With J: Fractal Forecasting

Cliff Reiter
Mathematics Department
Lafayette College,
Easton PA 18042, USA
reiterc@lafayette.edu

Modeling techniques forecast future behavior based upon available data. Traditional models use linear or differential equations to create deterministic models that use the available information to estimate parameters in the model and then current information is used as input to the model to forecast subsequent evolution. The idea of fractal or chaotic forecasting is to use historical data that closely matches the pattern of the current situation, and only utilize the most relevant historical data for the construction of the forecast. The technique is fairly simple and can be effective, especially if the phenomenon contains subtly hidden, deterministic, nonlinear behavior. For more discussion of fractal forecasting, see [1,2]; earlier versions of some of these experiments and some variants of the implementation appear in [3].

The Henon Map

We begin by considering the Henon map. This is a nonlinear map that depends upon the previous two data values. In particular, it is defined by the following.

$$h_n = 1 + 0.3h_{n-2} - 1.4h_{n-1}^2$$

We can implement it in J as a function that takes a pair (h_{n-2}, h_{n-1}) and results in the subsequent pair (h_{n-1}, h_n) as follows. Notice that when we iterate a few steps, the first value in a pair matches the last value in the previous pair. Thus, to obtain 1000 steps of the Henon map as `data`, defined below, we need only one column of the result of the iteration.

```
   hen=:{:,1:+0.3&*@{.+_1.4&*@*:@{:

   hen^:(i.6) 0.1 0.5
     0.1      0.5
     0.5     0.68
    0.68  0.50264
 0.50264 0.850294
0.850294 0.138592
0.138592   1.2282

   $data=:{.|:hen^:(3+i.1000) 0 0
1000
```

Figure 1 shows the first 80 values in the data. Notice that the data appears to be fairly random, yet certain patterns of zigzag up-downs seem to roughly reappear. That is the type of subtle, almost repeated, behavior that fractal forecasting exploits.

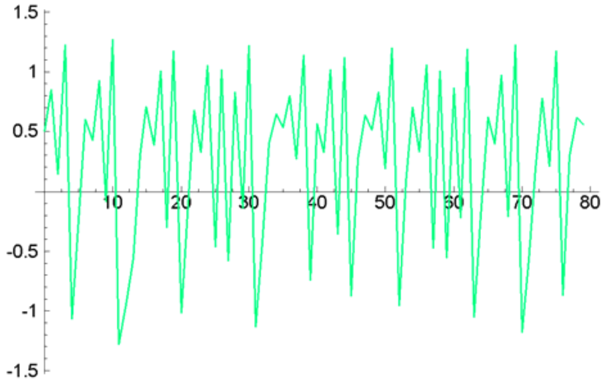

Figure 1. Some Steps of the Henon map.

Fractal Forecasting

We now will transform the first 900 points in our time series into a reference set and try to use it to predict the behavior of some sample data that starts at a latter time. For this time series data we have special knowledge. We know that pairs of points determine the next value. Thus, we will use infix to form moving windows of size 2 into a reference list of pairs of Henon map data. We also select a single sample pair, not in our reference set, from which we will forecast. We define a function `dist` that computes the distance between points. We use it to find the distance between the sample point and the reference data points and then we locate the 5 closest points in the reference list.

```
   dist=: +/&.:*:@:-"1

   4 0 1 dist 2 2 2
3

   w=:2    NB. window size
   $ref =:}: w ]\ 900{.data
898 2

   _3{.ref
_0.353521 0.574497
 0.574497 0.431878
 0.431878 0.911223

   ]sam=:data {~ 950+i.2
1.09422 _0.794183

   5{.sam dist ref
1.74765 0.964141 2.23679 0.306888
2.2355

   5{./:~ sam dist ref
0.00173705 0.00854398 0.0208798
0.0283261 0.031843
```

Notice above that the 5 closest points are a couple of orders of magnitude closer than the first 5 points.

In practice, it is convenient to have the indices where those smallest distances occur. Then we create a least squares linear model using the 5 nearest reference points and their subsequent values. There is nothing special about our choice of using 5 points. We must choose at least one more point than the window width in order to compute the linear model; using too few points risks highly sensitive predictions and using too many points would not take advantage of the pattern that appears in the sample point. The coefficients of the linear model, given in `coef` below, are then used to make a forecast from the sample point and that is compared to the true Henon value below. Notice they do not differ until the third decimal place.

```
   ]j=:5{./: sam dist ref
763 292 497 84 470

   ]coef=:((w+j){data)%. 1,.j{ref
_2.05754 5.19404 4.00548

   coef +/ . * 1,sam
0.444781

   952{data
0.445247
```

Of course, the forecast and the last entry in the sample point can now be used as a new sample point and further forecasts can be made. We automate the process with the function `ffcast`. It is a dyadic function where both arguments are a list of two items. The left argument specifies the number of nearest reference points to use and the number of steps to forecast into the future. The right argument is a boxed list of the reference data as a vector and the sample point from which to forecast. Notice it uses two locally defined functions; one of those functions locates the indices of the best reference points for the given data and the other gets the linear model fitting those points. The last line iterates the process of creating and applying those linear models. We then compare 4 steps of the forecast with 4 steps of real Henon data. Figure 2 compares the prediction with the real data over 20 steps.

```
   ffcast=: 4 : 0
'y sam'=. y.
w=.#sam
'k n'=. x.
ref=.}:w ]\ y
get_j=.k"_ {. [: /: ref"_ dist ]
get_coef=.({&y@:(w&+)%.1:,.{&ref)@:get_j
{:|:(}.,get_coef +/ . *1:,])^:(1+i.n)sam
)

   $pred=:5 4 ffcast (900{.data);sam
4

   $real=: (952+i.4){data
4

   pred,:real
0.444781 0.484614 0.804691 0.238852
0.445247 0.484202 0.805342 0.237254
```

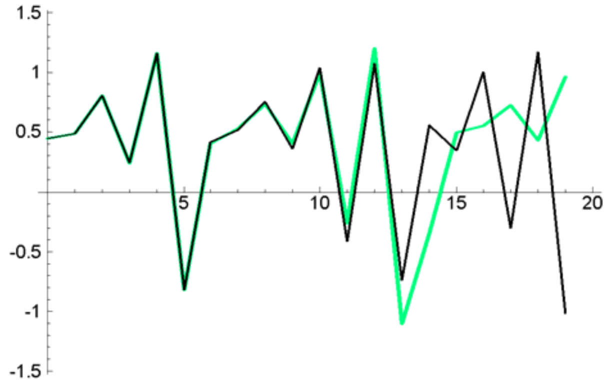

Figure 2. Comparison of Forecast and Henon Data.

In Figure 2, the forecast is shown in black while the real data is lighter. Notice that the forecast does quite well for about 10 steps but eventually diverges from the true data.

Sunspots

As a second illustration we will consider fractal forecasting applied to sunspot data. Sunspot observations have been recorded for over 400 years and can be estimated from other phenomena [4]. We use 400 years worth of data available in a script associated with [3].

```
   load 'fvj2\ts_data.ijs'

   $sunspots
400
```

Figure 3 shows 200 years worth of sunspot data. Notice the periodicity, yet near repetitions are not obvious.

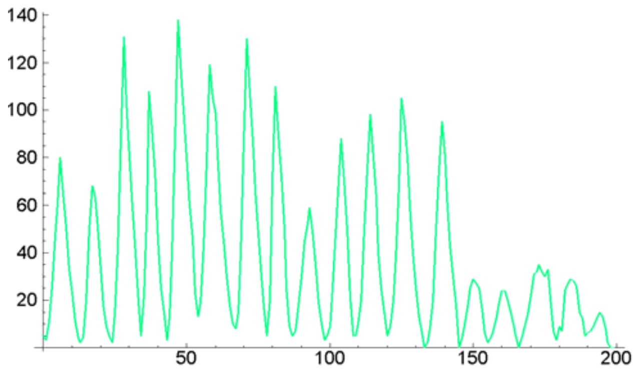

Figure 3. Sunspot Data

Sunspot cycles are around 11 years, so we will use a width of 12 for our reference and sample points. We use the 20 closest reference points to make our estimate and attempt a 25 year forecast as follows. The results are shown in Figure 4.

```
   w=:12

   sam=:(351+i.w){sunspots

   refdat=:350{.sunspots

   pred=:20 25 ffcast refdat;sam

   real=: (w+351+i.25){sunspots
```

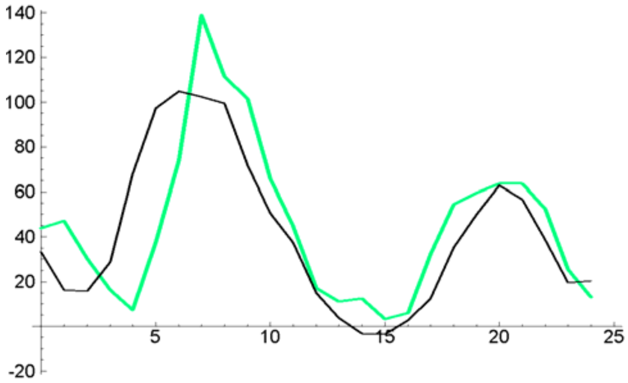

Figure 4. Comparison of Sunspot Data and Forecasts.

While the general trend of sunspot activity is followed, the forecasting seems quite sensitive to the window width and number of reference points used. Indeed, since the pattern diverges more on the earlier steps, the fit shown in that figure involves luck. Still, the technique seems to follow the cyclic pattern fairly reliably and gets the scale of the peaks correct with reasonable frequency.

Like any forecasting technique, fractal forecasting is not magic and is only appropriate in certain contexts. However, it seems to be remarkably able to utilize subtle nonlinear relationships in data provided there is an ample historical data base and fractal forecasting is quite easy to implement.

References

[1] S. D. Balkin, Fractal and Chaotic Time Series Analysis, *Master of Arts Paper*, The Pennsylvania State University, 1996.

[2] M. Casdagli, Chaos and Deterministic versus Stochastic Non-linear Modeling, *Journal of the Royal Statistical Society, Series B*, **54** 2 (1992) 303-328.

[3] C. A. Reiter, Fractals, *Visualization and J, 2nd Edition*, Jsoftware, Inc., Toronto, 2000.

[4] D. J. Schove, *Sunspot Cycles*, Hutchinson Ross Publishing, New York, 1983.