University of Colorado
Department of Computer Science

Chaotic Dynamics – CSCI 4446/5446
Spring 2012

Problem Set 9

Issued: 13 March 2012

Due: 20 March 2012

Reading: *section 3.2 of Liz's time-series analysis notes; chapter 5 and appendix A of Kantz & Schreiber; section 10.5 of Strogatz; Liz's notes on Wolf's algorithm*; sections 5.3 and 5.5 of Abarbanel; "Coping...", chapters 4 (an overview) and 8 (reprints of journal papers).

NOTE! you should work on getting the TISEAN code downloaded and running (*viz.*, problem 1) SOON so you have time to ask for help if you run into any snags.

Bibliography:

- H. Abarbanel *et al.*, "Lyapunov exponents in chaotic systems: their importance and their evaluation using observed data," *International Journal of Modern Physics B*, **5**:1347-75 (1991). There's a related paper by Abarbanel on page 229 of the Casdagli reprint collection, which is on library reserve.
- H. Abarbanel *et al.*, "Variation of Lyapunov Exponents on a Strange Attractor," *J. Nonl. Sci.* **1**:175 (1991). [*local* λs]
- P. Bryant *et al.*, "Lyapunov exponents from observed time series," *Physical Review Letters*, **65**:1523-6 (1990).
- P. Cvitanovic, `http://www.nbi.dk/~predrag/papers/preprPOT.html`. Lots of good papers on UPOS. Also see Gunaratne and So papers listed below.
- J.-P. Eckmann and D. Ruelle, "Ergodic theory of chaos and strange attractors," *Rev. Mod. Phys* **57**:651 (1985).
- J.-P. Eckmann *et al.*, "Liapunov exponents from time series," *Phys. Rev. A* **34**:4971 (1986).
- G. Gunaratne *et al.*, "Chaos beyond Onset: A Comparison of Theory and Experiment," *Phys. Rev. Lett.* **63**:1-4 (1989). (Gives a good algorithm for finding unstable periodic orbits.)
- P. So *et al.*, "Detecting unstable periodic orbits in chaotic experimental data," *Physical Review Letters* **76**:4705-4708 (1996).

- M. Sano, "Measurement of the Lyapunov Exponents from a Chaotic Series," *Physical Review Letters* **55**:1082-1085 (1983).
- A. Wolf, "Quantifying chaos with Lyapunov exponents," in *Chaos*, Princeton University Press, 1986. (A reasonable review article.)

Problems:

1. Download and install TISEAN (links on the course webpage). If this turns out to be a challenge, you may access this toolset in the CSEL, so don't spend a lot of time fighting with the makefiles! If you can't get it installed *and* you're not a CS major, please email me and I'll arrange for access to the CSEL for the rest of the semester.

2. The standard first step in the analysis of a scalar time series from a nonlinear dynamical system is to choose the delay τ. Use TISEAN's `mutual` tool to construct a plot of mutual information versus τ for `data2.first250sec`. 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 τ value on the plot, write down the value, and turn in a copy. Also write down the `mutual` call that you used to produce the data in the plot (`mutual -o blah...`).

3. The standard second step in nonlinear time-series analysis is to choose 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 2 as the delay parameter (`-d`), 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 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.

4. Use TISEAN's `delay` tool to embed `data2.first250sec` using the τ and m you found in problems 2 & 3. Turn in a plot of a 2D projection (any one you want) of the data and write down the `delay` call that you used to produce the data in the plot. Compare this plot to the one that you constructed in PS8; discuss any similarities/differences.

5. (a) Use TISEAN's `lyap_r` tool to estimate the largest Lyapunov exponent of the `data2.first250sec` data set. You'll need to supply this algorithm with values for the embedding parameters. For this part of this problem, use the τ and m you found in problems 2 & 3. Turn in the plot of the results from which you calculated λ and explain how you obtained your answer—i.e., the value of λ—from that plot. Does that

answer make sense, given that this is a trajectory from a device that is exhibiting chaotic behavior?

(b) Now try varying m. If your m value from problem 3 was correct, then increasing it above that value should not change the `lyap_r` results (much). If it was *not* correct—too small, for instance—then the `lyap_r` results will keep changing as you raise m beyond that value...either until m gets big enough to allow the embedding process to unfold the attractor—at which point the results will settle out—or until your algorithm gets tripped up by noise or data effects, in which case they will never settle. What do you observe in your results? What does that suggest about your m value and/or the data?

(c) *[Optional for those registered for CSCI 4446]* Play with at least one of the other parameters of the `lyap_r` algorithm: the Theiler window `-t`, the epsilon size `-r` (and, if you wish, the number of iterations `-s`). Describe **and explain** their effects upon the results. These explanations are going to require some digging around on the TISEAN website and corroboration against the class readings and your lecture notes. Please comment on of the results you trust more, and why. What were your "lessons learned" from your experience with this part of this problem?

6. (a) A good way to test out any data-analysis tool is to feed it a "synthetic" data set—a cooked-up one where you know the right answer. Use RK4 to generate a 30,000 point trajectory starting from some point near the $a = 16, r = 45, b = 4$ Lorenz attractor using a step size of 0.001. Throw out the y and z coordinates and embed the x coordinate. Use `mutual` to estimate τ but don't repeat the `false_nearest` analysis to choose m; rather, just use $m = 7$ (which is the Takens requirement for a three-dimensional system like Lorenz).

What does `lyap_r` think the λ is for the reconstructed attractor? Again, turn in the plot that you used to produce that answer. Does that value make sense, given what you know about this system?

(b) Run your PS7 code from the same initial condition as in part (a) of this problem, with the same solver and system parameters, then compute the eigenvalues of the resulting variational matrix and use them to calculate the Lyapunov exponents:

$$\lambda_i := lim_{t \to \infty} \frac{1}{t} \ln |m_i(t)|$$

where $m_i(t)$ are the eigenvalues of the variational matrix $\{\delta_{ij}\}$. Compare these results to your answers from part (a) of this problem.