University of Colorado
Department of Computer Science

Chaotic Dynamics – CSCI 4446/5446
Spring 2012

Problem Set 10

Issued:	20 March 2012
Due:	3 April 2012

Graduate Students: project proposal pages are due in class on 5 April!

Reading: *Strogatz, sections 11.4-5, page 447, and plate 3; Kantz&Schreiber, chapter 6.*

Bibliography:

- J.-P. Eckmann and D. Ruelle, "Fundamental limitations for estimating dimensions and Lyapunov exponents in dynamical systems," *Physica D* **56** (1992). In "Coping..."
- P. Grassberger, "Estimating the fractal dimensions and entropies of strange attractors," in *Chaos* (Holden, ed), Princeton University Press.
- F. Hunt and F. Sullivan, "Efficient algorithms for computing fractal dimensions," in *Dimensions and Entropies in Chaotic Systems*, Springer-Verlag, Berlin, 1985, pp 74–81. A synopsis of this paper appears on pp 180–181 of Parker & Chua.
- F. Pineda and J. Sommerer, "Estimating generalized dimensions and choosing time delays: A fast algorithm," in *Time Series Prediction: Forecasting the Future and Understanding the Past*, which is on library reserve (pp 367-386). A synopsis of this algorithm appears on pp 16-17 of Liz's TSA Notes.
- D. Russell, J. Hanson, and E. Ott, "Dimension of strange attractors," *Physical Review Letters* **45**:431-434 (1980).
- R. Smith, "Optimal estimation of fractal dimension," pp115-137 of the *Nonlinear Modeling and Forecasting* reprint collection, which is on library reserve. Heavy going, but very thorough.
- Froehling *et al*, "On determining the dimension of chaotic flows," *Physica D* **3**:605-617 (1981).

Problems:

1. Implement the box-counting algorithm for computing capacity dimension from an experimental data set. You are welcome to use any of the several versions that appear in

the assigned reading—and to use any data structure and algorithm that you wish—but please give an english description of what you're doing.

2. (a) Test your algorithm on the embedded Lorenz attractor that you created for problem 6(a) on PS9. Play with the box size: start large and work your way down until either the dimension stabilizes or you run out of memory. Explain your results and turn in a plot of $\log N(\epsilon)$ versus $\log(1/\epsilon)$.

(b) Now try your box-counting algorithm on the original trajectory whose x values you used to create that embedding. Compare/contrast your results—both the values that your algorithm produces and the time/memory demands involved.

(c) Repeat part (b) of this problem with a longer version of the same trajectory (i.e., same time step, but more steps). Do your results change? If so, which do you trust most? Why?

3. *[thought experiments]* (a) Suggest one or two ways to get around the box size/memory limitation you encountered in problem 2 (two-three sentences, total).

4. *[Optional for those registered for CSCI 4446]* Apply TISEAN's correlation dimension calculator (`d2`) to the same Lorenz data. Turn in plots of $C(m, \epsilon)$ versus ϵ and $D(m, \epsilon)$ versus ϵ (see section 6.3 of Kantz&Schreiber) and discuss the results. As with any TISEAN tool, you should explore different values for the various parameters—except the Theiler window, which shouldn't really matter in this algorithm, and so should just be set to zero.

Per our discussion in class, understanding these results is going to involve reading Chapter 6 of Kantz&Schreiber (especially section 6.4), looking carefully at the plot of the `d2` output and interpreting it in light of all of the reading and thinking we've done about these kinds of algorithms in general, and this algorithm in particular. The admonition on the TISEAN webpage echoes this warning:

> If you are looking for a program that reads your signal and issues a number that says "correlation dimension", you got yourself the wrong package. We think you are still better off than getting such a wrong answer. The programs in this section carry out the calculations necessary to detect scaling and self similarity in a fractal attractor. You will have to establish scaling and eventually, in favourable cases, extract the dimension or entropy by careful evaluation of the data produced by these programs.