University of Colorado
Department of Computer Science

Chaotic Dynamics – CSCI 4446/5446
Spring 2016

Problem Set 10

Issued:	17 March 2016
Due:	29 March 2016

Graduate Students: project proposal pages are due in class on 31 March.

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

Bibliography:

- Papers about how much data you need to get nonlinear time-series analysis to work well: J.-P. Eckmann and D. Ruelle *Physica D* **56** (1992), which is in the "Coping..." collection. Also A. Tsonis *et al. J. Atmos. Sci.***50**:2549-2555 (1993) and L. Smith *Phys. Lett. A* **133**:283–288 (1988).
- 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.

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 Lorenz trajectory that you created for problem 1(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, turn in a plot of $\log N(\epsilon)$ versus $\log(1/\epsilon)$, and discuss the shape of that plot. Is it what you expected?

(b) Repeat part (a) 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?

(c) *[Optional for those enrolled in CSCI 4446]* Now try your box-counting algorithm on the embedding of the x coordinate of that attractor that you created for problem 3 of PS9. Compare/contrast the values that your algorithm produces for the true and reconstructed trajectory; comment on similarities and differences.

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

4. *[Optional for those registered for CSCI 4446]* Apply TISEAN's correlation dimension calculator (`d2`) to the reconstructed trajectory that you used in problem 2(c) above. 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.