ficiently accurate, we can use it to obtain statistical estimates that are more accurate than those obtained directly from the data itself. There are several possibilities for doing this. For example, suppose we wish to compute the dimension of an attractor, and we are limited by the amount of available data. If we have a good model for the data, we can iterate the model to obtain new "simulated data". If the measure of the simulated data is sufficiently close to that of the true data, then we can simply apply standard algorithms to compute the dimension of the new time series. This can similarly be done to compute quantities such as the Lyapunov exponents.

A central question is how much the the data can be extended and still produce accurate estimates on a given scale. For short prediction times, the error of approximation is roughly $\bar{E} \sim N^{-\frac{q}{D}}$. If we only needed to extend the data by a few data points, we can expect these new data points to be faithful to the measure to this accuracy. As we iterate the map further, although each estimate is only for a short time, the errors in the distribution of the points may accumulate. We do not understand how to estimate this in detail, but since the evolution of the measure depends on the derivatives of the flow, we think that the accuracy will depend on the accuracy for estimating derivatives. [18] For local schemes the order of approximation for the derivative is typically one less than that of the map itself. Thus, we hope that averages computed in this way should also be accurate to this same degree of approximation. These arguments are admittedly very vague, however, and need to be made more precise, and backed up by numerical results.

Assuming that we can approximate the measure to order $q-1$, this implies that we can generate roughly $\tilde{N} \sim N^q$ new data points and still remain faithful to the true dynamics. We can then improve our estimate of the dimension by applying the usual ball scaling algorithms to the newly generated points. Since there are now N^q of them, we can get a much better estimate. Of course, this involves more computation. Nonetheless, if the computation is limited by available data rather than computer resources, this implies that the analysis of a D dimensional attractor can be done as accurately as a $\frac{D}{q-1}$ dimensional attractor using first order methods.

Similarly, we can use this same method to improve estimates of the Lyapunov exponents. An algorithm for doing this is originally due to Wolf *et al.* [77], and alternatives have been suggested by Eckmann and Ruelle [21,20] and Sano and Sawada [66]. Since the Wolf algorithm involves manipulation of existing trajectories, while the Eckmann/Ruelle/Sano/Sawada algorithm employs local linear maps, on the surface it might seem to involve a higher order of approximation than the Wolf algorithm. However, since the estimate of derivatives through a linear map is only first order, in fact the order of approximation is the same.

If a higher order approximation can be found, the accuracy of these estimates can be improved in two ways. First, the estimates of local derivatives become more accurate. Second, by generating more data we can expect to get better statistics. Assuming the local derivatives are independent, the error due to statistical fluctuations goes as $\sigma \tilde{N}^{-1/2}$, where σ is the standard deviation of the local contributions,

[18] We thank Martin Casdagli for helping to clarify this point.

and $\tilde{N}$ is the number of simulated data points. The accuracy of approximation for the individual derivatives scales as $E \sim N^{-\frac{D}{q-1}}$, where N is the number of true data points. The dominant effect depends on the order of approximation as well as the dimension of the attractor, and whether the estimates for the derivatives are biased.

Alternatively, as we showed in reference [23], scaling properties of error estimates can be used to estimate the dimension. If the order of approximation q is known, then Equation (17) can be turned around to give a relationship for the dimension.

$$D = \lim_{N\to\infty} q \frac{\log N}{|\log \bar{E}|} \tag{47}$$

There are several problems with this method. First, the biggest potential improvement comes about when q is large. For this method to give a reliable estimate of the dimension we must know the order of approximation *a priori*. Our numerical work so far indicates that it is difficult to achieve orders of approximation larger than two reliably when D is large. Unless we can be certain in advance of achieving a given order of approximation, we cannot estimate the dimension this way. Furthermore, the statistical stability of an estimate of $\bar{E}$ is limited by the number of points in the time series, and for a small time series statistics are not good. Thus, while checking for the proper scaling is a good measure of self consistency, we expect that iterating to generate "simulated data" will give superior estimates of the dimension.

4.3 Forecasting as a measure of self-consistency

Forecasting provides a hard test for the presence of chaos, especially when combined with the tests for the scaling properties expected from the error estimates of Section 3. It is very unlikely to make forecasts with the statistical significance that we achieve here by guessing at random. Although it is possible to construct counterexamples that would appear very much like chaos by running high dimensional noise through appropriate nonlinear filters, we doubt that such contrived examples are very likely. To paraphrase Joe Ford, "If it walks like a duck, talks like a duck, quacks like a duck, and even smells like a duck, then by golly, it seems pretty reasonable to assume it really is a duck."

5 Noise Reduction

In this section we introduce a new method for reducing noise in a dynamical system. The basic idea is nonlinear smoothing; once we can make forecasts, we can transport different points to the same point in time and average them together, to reduce the effect of noise. By applying this procedure recursively the reduction can be substantial. The basic idea is schematically illustrated in Figure (11).

We will assume that the time series $\{x_t\}$ is of the form

$$x_t = y_t + n_t, \tag{48}$$

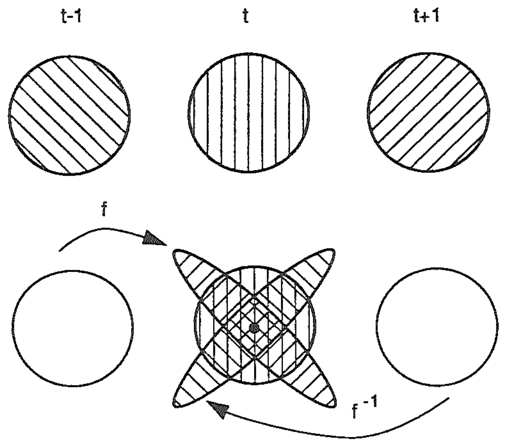

Figure 11: *Nonlinear smoothing.* The circles represent noisy measurements of a deterministic trajectory at three different times. As successive measurements are transported to the same point in time, the associated noise probability distributions distort. By weighting the values of the transported points correctly, they may be averaged together to produce a better estimate of the true value y_t. The estimates can be improved by iterating this process.

where $y_{t+1} = f(y_t)$, and the sequence n_t is uncorrelated, with $\langle n_i n_j \rangle = \sigma^2 \delta_{ij}$. We assume that n_t is unpredictable, and values at different times are independent. We will also assume that $\{n_t\}$ has a symmetric Gaussian distribution,

$$P(n_t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-n_t^2/2\sigma^2}. \qquad (49)$$

This assumption is convenient, but the final result does not depend on it; for example, our numerical results demonstrate that our method works quite well for noise with a bounded uniform distribution.

A standard method for reducing noise is to simply average together nearby points. For this to be effective, the average must be taken over a very short time interval, since otherwise the intrinsic dynamics of the signal dominate the noise. But if we can forecast accurately we can take the dynamics into account, and average over much longer periods of time. As we shall see, this is particularly effective in chaotic systems.

We want to transport points at different times to the same time, as illustrated in Figure (11). Conceptually, we are pulling-back and pushing-forward the probability distribution of the noise, under the action of the deterministic dynamics f. A probability distribution has an induced transformation under the mapping f that we will denote $\underline{f}$, *i.e.*,

$$P(x_{t+1} - y_{t+1}) = \underline{f}(P(x_t - y_t)).$$

Suppose we transport measurements from times $\{t-\alpha, \ldots, t+\beta\}$ to time t. Since n_t is independent, the joint probability distribution is

$$\overline{P}(f^{\alpha}(x_{t-\alpha}), \ldots, f^{-\beta}(x_{t+\beta})) = A \prod_{j=\alpha}^{j=-\beta} \underline{f}^j(P(x_{t-j} - y_{t-j})), \qquad (50)$$

where A is a normalization constant and $\alpha \geq 0$ and $\beta \geq 0$. If we assume that n_t is small we can linearize $\underline{f}$,

$$\underline{f}(P(x_t - y_t)) \approx A'P((f(x_t) - f(y_t))df(x_t)^{-1}) = A'P((f(x_t) - y_{t+1}))df(x_t)^{-1}), \qquad (51)$$

where $df(x_t)$ is the derivative of f at x_t and A' is a normalization constant. This approximation depends on the assumption that the noise is small compared to the nonlinearities of f, so that the dynamics is locally linear (so equiprobable surface remain ellipsoids under transformation, as shown in Figure (11)). Combining equations (49), (50), and (51) gives

$$\overline{P}(f^{\alpha}(x_{t-\alpha}), \ldots, f^{-\beta}(x_{t+\beta})) \approx A \prod_{j=\alpha}^{j=-\beta} e^{-\|(f^j(x_{t-j}) - y_t)[df^j(x_{t-j})]^{-1}\|^2/2\sigma^2} \qquad (52)$$

We want to estimate y_t. A standard way to do this is to make a maximum likelihood estimate, which amounts to assuming that the particular sequence of fluctuations that we observe are the most likely ones, so that they lie at the maximum of the joint probability distribution. We choose our estimate Y_t to force this to be true. Since $\log \overline{P}$ has the same maximum as $\overline{P}$, we can more conveniently enforce this by setting $\frac{\partial \log \overline{P}}{\partial Y_t} = 0$. Setting $Y_t = y_t$ in Equation (52), differentiating, and solving for Y_t yields

$$Y_t = \left(\sum_{j=\alpha}^{j=-\beta} \Theta_j \right)^{-1} \sum_{j=\alpha}^{j=-\beta} \Theta_j f^j(x_{t-j}) \qquad (53)$$

where

$$\Theta_j = \left([df^j(x_{t-j})]^T df^j(x_{t-j}) \right)^{-1}.$$

Θ_j is a $d \times d$ symmetric matrix that depends on x_{t-j}. It contains weighting factors that depend on local expansion and contraction rates, and take into account distortion of the noise as it is transported to different times. The directions in which the noise distribution is compressed contain more useful information, and receive higher weights.

To implement this procedure we have to estimate both $f^j(x_{t-j})$ and $df^j(x_{t-j})$. In practice, because of nonlinearities it is wise to keep the smoothing times short by keeping α and β fairly small. Further reductions are made by applying Equation (53) recursively. Since this makes it possible to keep each step short, this minimizes the effect of nonlinearities. With every pass we reduce the noise level, so that the local linear assumption of Equation (51) becomes increasingly valid. Thus once the algorithm starts to converge, further convergence is guaranteed. The recursive use of this algorithm is reminiscent of the "pull-back" algorithms for estimating Lyapunov exponents [4,69].

When we know the map exactly, with even a small amount of data and fairly large noise we can reduce the noise almost down to machine precision. To demonstrate this, we have applied this to the Henón map, as shown in Figure (12). Note that we achieve a noise reduction of roughly 10^{10}, more than 100 decibels. When the true map is not known, this is limited by forecasting accuracy. Still, even when we approximate the map we have been able to reduce the noise by a factor of 1000 or more.

As is apparent in the figure, points near either end of the time series are not smoothed nearly as accurately as those in the middle. The reason is that in a chaotic system the pulled-back values are accurate along the unstable manifold, while the push-forward values are accurate along the stable manifold. Points near the beginning of the time series have no history, and therefore noisy fluctuations along the stable manifold cannot be reduced. Similarly, points near the end of the time series have no future, and fluctuations along the unstable manifold cannot be reduced. For the Henón map, for example, as we move j steps from the beginning of the time series toward the middle, the noise is reduced along the stable manifold by a factor of roughly $\Lambda_s{}^{-j}$, where Λ_s is the Lyapunov number associated with the stable manifold. Similarly, as we move j steps from the end toward the center the noise is reduced by roughly $\Lambda_u{}^j$, where Λ_u is the Lyapunov number associated with the unstable manifold (and is greater than one).

So far we have assumed that the noise was added to an ideal trajectory, and had no effect on the dynamics. Suppose that instead the noise is coupled to the dynamics, and is included in computing the next state. In this case, there is no unique "true" trajectory. This leads to the shadowing problem: Is every noisy trajectory "shadowed" by a nearby deterministic trajectory? For hyperbolic systems, Anasov and Bowen independently proved that this true [1,8]. Although Anasov and Bowen discussed only hyperbolic systems, recently Hammel *et al.* [42] have applied a modified version of the Anasov-Bowen algorithm to simple systems such as the Henón map, demonstrating that they can usually find shadowing trajectories close to noisy trajectories. Although they posed the problem in a different context, their work can be viewed as noise reduction. [19] Our method is reminiscent of the Anasov-Bowen procedure, but it is applicable not only to non-hyperbolic systems, but even *non-chaotic* systems. Our method is also easy tò implement in any dimension.

It is ironic that it is much easier to remove noise from a chaotic time series than from a regular time series. Without exponential expansion and contraction, according to the central limit theorem, smoothed values only become more accurate according to the square root of the number of data points, in contrast to the exponential behavior of chaotic systems. The number of points needed to achieve a given level of noise reduction for regular dynamics is therefore much larger than for chaotic dynamics.

The limit to noise reduction is usually given by the accuracy for approximating f.

[19] Recently this approach has been modified by E. Kostelich and J. Yorke to make it more practical (private discussion).

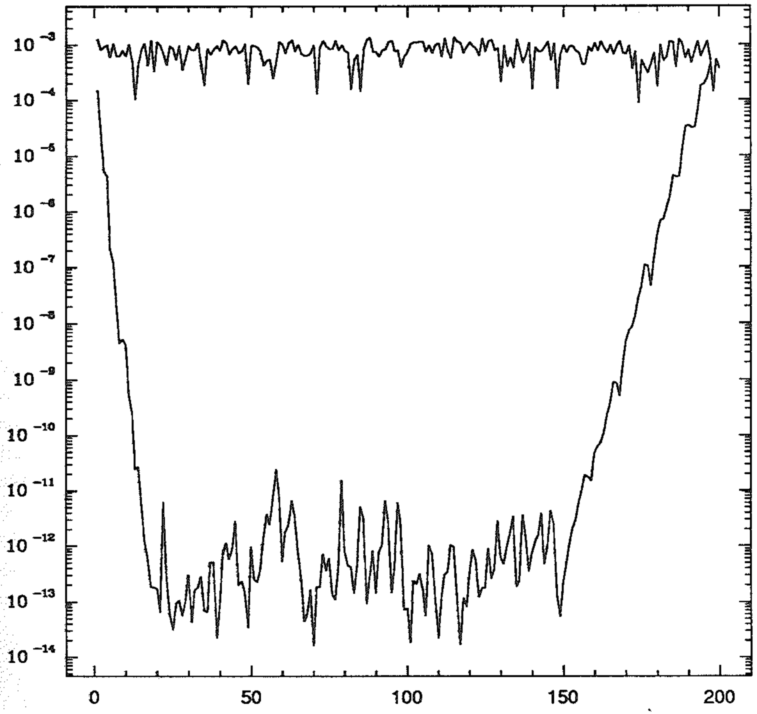

Figure 12: The noise reduction technique of Equation (53) applied to the Henón map. The map was iterated at double precision, generating a "clean" time series $\{y_t\}$, $t = 0, 1, \ldots, 200$. We then added pseudo-random numbers $\{n_t\}$ to each point, forming a noisy time series $x_t = y_t + n_t$. n_t is uniformly distributed, with a variance roughly 0.1% of that of y_t. The logarithm of $|n_t|$ is plotted at the top of the figure. Below it we plot $\log|Y_t - y_t|$, where Y_t is the smoothed value after applying Equation (53) 10,000 times to all 200 points, using $\alpha = \beta = 3$, and using the known map to compute f^j and Θ_j.

Estimates of Θ are not as important, since it is just a weighting matrix. As long as the eigendirections are roughly correct, we get a net noise reduction on every iteration. The approximation error for f is helped by the fact that the directions where we need accuracy are *contracting*; we need to forecast the stable manifold accurately forward in time, and the unstable manifold accurately backward in time. There is little decay of predictability with time, and forecasts may even improve! We do not have to fight the exponential divergence of trajectories to reduce noise.

The main numerical problem comes from homoclinic tangencies, *i.e.*, places where the stable and unstable manifolds are in the same direction. In Figure (11), for example, this would mean the two ellipsoids have the roughly the same principal axis. An exact homoclinic tangency causes the same problem that occurs at the ends of the time series – noise reduction is only possible along one direction. In general, however, homoclinic tangencies are not exact, and even if they are exact, for small α or β the orientations may vary, so that some combinations of α and β give better results than others. Also, the reduced noise levels elsewhere in the time series are transmitted to the bad points, and if they are strong enough they can overcome the instabilities.

Nonuniformities may also cause similar problems, if the signs of the eigenvalues of Θ are atypical. For example, if one Lyapunov number is greater than one and the other is less than one, at a point x_k where both eigenvalues of Θ_1 are greater than one, the push-forward of $P(x_k)$ will expand in both directions. This can be cured by reaching further into the past, increasing α until Θ_α acquires an eigenvalue less than one.

6 Adaptive Dynamics

There has been a great deal of interest recently in solving artificial intelligence problems with *adaptive networks* such as neural nets [65,15] and the classifier system [44]. Although on the surface the straightforward approximation techniques that we employ here seem quite different from neural nets, the underlying principles are actually much the same. However, since our representations are more convenient numerically, fitting parameters is hundreds of times faster.

Forecasting is an example of what is often called *learning with a teacher*. The task is to predict "outputs", based only on "inputs". For forecasting the input is the present state and the output is the future state. The record of past states provides a set of known input-output pairs which acts as a "teacher". The problem is to generalize from the teaching set and estimate unknown outputs.

We can restate the problem more formally as follows: Given an input x_i and an output y_i, we want to find maps F and G of the form

$$\begin{aligned} \hat{y}_i &= F(x_i, \alpha_i) \quad (54) \\ \alpha_{i+1} &= G(x_i, \hat{y}_i, y_i, \alpha_i) \quad (55) \end{aligned}$$

that minimize $\|y_i - \hat{y}_i\|$, where $\hat{y}$ is an estimate of y, and the metric $\|\ \|$ provides a criterion for the estimation accuracy. α_i are parameters for F, and G is a map that changes α_i, *i.e.*, a *learning algorithm*. x, y and i can be either continuous or discrete. For a forecasting problem, for example, i corresponds to time, x is the current state x_t, and y is a future state, $y_t = x_{t+T}$.

Neural nets correspond to a particular class of functional forms for F and G. Although this form was originally motivated by biology, there is no reason to be constrained by this in artificial intelligence problems, as reflected by many recent developments in this field. Neural nets have had success in certain problems that can be solved by learning with a teacher, for example, text to speech conversion [67] or finding gene locations on DNA sequences [48]. Lapedes and Farber have also shown that neural nets can be effective for forecasting [49].

However, as recently pointed out by Omohundro [58], alternative approaches that depart significantly from the usual form of neural nets may be computationally much more efficient. Our approach to forecasting provides a good example of this; our methods give equivalent or more accurate forecasts than the neural net of Lapedes and Farber [49], and are several orders of magnitude more efficient in terms of computer time. Furthermore, since the computations can be performed in parallel, we expect that this speed discrepancy will persist even with future parallel hardware. Omohundro has pointed out that similar methods may be employed for other problems, such as associative memory, classification, category formation, and the construction of nonlinear mappings. Many aspects of the methods that we have proposed here are applicable to this broader class of problems.

Although we have assumed that x and y are continuous, with the addition of thresholds our methods are easily converted to the discrete domain. Our work, taken together with that of Lapedes and Farber, makes it clear that the neural network solves problems by surface estimation. They show that the same is true in the discrete domain, except that answers are obtained by "rounding" the surface, truncating to a discrete value. Generalization occurs through the extrapolation of the surface to regions in which there is no data [47]. There is no *a priori* reason to constrain the functional representation to those that are currently popular in the study of neural networks.

Radial basis functions provide a particularly promising possibility. One of the key properties of the two layer tanh network is *localization*; the composition of two tanh functions forms a well-localized bump, and by adding these together it is possible to represent arbitrary functions [47]. Radial basis functions are designed to be interpolants with good localization properties, and so should be ideal replacements for the tanh. Since their parameters can be fit through linear least squares, unique solutions for radial basis functions can be found very quickly. Furthermore, as recently shown by Casdagli [12], under favorable circumstances radial basis functions can achieve orders of approximation as high as six.

Clearly these possibilities deserve more investigation.

7 Conclusions

By assuming that a random process is produced by deterministic chaos, finding a good model reduces to two parts: (1) Finding a state space embedding that maximizes determinism, and (2) fitting a nonlinear functional form to the map that sends current states to future states.

The importance of the first problem should not be underestimated. The usual time consuming procedure of searching for a good embedding by trial and error is far from optimal, both because it is time consuming and because the results are not necessarily ideal. Some improvements on this have been suggested by Fraser and Swinney [30] and by Broomhead and King [11]. We have suggested an improvement on the technique of Broomhead and King which eliminates the last free parameter. We intend to address this problem in more detail in the future.

The next problem is to approximate the dynamics from the data. The primary approach we investigate here is approximation as a discrete time map, which has the advantage of being convenient and fast. Approximation in differential terms may promise more accuracy, however, and we intend to compare these two approaches in a future paper. In either case, the problem boils down to approximating the graph of a nonlinear function. Success depends on picking a good representation. There are two basic approaches, global and local. Global approximation is convenient, but unless the representation is well matched to the map it may not produce good results, especially since many of the standard nonlinear representations undergo an explosion of parameters as the dimension increases. Local approximation has the advantage that it is less dependent on representation, and is guaranteed to get better as the number of data points increases. When used in conjunction with a data structure such as the k-d tree, it can be quite fast. The best approach depends on the details of the problem, such as the nonlinear function being approximated, the number of data points, *etc.*

An advantage of formulating the forecasting problem in the language of deterministic chaos is that it makes it possible to derive error estimates. These estimates are couched in terms of properties of the dynamics, such as the Lyapunov exponents and the dimension, the length of the data set, its signal to noise ratio, and the extrapolation time. We arrive at the conclusion that iterative forecasts are better than direct forecasts, *i.e.*, it is better to make long-term forecasts by approximating the dynamics for a short time and iterating rather than approximating directly. The iterative approach takes advantage of the recursive form of the higher iterates of dynamical systems. With the iterative approach approximation errors grow at the same rate as errors due to uncertainty in the initial state, *i.e.*, they grow exponentially according to the largest Lyapunov exponent. These results are derived in the limit as $\bar{E} \to 0$; we have observed some counterexamples. Note that in order to study the behavior of direct forecasts we had to introduce the new concept of higher order Lyapunov exponents.

Having a model of the dynamics extends all the numerical techniques that were previously available only in numerical experiments to the analysis of data in real experiments. Furthermore, when it is possible to achieve higher orders of approximation, it becomes possible to extend the available data and obtain much more accurate results than would otherwise be possible. We have given some suggestions for this, but many questions remain to be investigated.

The ability to approximate nonlinear dynamics naturally leads to a method for reducing external noise through nonlinear averaging. When the dynamics are known exactly, this technique makes noise reductions of as much as ten orders of magnitude possible. When the dynamics must be learned the limitation on this technique comes from the the accuracy of the model. However, for low dimensional systems with a modest number of data points we can produce noise reductions of several orders of magnitude. Surprisingly, noise reduction is much easier for chaotic motion than for regular motion.

All of the methods discussed above work well for low dimensional deterministic chaos. When the dimension is low they give results that are orders of magnitudes better than those of standard linear methods. However, they lose their effectiveness when the dimension is too large. The limits can be estimated through the error estimate of Equation (34). As seen from this equation, they also depend on the method used: For a given number of points higher order approximation is more accurate than low order approximation. Of course, if we have extra information, such as the functional form of the dynamics, it may be possible to overcome the constraint of large dimensions.

The methods we have described here are new and not fully explored. We anticipate that there will be considerable progress in this area in the near future.

Acknowledgements We would like to thank Martin Casdagli, Ute Dressler, Pat Hagan, Alan Lapedes, Y.C. Lee, Steve Omohundro, Norman Packard, David Rand, Rob Shaw, and Blair Swartz for valuable conversations. We are grateful for support from the Department of Energy and the Air Force Office of Scientific Research under grant AFOSR-ISSA-87-0095.

We urge the reader to use these results for peaceful purposes.