University of Colorado
Department of Computer Science

Chaotic Dynamics – CSCI 4446/5446
Spring 2019

Problem Set 8

Issued:	5 March 2019
Due:	12 March 2019

Graduate Students: project paragraphs are due in class on 14 March. I will send around a doodle poll that day as well for the one-on-one project meetings (which are scheduled for the week before spring break).

Reading: *Please see the "Assigned reading for PS8-10" handout on the course webpage.*

NOTE! you should make sure that you get access to TISEAN SOON so you have time to ask for help if you run into any snags. See problem 4 below.

Online assignment: Tuesday: units 8.1-8.3 videos. Thursday: units 8.4-8.5 videos. Friday: quizzes 8.1-8.5.

Bibliography:

- H. Abarbanel, *Analysis of Observed Chaotic Data*, Springer, 1995. Another good book on time-series analysis, from embedding onwards. This is on library reserve.
- E. Bradley and H. Kantz, "Nonlinear Time-Series Analysis Revisited," *Chaos* **25**:097610 (2015). `DOI: 10.1063/1.4917289`.
- A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical Review A* **33**:1134-1140 (1986). The original paper on the use of average mutual information in estimating τ.
- J. Garland and E. Bradley, "Prediction in Projection," *Chaos* **25**:123108 (2015) `DOI: 10.1063/1.4936242`.
- M. B. Kennel *et al.*, "Determining minimum embedding dimension using a geometrical construction," *Physical Review A*, **45**:3403-3411 (1992). The original paper on the use of average mutual information in estimating m. (A synopsis of this algorithm appears on page 17 of my notes on nonlinear time-series analysis, which you can find on the course webpage. If you want the full paper, it's in the "Coping..." collection.)
- N. Packard *et al.*, "Geometry from a Time Series," *Physical Review Letters* **45**:712 (1980). The first of the two original papers on embedding; also in the "Coping..." collection.
- T. Sauer *et al.*, "Embedology," *Journal of Statistical Physics*, **65**:579-616 (1991). The definitive paper about the broader field of embedding, but long & highly technical. Proves $m < 2d_{cap}$.

- T. Sauer, "Interspike interval embedding of chaotic signals," *Chaos*, **5**:127 (1995). An embedding method that uses the intervals between discrete events, rather than evenly-sampled measurements of the whole time line.

- F. Takens, "Detecting strange attractors in fluid turbulence," in *Dynamical Systems and Turbulence* (D. Rand and L.-S. Young, eds.), Springer, Berlin, 1981. The other original paper on delay-coordinate reconstruction.

Problems:

In this problem set, you will explore some simple embedding algorithms, using position-versus-time data gathered from the driven pendulum that I've showed you in videos. I have posted three data sets on the class webpage; see the PS8 entry on that page for directions (and a clickable link) to these data. In all three runs, the angle was measured every Δt seconds using an optical encoder with a resolution of 0.4 degree. The drive amplitude was fixed; the drive frequency (the bifurcation parameter) was different for each data set:

- in `data1`, the drive was turned off
- in `data2`, the drive is on, with a medium frequency
- in `data3`, the drive is on, with the same amplitude but a higher frequency

> *Size Issues:* These data files contain up to 6MB of information. It would make sense to debug your code on test files that consist of *small chunks* of these files.

Each file captures a single trajectory of the driven pendulum (except that data2 was so big that I broke it down into four pieces: `data2.first250sec`, `data2.second250sec`, and so on). Each line of each file represents a single time-sample of the pendulum's angular position. Each of these data points looks like this:

θ `time`

...where time is in seconds and θ is mod 2π. Depending on when I hit the reset button on the data-acquisition system, θ may contain an offset, so "θ=0" may not be "vertical." Also, note that the sampling *rate* was different; `data1` and `data3` were sampled at $\Delta t = 0.001$ seconds and `data2` at $\Delta t = 0.002$ seconds.

The time base and thus the sampling interval in the data acquisition channel were not quite uniform. Together with the finite precision of the angle sensor, this standard problem with computer-mediated data acquisition has two important implications:

- Any ωs that you reconstruct using divided differences from the θ and time data may be inaccurate. You'll explore this in problem 1.

- Nonuniform sampling violates the conditions of the Takens theorem, so any attractors constructed via embeddings of these data are *not* true diffeomorphic copies of any attractor that may exist in the system...but they're pretty close. If we wanted, we could mitigate the effects of that by using embedding intervals that are much larger than the experimental sampling interval, or by interpolation, if we knew exactly how far off our samping interval was; see the optional reading listed above for details. (There is no need for you to do that in this assignment.)

1. Write a program that steps through a data file, constructs values for ω using divided differences—first-order forward is good enough, but you may use something smarter if you want—and plots the results in state-space form, with θ mod 2π.

Note that if data are *oversampled*—that is, if the sampling rate is much faster than the device's dynamics, as is the case in the data that you're working with here—you have to be a little careful about the divided difference formulae. In particular, you're going to need to *downsample* the data in order to get a sensible plot. The choice of downsampling rate is part of the thinking part of this problem.

If the sampling rate is *slower* than the dynamics, on the other hand, you've missed some of the behavior. That's a different and nastier problem that is not at issue here.

Apply this program to `data1` and turn in a plot. Since the drive is off, this plot *should* be a clean spiral (why?). Please comment on what it really looks like, as well as on possible causes for this.

2. Write a program that steps through a data file and *embeds* the θ data, producing the corresponding trajectory in *reconstruction space*. This program should take a time interval τ, a dimension m, and indices j, k of a pair of axes on which to plot the results. It should produce a list of m-vectors (points in reconstruction space) each of whose i^{th} element is $\theta(t + i\tau)$ for $i = 0..m-1$. Finally, for each m-vector, it should plot the j^{th} element against the k^{th} element, both mod 2π.

> *Aside:* τ is usually an integer multiple of the sampling interval Δt in the data set; if it isn't, interpolation may be called for. (This is not an issue in this problem set.) In cases like this, people often sidestep the issue and just use the data point that is closest to the sampling interval. Sometimes, they do a linear interpolation between the points on opposite sides of the interval boundary. Note that $\theta(t)$—the first coordinate of the reconstruction-space point—should *always* be a real data point. Again, see the optional reading listed above for details.

(a) Run your embedding program on the `data2` set with $\tau = 0.15sec$ and $m = 7$. Plot the zeroth element of the reconstructed state vector—$\theta(t)$—on the vertical axis and the second ($\theta(t + 0.3)$, here) on the horizontal axis (i.e., $j = 0$ and $k = 2$). What kind of attractor is this? Turn in a copy of the plot.

(b) Now play with the `data3` set: use $m = 7$ and start with $\tau = 0.01sec$, then raise τ to 1.5, checking at least three intermediate points along the way—e.g., $\tau = 0.01,\ 0.15,\ 0.5,\ 1,\ 1.5sec$. What kind of attractor is this? Describe and explain the effects of the different τs and turn in one or two interesting plots—of $\theta(t)$ against $\theta(t+5\tau)$ this time—that back up your explanations.

3. *[Thought experiment]* (a) In all of the problems above, we used $m = 7$ whether or not the drive was on. What requirements does the Takens theorem place on m for a successful embedding of the *driven* pendulum? What about the *undriven* pendulum?

(b) What do you think would happen to the reconstructed trajectory—not just your picture, but the full trajectory—in part (b) of problem 2 if you had used $m = 2$ or $m = 25$? (one or two sentences only, please).

(c) What do you think would happen to the reconstructed trajectory in part (a) of problem 2 if you had used $\tau = 10^{-16}$—which would require much more frequent sampling, obviously; for the purposes of this answer, assume that that was the case—or $\tau = 10^6$? (one or two sentences only, please). What would your pictures look like?

> You'll need access to the TISEAN package to do the next two problems. It's easiest to just use it on the CSEL "virtual machine," but you may want to download and install it separately on your own machine using the links on the course webpage. See the PS8 entry on the course homepage for more information.

4. The τ parameter is critical to a successful embedding. The standard first step in the analysis of a scalar time series from a nonlinear dynamical system is to choose a good value for that parameter.

Use TISEAN's `mutual` tool to construct a plot of mutual information versus τ for `data2.first250sec`. Note! In `mutual`'s output, τ is actually reported in *units of the sample interval* (Δt), not *seconds*. Start with the default values of the algorithm parameters—that is, just run:

```
mutual data2.first250sec -o outputfile
```

What you're looking for is the *first minimum* of this curve. Do you see one? (Hopefully not; I didn't when I ran that experiment.) That means that the default value that TISEAN uses for the "max time delay" parameter of the `mutual` algorithm is not high enough for this data set. Increase it until you see the first minimum. Mark that value on the plot and convert that value into seconds. To do this calculation, you'll need to look at the data file to determine the sample time Δt. Also write down the `mutual` call that you used to produce the data in the plot (`mutual -o blah...`)

5. The standard second step in nonlinear time-series analysis is to choose a good value for the embedding dimension m. To do this, use TISEAN's `false_nearest` tool to construct a plot of the percent of false-near neighbors versus m for `data2.first250sec`. Use an m range of [1,10], plug in your value for τ from problem 4 as the delay parameter, and leave the rest of the parameters at their default values. Make a plot of the results. The first column in the file that `false_nearest` produces is the m and the second is the *ratio* of false neighbors that it found at that m. The standard rule of thumb is to choose the m value where that ratio first gets below 0.1 (i.e., 10% false neighbors). Mark that m value on the plot, write down the value, and turn in a copy. Also write down the `false_nearest` call that you used to produce the data in the plot.