Time Series II — Nonlinear Analysis

Prof. Ned Wingreen

MOL 410/510

Example — model four in Kaplan and Glass

Imagine our time series of data is generated by the logistic map:

$$x_{t+1} = \mu x_t(1 - x_t).$$

With $\mu = 4$, this map generates chaos. The system is indeed deterministic, but

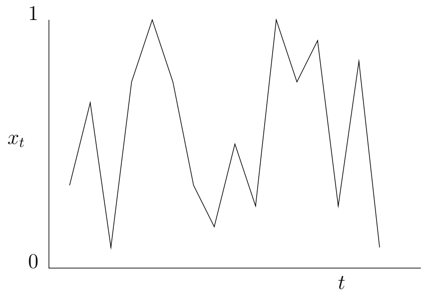

Figure 1: Logistic map

it looks "noisy". The autocorrelation function is the same as for measurement

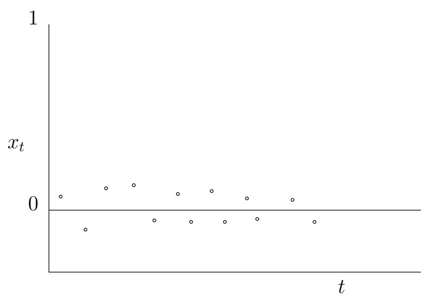

Figure 2: Autocorrelation function

noise ("model one" from the last lecture). Recall that $R(k)$ captures linear correlations, but there is no guarantee for nonlinear systems. This suggests two questions:

- How can we determine if the behavior of x_t is deterministic?

- And if it is, can we reconstruct the dynamics?

Idea: if the dynamics is deterministic, x_{t+1} should depend on previous x_t, x_{t-1}, etc.

Example — logistic map

Plot $V_{t+1} = x_{t+1} - \langle x \rangle$ vs. $V_t = x_t - \langle x \rangle$ as a scatter plot: the "Return plot"

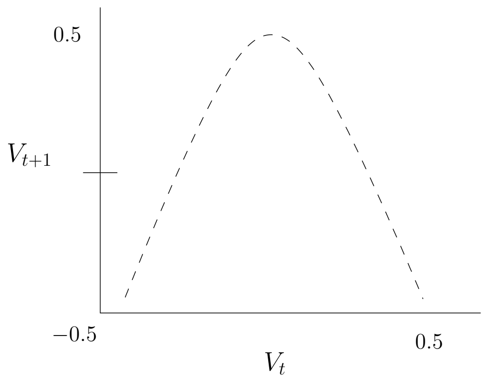

Figure 3: Return plot

shows that x_t is deterministic, and x_{t+1} depends only on previous x_t.

What if the system is deterministic but more complex than 1d FDE?

Example — two coupled ODEs

$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$

If we know $x(t)$ and $\dot{x}(t)$ (or two other pieces of information), then the dynamics are defined. So we can define an alternative phase plane using x and $\dot{x}$ without ever measuring y. Furthermore,

$$\dot{x} \approx \frac{x(t+h) - x(t)}{h}.$$

We can just use $x(t+h)$ and $x(t)$ to define a phase plane. Take, for example, the Van der Pol system. As figures 6.18 and 6.19 in Kaplan and Glass show, the approximate and exact phase plots are not identical, but they are very similar. Equations of this form are clearly deterministic and we have (approximately) reconstructed the dynamics.

What if $d > 2$?

Embedding a time series

Generalize the approach used to reconstruct a 2d phase portrait to higher dimensions by using additional time-lagged measurements:

$$\vec{x}_t = (x_t, x_{t-h}, x_{t-2h}, \ldots, x_{t-(p-1)h}),$$

where

$$p = \text{ embedding dimension}$$
$$h = \text{ embedding lag}$$

In practice, we can keeping adding dimensions until we uncover the dynamics, or we get tired.

Example — Lorenz equation

$$\frac{dx}{dt} = 10(y - x)$$
$$\frac{dy}{dt} = 28x - y - xz$$
$$\frac{dz}{dt} = 28xy - \frac{8z}{3}$$

There are 3 coupled equations, so we need at most $p = 3$. In practice, we might only have $x(t)$. The we would embed with $p = 2, p = 3$, etc. Figures 6.21 and 6.22 in Kaplan and Glass illustrate this technique.

How do we know what h value to use? If h is too small, then $x_t \approx x_{t-h} \approx x_{t-2h}$. If h is too big, then x_t, x_{t-h}, x_{t-2h} are uncorrelated. Best bet is to use an h that reflects the time scale of the dynamics, e.g. approximate periodicity.

Aside — Measuring dimension

Take, for example, the Lorenz equation. The attractor has dimension $D \approx 2.06$. How to measure d from reconstructed or real dynamics?

We use the "Box-counting" dimension:

1. Cover all points with boxes of integer dimension p and edge length ϵ_0, and count the number of boxes needed: $N(\epsilon_0)$.

2. Repeat step 1 using boxes of edge length $\epsilon_1 = \epsilon_0/2, \epsilon_2 = \epsilon_1/2$, etc.

3. The dimension D is given by

$$\lim_{\epsilon \to 0} N(\epsilon) = c/\epsilon^D,$$

so that

$$D \approx \frac{\log[N(\epsilon_{i+1})/N(\epsilon_i)]}{\log[\epsilon_i/\epsilon_{i+1}]}.$$

Usually we find that, over a range of ϵ, D is approximately a constant, this in practice is the "dimension".

Detecting determinism in high dimensions

We use the fact that $x(t)$ should be determined by previous values of x, that is, future values can be predicted from past values.

We construct an algorithm to predict values of the time series based on the past time series.

To predict x_{t+1} from $x_1, x_2, \ldots, x_t$:

1. Embed the time series $\vec{x}_t = (x_t, x_{t-h}, \ldots, x_{t-(p-1)h})$.
2. Find the point $\vec{x}_a$ closest to $\vec{x}_t$. $(a < t)$
3. Prediction: $x_{t+1} \approx x_{a+1}$.

Note: we didn't need to build a model for the data to make a prediction, we just used the existing datat directly: this is a "data-implicit model". Variants:

2′. Find k points $\vec{x}_{a_1}, \vec{x}_{a_2}, \ldots, \vec{x}_{a_k}$ closest to $\vec{x}_t$.

3′. Prediction: $x_{t+1} \approx \frac{1}{k}\sum_{i=1}^{k} x_{a_i+1}$, the average of k closest point predictions.

Questions — What are the advantages of embedding and averaging?

p is required to be big enough to capture underlying dynamics, e.g. for Lorenz equation setting $p = 1$ (i.e., just one previous x) is not enough since $d = 2.06$.

Overall, this is a useful approach for complex systems – the use of data is unbiased.

Applications

- Using iteration to predict future beyond x_{t+1} (e.g. the ice ages example in the text)
- Strategies: can the data be used to detect determinism? Can the data predict itself?

Given $x_1, \ldots, x_n$, make predictions for some subset of x_{t+1}'s: $P_{t_1+1}, P_{t_2+1}, \ldots, P_{t_m+1}$. To see if a system is deterministic, compare predictions to actual x_{t+1} values:

$$\text{Prediction error } E = \frac{1}{M}\sum_{i=1}^{M}(P_{t_i+1} - x_{t_i+1})^2.$$

Compare — if we used the sample mean as our prediction for every x_{t+1}:

$$\frac{1}{M}\sum_{i=1}^{M}(\text{mean } - x_{t_i+1})^2 \approx \sigma^2.$$

So, the quality of our prediction depends on how E compares to σ^2.

$$\frac{E}{\sigma^2} = \text{relative prediction error: } \begin{cases} \geq 1 & \text{random} \\ 0 & \text{perfect determinism} \\ 0 < \frac{E}{\sigma^2} < 1 & \text{some determinism} \end{cases}$$