frac{H(P)}{\log M}.$$

Note that

$$\mathcal{J}(P_1, P_2) = H\left(\frac{P_1 + P_2}{2}\right) - \frac{1}{2}(H(P_1) + H(P_2)),$$

therefore,

$$\begin{aligned}
\mathcal{J}_{\max} = \mathcal{J}(P_U, P_C) = H\left(\frac{P_U + P_C}{2}\right) - \frac{1}{2}(H(P_U) - H(P_C)) = \\
= H\left(\frac{P_U + P_C}{2}\right) + \frac{1}{2}\log M - \frac{1}{2} \cdot 0 \\
= \frac{1}{2}\log M - \frac{M-1}{2M}\log\left(\frac{1}{2M}\right) - \left(\frac{1}{2M} + \frac{1}{2}\right)\log\left(\frac{1}{2M} + \frac{1}{2}\right).
\end{aligned}$$

2. Largest Lyapunov Exponent

Def. 1 The largest (senior) Lyapunov exponent is a measure of the exponential speed at which trujectories diverge.

It can be calculated in the following manner:

1. Given a time series $Y = (y_1, ..., y_N)$, conduct a reconstruction $z_i = (y_i, ..., y_{i+m-1})$.
2. Select the nearset neighbours

$$j_i = \{j : \varepsilon_{\min} < \rho(z_i, z_j) < \varepsilon_{\max}, |i - j| > \varepsilon_t\}.$$

From the set of points satisfying these conditions k are selected which will be denoted as $N_i = \{z_{j_1}, ..., z_{j_k}\}$.

3. For each z_i and each of its neighbours $z_j \in N_i$ the evolution of distance between them over time is computed:

$$d_{ij}(k) = \|z_{i+\tau} - z_{j+\tau}\|, \ \ \tau = 0, 1, ..., \text{max\_time}.$$

4. For each time lag τ the average logarithmic divergence is calculated:

$$S(\tau) = \frac{1}{M'} \sum_{i=1}^{M\prime} \frac{1}{|N_i|} \sum_{z_j \in N_i} \ln d_{ij}(\tau),$$

where M' is the number of points z_i for which enough neighbourse are found, i.e. $|\{i : |N_i| \geq k\}|$. This function represents the average log distance between trajectories after τ steps.

5. The largest Lyapunov exponent can be exstimated as follows:

$$\lambda_{\max} = \frac{S(\tau_2) - S(\tau_1)}{(\tau_2 - \tau_1)\Delta t},$$

where $[\tau_1, \tau_2]$ is the range of lags over which linear growth of S (tau) can be observed. However, in practice a linear estimation algorithm is ofter used:

$$S(\tau) \approx a + \lambda_{\max} \cdot \tau \Delta t.$$

3. Lyapunov Spectrum

3. 1. Local linear maps method

Assume that locally $Y_{i+1} \approx A_i B_i + b_i$.

Let

$$\begin{pmatrix} Y_{j_1+1}^T \\ Y_{j_2+1}^T \\ \vdots \\ Y_{j_k+1}^T \end{pmatrix} = \begin{pmatrix} Y_{j_1}^T & 1 \\ Y_{j_2}^T & 1 \\ \vdots & \vdots \\ Y_{j_k}^T & 1 \end{pmatrix} \times \begin{pmatrix} A_i^T \\ b_i^T \end{pmatrix}.$$

Step-by-step:

1. Reconstruction: $\{x_i\} \to \{Y_i\}$.
2. Search for the k nearest neigbours.

$$N_i = \left\{Y_{j_1}, ..., Y_{j_k}\right\},\ Y_{j_k} \in \left\{Y_j : \quad \|Y_i - Y_j\| < \varepsilon, |i - j| > \varepsilon_t\right\}.$$

3. $\forall Y_j \in N_i : Y_{j+1} \approx A_i Y_j + b_i$. Then, minimize MSE:

$$\sum_{Y_j \in N_i} \|Y_{j+1} - A_i Y_j - b_i\|^2 \to \min_{A_i, b_i}$$

4. Form an orthonormal basis:

    4.1. Choose an initial point Y_{i_0} on the trajectory.

    4.2. Initialize an orthonormal basis: $Q_0 = [q_1^0, q_2^0, ..., q_m^0],\ Q_0^T Q_0 = I$.

    4.3. Initialize the accumulators for log stretching coefficients:

$$L_j = 0,\ j = 1, ..., m.$$

5. Find A_{i_n} for each Y_{i_n} (see step 3) and apply it to the current basis:

$$V_{n+1} = A_{i_n} Q_n.$$

Then, every T steps (or as needed) use QR decomposition $V_{n+1} = Q_{n+1} R_{n+1}$ to get a new orthonormal basis Q_{n+1}, and an upper-triangular matrix R_{n+1}.

6. Accumulate the exponents: $L_j = L_j + \ln(R_{n+1})_{jj}, j = 1, ..., m$.
7. Go to the next point: $i_{n+1} = i_n + 1$.

7. Calculate the Lyapunov exponents:

$$\lambda_j = \frac{L_j}{N_{\text{iter}} \Delta t}, \; j = 1, ..., m.$$

Note that A_i is a jacobian matrix of our dynamical system.

3. 2. Wolf method

Step-by-step:

1. Reconstruction $\{x_i\} \to \{Y_i\}$.
2. Initialization. Let $Y_0 = Y_i$, define an orthonormal basis for it: $q_1^0 = [1, 0, ..., 0]^T$, $q_2^0 = [0, 1, 0, ..., 0]^T$, ...
3. Take Y_k and evolve it in time by $\left\{q_1^k, q_2^k, ..., q_m^k\right\}$, $Y_{k+1} = Y_{i+1}$, then $\forall q_i$:

$$\|Y_j - Y_k\| < \varepsilon_{\max}, |j - k| > \varepsilon_t$$

$$\delta = Y_j - Y_k : \alpha = \arccos\left(\frac{\delta \cdot q_i^k}{\|\delta\| \quad \|q_i^k\|}\right) < \varepsilon_{\min}$$

4. $v_j = Y_{j+\Delta} - Y_{k+\Delta} \Rightarrow \{v_1, v_2, ..., v_m\}$.
5. For $j = \overline{1, ..., m}$:

   for $j = 1$: $u_1 = \frac{v_1}{\|v_1\|}$, $L_1^k = \ln\|v_1\|$

   for $j = 2$: $w_2 = v_2 - (v_2 \cdot u_1)u_1$, $u_2 = \frac{w_2}{\|w_2\|}$. $L_2^k = \ln\|w_2\|$ and so on.
6. $q_1^{k+1} = u_1, ..., q_m^{k+1} = u_m$.

   for j:

$$w_j = v_j - \sum_{i=1}^{j-1} (v_j u_i) u_i$$

$$u_j = \frac{w_j}{\|w_j\|}$$

$$L_j^k = \ln\|w_j\|.$$

Then,

$$\lambda_j = \sum_{k=1}^{\text{max\_iter}} \frac{L_{jk}^k}{\text{max\_iter} \cdot \Delta \cdot \Delta t}.$$

3. 3. How to choose the size of reconstruction

False nearest neigbours approach. For $m = m_{\min}, ..., m_{\max}$.

1. Sample z_i – vectors of size m.
2. Find the number of nearset neigbours:

$$\mathrm{NN}_i = |\{z_j : \|z_i - z_j\| < \varepsilon\}|, \quad \mathrm{NN} = \sum_i \mathrm{NN}_i$$

3. Sample $\tilde{z}_i$ – vectors of size $m + 1$.
4. Find the number of false nearest neigbours:

$$\mathrm{FNN}_i = |\{\tilde{z}_j : \|z_i - z_j\| < \varepsilon_1, \|\tilde{z}_i - \tilde{z}_j\| \geq \varepsilon_2, |i - j| > \tau\}|$$

and $\mathrm{FNN} = \sum_i \mathrm{FNN}_i$.

5. The optimal m is the one that achieves the preset FNN to NN ratio (typically 1% to 5%) first.

4. Kolmogorov-Sinai Entropy

How to calculate the K-S entropy:

1. Subdivide the phase space into cells A_i with side ε.
2. Take $\rho_i = \mu(A_i)$ – measures of A_i and the $f^{-k}(A_i)$ – the set of all points that arrived to A_i in k steps.
3. Take

$$A_i^{(1)} = A_i,$$
$$A_{i_1 i_2}^{(2)} = A_{i_1} \cap f^{-1}\left(A_{i_2}\right),$$
$$A_{i_1 i_2 i_3}^{(3)} = A_{i_1} \cap f^{-1}\left(A_{i_2}\right) \cap f^{-2}\left(A_{i_3}\right)$$

etc. up to $A_{i_1 \ldots i_k}^{(k)}$.

4. Calculate

$$H^{(k)} = -\sum_{i_1, \ldots, i_k} \mu\left(A_{i_1 \ldots i_k}^{(k)}\right) \log\left(\mu\left(A_{i_1 \ldots i_k}^{(k)}\right)\right)$$

5. The K-S entropy would be:

$$K(\mu) = \lim_{\varepsilon \to 0} \lim_{k \to +\infty} \left(H^{(k+1)} - H^{(k)}\right).$$

Interpretation:

- $K(\mu) > 0$ is indicative of chaos;
- $K(\mu) = 0$ indicates that the system is deterministic.

5. Fractal and topological dimensions. Fractal dimension approximations

5. 1. Topological dimension

Topological dimension is denoted as d_T. Topological dimensions of several objects: $d_T(\emptyset) = -1$, $d_T(\text{point}) = 0$, $d_T(\text{line}) = 1$.

Consider a set A. Split it into subsets A_i, diam $A_i < \varepsilon$. Let

$$m(\varepsilon, p) = \inf_{\{A_i\}} \sum_i (\operatorname{diam} A_i)^p,$$

$$d_M = \sup_p \left\{ p \mid \sup_{\varepsilon > 0} m(\varepsilon, p) > 0 \right\}.$$

Note that if $d_M > d_T$ A is a fractal.

Let $N(\varepsilon)$ be the number of non-empty cubes with diam $= \varepsilon$. Then, capacity is given by

$$D_0 = \lim_{\varepsilon \to 0} \frac{\ln N(\varepsilon)}{\ln\left(\frac{1}{\varepsilon}\right)}.$$

5. 2. Fractal dimension estimation

1. $\{x_1, ..., x_N\} \to \{y_1, ..., y_M\}$, $y_i = \left[x_i, x_{i+\tau}, ..., x_{i+\tau \cdot (m-1)}\right]$. x_i — scalars, y_i — vectors, $y_i^{(k)}$ — k-th value of y_i
2. Normalization:

$$\tilde{y}_i^{(k)} = \frac{y_i^{(k)} - \min_j y_j^{(k)}}{\max_j y_j^{(k)} - \min_j y_j^{(k)}},$$

which results in all coordinates lying within unit hypercube $[0, 1]^m$.

3. Choose a sequence of box sizes

$$\varepsilon_l = \varepsilon_{\max} \cdot q^l,\ l = 0, 1, ..., L,$$

where $q \in (0, 1)$, L is such that $\varepsilon_{\min} \ll 1$ and $N(\varepsilon_{\min}) \gg 1$.

4. Calculating $N(\varepsilon)$. For each box size ε_l the entire unit cube is partitioned into non-overlapping hypercubes with side length ε_l giving $K = \lceil \frac{1}{\varepsilon_l} \rceil$ boxes along

each dimension. For each point y_i the indices of the box containing it are computed:

$$\text{Index}_k = \lfloor \frac{y_{i,k}}{\varepsilon_l} \rfloor,\ k = 1, ..., m.$$

Unique sets of indices are markes as occupied boxes, $N(\varepsilon)$ is the number of such boxes containing at least one point of hte attractor.

Plotting $\ln N(\varepsilon)$ against $\ln \frac{1}{\varepsilon}$ we get a line: $\ln N(\varepsilon) = \alpha + D_0 \ln\left(\frac{1}{\varepsilon}\right)$.

5. 3. Correlation dimension

$$D_2 = \lim_{r \to 0} \frac{\ln C(r)}{\ln r},$$

where $C(r)$ is correlation integral.

Consider a set fo points in m-dimensional phase space $\{y_i\}_{i=1}^{M}$, then:

$$C(r) = \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} \theta\left(r - \|y_i - y_j\|\right),$$

where $\theta(x)$ is a Heaviside function. Generally,

$$C(r) = \int \mu(B(x, r)) d\mu(x)$$

where $B(x, r)$ is ball of radius r with center at x and μ is a metric function.

1. Reconstruction $x_i \to y_i$.
2. Define a grid for r (usually as geometric progression).
3. $d_{ij} = \|y_i - y_j\|$
4. $C(r) = \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} \theta\left(r - d_{ij}\right)$.
5. $C(r) \propto r^{D_2} \Rightarrow \ln C(r) = \alpha + D_2 \cdot \ln r$ (use only part of data that creates the line).

$$
\begin{aligned}
H_q &= \frac{1}{1-q}\log\left(\sum_i p_i^q\right), \\
H_q(\varepsilon) &= \alpha + D_\varepsilon \log\frac{1}{\varepsilon}, \\
D_q &= \lim_{q\to\infty}\frac{H_q(\varepsilon)}{\log\frac{1}{\varepsilon}}.
\end{aligned}
$$

5. 4. Lyapunov dimension

$$
D_L = k + \frac{\log(\lambda_1\lambda_2 \ldots \lambda_k)}{\log(\lambda_{k+1})},
$$

where k is the largest integer such that $\lambda_1, \ldots, \lambda_k \geq 1$.

5. 5. Restoring the equation of a dynamical system

Consider a system

$$
\begin{cases}
\dot{x} = \sigma(y-x) \\
\dot{y} = x(\rho - z) - y \\
\dot{z} = xy - \beta z
\end{cases}
$$

where $x = x(t),\ y = y(t),\ z = z(t)$.

1. Take a matrix

$$
\Theta = \begin{pmatrix}
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
1 & x & y & z & x^2 & y^2 & xy & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots &
\end{pmatrix}
$$

2. Take Ξ

$$
\Xi = \begin{pmatrix}
\vdots & \vdots & \vdots \\
\xi_1 & \xi_2 & \xi_3 \\
\vdots & \vdots & \vdots
\end{pmatrix}
$$

such that $[\dot{x}, \dot{y}, \dot{z}] = \Theta \times \Xi$ is equivalent to the initial system, or

$$\hat{y} = X \cdot \Theta, \quad \|\hat{y} - y\| \to \min_{\Theta}.$$

$$\|\dot{Y} - \hat{\dot{Y}}\| + \alpha \cdot \|\Xi\|_1 \to \min$$

where $\alpha \cdot \|\Xi\|_1$ is l_1 regularization term (generally with a sizable α value).

Take an autoencoder, where Ψ is the encoder, Φ is the decoder, x is the input, x' is the output and z is the latent space value.

$$\dot{z} = \Theta \times \Xi = \begin{pmatrix} \vdots & \vdots & \vdots & & \\ 1 & z & z^2 & z^3 & \ldots \\ \vdots & \vdots & \vdots & & \end{pmatrix}.$$

where z is a latent variable / space / idk. Then, the loss would be:

$$\mathcal{L} = \alpha_1 \ \|x' - x\| + \alpha_2 \ \|\hat{\dot{z}} - \dot{z}\| + \alpha_3 \|\Xi\|_1 \to \min_{\Xi, \Psi, \Phi}.$$

6. Calculating fractal dimension of an attractor from time series data

7. Hurst exponent and how to calculate it

Def. 1 The Hurst exponent is a quantitative measure of the persistance (long-term memory) of a time series.

It can be interpeted accodin to the following rule:

- $H > 0.5$ is characteristic for series with a persistent trend;
- $H = 0.5$ is characteristic for random series (i.e. those that lack persistent memory);
- $H < 0.5$ is characteristic for a persistent anti-trend (trend tends to reverse).

It can be calculated using multiple differenct algorithms, for example R / S algorithm.

7. 1. R / S algorithm

Consider a time series $\{X_i\}_{i=1}^{N}$.

1. Reconstruct the series into a set of embeddings of length m. Denote the size of embedding set itself as n.
2. For each embedding calculate the mean and standard deviation:

$$X_k = \frac{1}{m}\sum_{i=1}^{m} X_{(k-1)m+i},$$

$$S_k = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(X_{(k-1)m+i} - X_k\right)^2}.$$

3. Compute the normalized time series (i.e. cumulative sum of deviations from the mean):

$$Y_{k,i} = \sum_{j=1}^{i}\left(X_{(k-1)m+j} - X_k\right), i = 1, ..., m.$$

4. Compute the range for each embedding:

$$R_k = \max_{1\le i\le m} Y_{k,i} - \min_{1\le i\le m} Y_{k,i}$$

5. Normalize the ranges:

$$(R/S)_k = \frac{R_k}{S_k}, S_k \neq 0.$$

6. Average the ranges over all embeddings to get R/S value for the selected m:

$$(R/S)^m = \frac{1}{n}\sum_{i=1}^{n}(R/S)_i.$$

7. Repeat the previous steps for various values of m, typically $10 \leq m \leq \frac{N}{2}$ with logarithmic step.
8. Fit a linear regression on $(R/S)^m$ for various m:

$$\log{(R/S)^m} = a + H \cdot \log m + \varepsilon,$$

where H is Hurst exponent.

8. Why we can forecast times series? Taken's theorem with application for time series forecasting

8. 1. Smooth manifolds and smooth maps

Let $\mathbb{R}^k$ be a k-dimensional euclidean space (i.e. linear space with scalar product defined), then $x \in \mathbb{R}^k,\ x = (x_1, ..., x_k)$.

Def. 1 Let $U \subset \mathbb{R}^k,\ V \subset \mathbb{R}^l$ be two open sets. A mapping $f : U \to V$ is called *smooth* if all partial derivatives $\frac{\partial^n f}{\partial x_{i_1} ... \partial x_{i_k}}$ exist and are continuous.

Def. 2 A map $f : X \to Y$ is a *homemorphism* if:

1. $f(X) = Y$ is a bijection;
2. f and f^{-1} are continuous.

Def. 3 A map $f : X \to Y$ is a *diffeomorphism* if:

1. f and f^{-1} are smooth;
2. f is a homeomorphism.

If $f : X \to Y$ and $g : Y \to Z$ are smooth, then $g \circ f : X \to Z$ is smooth as well.

Def. 4 A set $M \subset \mathbb{R}^k$ is a *smooth manifold* of dimension m if $\forall x \in M$ exists a neighbourhood $W \cap M \neq \emptyset$ which has a diffomorphic map to an open set $U \subset \mathbb{R}^m$.

Note. Any diffeomorphism $g : U \to W \cap M$ is called a parametrisation of $W \cap M$. The inverse map $g^{-1} : W \cap M \to U$ is called a coordinate system on $W \cap M$.

8. 2. Mathematical foundations for time series analysis

Let $\varphi^t(x)$ be a dynamical system, P its phase space, τ the time step between two consequtive observations and a scalar function $h : P \to \mathbb{R}$ the observation function of states of the dynamical system.

Denote states of the dynamical system $\varphi^t(x)$ as $\vec{x}(t_i), \vec{x}(t_{i+1}), ...$ and time series values of the observation function as $y_i = h(\vec{x}(t_i))$. Then:

$$y_i = h(\vec{x}(t_i)) = h(\varphi^{t_i}(x_0))$$

For the sake of simplicity denote $x(t_i) = x_i$. Given time step τ, state transitions for a dynamical system could be represented in the following way:

$$x_{i+1} = \varphi^{\tau}(x_i), x_{i+2} = \varphi^{2\tau}(x_i), \dots$$

Then, a system of equations can be constructed as follows:

$$\begin{cases} y_i = h(x_i) = \Phi_0(x_i), \\ y_{i+2} = h(\varphi^{\tau}(x_i)) = \Phi_1(x_i), \\ \dots \\ y_{i+m-1} = \Phi_{m-1}(x_i). \end{cases}$$

This system describes how z-vectors are constructed. Next, for $x_i \in M^d \subset P$, define $\Lambda : M^d \to \mathbb{R}^m$ where

$$\Lambda(x_i) = \left(h(x_i), h(\varphi^{\tau}(x_i)), \dots, h\left(\varphi^{(m-1)\tau}(x_i)\right)\right) = \left(y_i, \dots, y_{i+m-1}\right).$$

There are several conditions placed upon Λ:

1. Λ should be a bijection;
2. Λ should be Lipshitz continuous.

Def. 5 A mapping $f : X \to Y$ is *Lipschitz continuous* if there exists such $L \geq 0$ that for all $x_i, x_j \in X$

$$\rho_X\left(f(x_i), f\left(x_j\right)\right) \leq L \cdot \rho_Y\left(x_i, x_j\right).$$

Def. 6 A manifold is *compact* if every open over of it has a finite subcover: if every collection C of open subsets of X such that

$$X = \cup_{S \in C} S,$$

there is finite subcollection $F \subseteq C$ such that

$$X = \cup_{S \in F} S.$$

Functionally it is a generalization of the notion of closed sets.

Theorem 1 (Taken's delay embedding theorem) Let $M \in \mathbb{R}^k$ be a compact smooth manifold, let τ be the lag between obsetvation, and let $\varphi : M \to M$ be a diffeomorphism. Given an observation function $h : M \to \mathbb{R}$ that produces scalar time series data one can assert that for a generic h, the map

$$\Lambda(x_i) = \left(x_i, h(\varphi(x_i)), ..., h\left(\varphi^{(m-1)\tau}(x_i)\right)\right)$$

is an embedding (a smooth bijection) for $m > 2k$.

Corollary 1 The reconstructed observation space contains all topological invariants and dynamical features of the original attractor including periodic orbits, Lyapunov exponents and entropy.

Let S be the image space of an embedding Λ. Then, the following dynamic systems could be defined:

$$\begin{aligned} x_i &= \Lambda^{-1}(z_i), x_{i+1} = \varphi^{\tau}(x_i), \\ z_{i+1} &= \Lambda(x_{i+1}) = \Lambda(\varphi^{\tau}(x_i)) = \Lambda\left(\varphi^{\tau}\left(\Lambda^{-1}(z_i)\right)\right) = \Psi(z_i), z_i \in S. \end{aligned}$$

Here, $z_i = \left(y_i, ..., y_{i+m-1}\right)$ and the pair of systems can be denoted as $\varphi : M \to M$ and $\Psi : S \to S$.

$\Psi : S \to S$ can be used to predict future values of the time series. Given $z_{i+1} = \Psi(z_i)$,

$$z_{i+1,m} = y_{i+m+1-1} = F(z_i) = F\left(\left(y_i, ..., y_{i+m-1}\right)\right).$$

Choosing parameters:

- m is the smallest embedding size that produces FNN to NN smaller than a preset value (e.g. $< 1\%$);
- τ should be eather the first zero fo ACF or the first minimum of mutual information.

9. Modern neural network methods for forecasting: NHITS, TimesNet, PatchedTST