Other approaches:

Once you have projected to higher dimensions, you can use neural networks as well.

KNN approach to time-series analysis:

Assuming that a d dimensional embedding has already been chosen, the only remaining variable in the procedure is to determine the number of neighbors to be used to create the map. If the number of neighbors M is chosen to be the total number of training data points, the algorithm reduces to a global linear fit. By choosing M to be relatively small, the degree of locality can be varied.

Dimension	Lag = 1	Lag = 3	Lag = 5
5	0.9157	0.8184	0.8375
10	0.7665	0.8213	0.9509
15	0.7041	0.9257	1.1195
20	0.6849	1.0355	1.2041
25	0.7159		

Number of neighbors = 2d.

Window Size and Sampling Lag

Given a time-series xt, xt-1, ..., xt-i,... in order to predict xt+n, we have to decide how many previous data points were needed, which we call the window size. We also had to decide how to sample the past value. A constant lag sampling consists of:

xt, xt-k, xt-2k, ..., xt-ik,...

where k is called sampling lag.

Direct vs. Iterative Approaches

When predicting multiple steps into the future, we have several choices. One approach is to directly predict xt+n from the sampled data.

Alternatively, we can construct function F1 to make 1 step ahead prediction first and then to use as if it were given, and iteratively apply function n times to predict Still another approach is to start with F1, but when moving one step forward, re-train it to get a new function F1’

Measures for Prediction Errors

Given a time-series x1, x2, ..., xn, the standard deviation is essentially the ``mean prediction error" if the mean of the time-series is always used as prediction. This is the best one can do if the series is a random walk. If we have a series of predictions then the prediction error is defined as:

error = sqrt(sum(x_i - x^_i)^2/(n-1))

Often we use the ratio of prediction error over standard deviation to measure the quality of the predictions. It basically shows the improvement over simply predicting the mean of the series.

Prediction Strategies

Training and Test Data Sets

the available data was divided into three sets:

Training Data Set - used to train a prediction system; its size varies in different experiments in order to find the optimal size.

Cross Validation Set - used to monitor the training, e.g, to decide whether the prediction system has overfitted the training set.

Test Set - only used at the very end, to test the true performance of the system trained.

Dimensional analysis of time-series data

How many dimensions should I use in the delay space embedding?

Use the Grassberger-Procaccia algorithm.
In essence, this algorithm attempts to see how well correlated you are with some point in the past.

If you are past the periodicity, you are okay and you dont need to increase the dimensionality any more.

Simple Exploratory Data Analysis

Is the series stationary / periodic?

- running means and running standard deviations.

- compute autocorrelations over lags.

The data set

The time-series was synthetically generated by numerically integrating the equations of motion for a damped, driven particle in the potential field:

$$V = a_4 (x_1^2 + x_2^2 + x_3^2 + x_4^2)^2 - a_2\sqrt{x_1^2 + x_2^2} - a_1 x_1$$

The value of a_1 has a small drift produced by integrating a Gaussian random variable.

The equations of motion were integrated with a simple fixed-step 4th order Runge-Kutta routine. 100,000 data points were generated. There are several advantages of using this time-series:

Since it is synthetically generated, in principle everything about it is known. Thus we can check what kind of information and correlation can various prediction algorithms can discover. Also since no background information was provided initially, this is a ``fair game" for all algorithms used.

The time-series contains relatively high-dimensional dynamics. The configuration space has 9 degrees of freedom (4 position, 4 velocity, and 1 forcing time), which was picked to be large but still realistically accessible. A long time-series is available. To recognize the presence of such high-dimensional dynamics, large data sets are needed, and can obviously be easily generated here. This often is a problem with real world data.

To simplify our discussion, we will focus on 5-step ahead prediction, i.e., given , to predict x_t, x_{t-1}, ..., x_{t-n},... to predict x_{t+5}. The same techniques can be extended to other cases.

Non-Linear Time-Series Prediction: An Example

There are several important issues in time-series prediction:

How large should the window size (dimension) be?
What are the sampling lags for the past data?
How can one predict multiple steps toward the future?

How many neighbors should be used?
How large a training set is needed?
What sort of distance metric should be used?

So what can we do?

There are many time-series problems in science and engineering that are to some extent predictable (such as the occurrence of sun spots, or the failure rate of a product), and further improvement on their prediction accuracies is of great importance.

How do we use k-nearest neighbor like techniques for time series analysis?

Let us try to predict the behavior of a time-series generated by a chaotic dynamical system. The training set consists of a time-series of up to 10,000 samples (say) from a trajectory along the attractor. The time-series is then transformed into a reconstructed state space using a delay space embedding. In the delay space embedding, each point in the state space is a vector X composed of time-series values corresponding to a sequence of d delay lags: x1(t) = x(t), x2(t) = x(t-tau), ..., xd(t) = x(t - (d-1) tau). To forecast a time-series value, we first transform the value into the state space representation. The nearest k (> d) neighbors in the state space representation are then located. A local linear map is created for the k neighbors and applied to the value to be forecast. The result of the mapping is the predicted value. Although higher dimensional maps (quadratic, etc.) could be used, no significant improvements over the linear map are noticed. Using this approach, people have been able to forecast time-series values for a number of systems (Mackey-Glass differential equation, Rayleigh-Benard convection, and Taylor-Couette flow) much more accurately than standard forecasting techniques.

What are cellular automata?

A Cellular automaton (CA) is a dynamical system with discrete time, discrete state space, and discrete geometrical space (like an ODE). Thus they can be represented by a state s(i,j) for spatial state i, at time j, where s is taken from some finite set. The update rule is that the new state is some function of the old states, s(i,j+1) = f(s). The following table shows the distinctions between PDE's, ODE's, coupled map lattices (CML) and CA in taking time, state space or geometrical space either continuous (C) of discrete (D):

	time	state space	geometrical space
PDE	C	C	C
ODE	C	C	D
CML	D	C	D
CA	D	D	D

Perhaps the most famous CA is Conway's game "life." This CA evolves according to a deterministic rule which gives the state of a site in the next generation as a function of the states of neighboring sites in the present generation. This rule is applied to all sites.

Is there chaos in the stock market?

In dynamical systems theory, chaos means irregular fluctuations in a deterministic system. This means the system behaves irregularly because of its own internal logic, not because of random forces acting from outside. Take the whole system together and you have one HUGE chaotic system.. but do we really care?

To be useful, economic chaos would have to involve some kind of collective behavior which can be fully described by a small number of variables. In the lingo, the system would have to be self-organizing, resulting in low-dimensional chaos. If this turns out to be true, then you can exploit the low-dimensional chaos to make short-term predictions. The problem is to identify the state variables which characterize the collective modes. Furthermore, having limited the number of state variables, many events now become external to the system, that is, the system is operating in a changing environment, which makes the problem of system identification very difficult.

How do I know if my data are deterministic?

It is difficult because in practice no time series consists of pure ‘signal.’ There will always be some form of corrupting noise, even if it is present as roundoff or truncation error or as a result of finite arithmetic or quantization. Thus any real time series, even if mostly deterministic, will be a stochastic processes.

All methods for distinguishing deterministic and stochastic processes rely on the fact that a deterministic system will always evolve in the same way from a given starting point. Thus given a time series that we are testing for determinism we

(1) pick a test state
(2) search the time series for a similar or ‘nearby’ state and
(3) compare their respective time evolution.

Define the error as the difference between the time evolution of the ‘test’ state and the time evolution of the nearby state. A deterministic system will have an error that either remains small (stable, regular solution) or increase exponentially with time (chaotic solution). A stochastic system will have a randomly distributed error.

Can computers simulate chaos?

Strictly speaking, chaos cannot occur on computers because they deal with finite sets of numbers. Thus the initial condition is always precisely known, and computer experiments are perfectly predictable, in principle. In particular because of the finite size, every trajectory computed will eventually have to repeat (an thus be eventually periodic). On the other hand, computers can effectively simulate chaotic behavior for quite long times (just so long as the discreteness is not noticeable).

What is sensitive dependence on initial conditions?

Consider a boulder precariously perched on the top of an ideal hill. The slightest push will cause the boulder to roll down one side of the hill or the other: the subsequent behavior depends sensitively on the direction of the push--and the push can be arbitrarily small. Of course, it is of great importance to you which direction the boulder will go if you are standing at the bottom of the hill on one side or the other!

Sensitive dependence is the equivalent behavior for every initial condition--every point in the phase space is effectively perched on the top of a hill.

What is phase space?

Phase space is the collection of possible states of a dynamical system. A phase space can be finite (e.g. for the ideal coin toss, we have two states heads and tails), countably infinite (e.g. state variables are integers), or uncountably infinite (e.g. state variables are real numbers).

What is chaos?

Effectively unpredictable long time behavior arising in a deterministic dynamical system because of sensitivity to initial conditions. It must be emphasized that a deterministic dynamical system is perfectly predictable given perfect knowledge of the initial condition, and further is in practice always predictable in the short term. The key to long-term unpredictability is a property known as sensitivity to (or sensitive dependence on) initial conditions.

For a dynamical system to be chaotic it must have a ‘large’ set of initial conditions which are highly unstable. No matter how precisely you measure the initial condition in these systems, your prediction of its subsequent motion goes radically wrong after a short time. Typically, the predictability horizon grows only logarithmically with the precision of measurement. Thus for each increase in precision by a factor of 10, say, you may only be able to predict two more time units.

Time series analysis:

What is Time-series analysis?

This is the application of dynamical systems techniques to a data series, usually obtained by "measuring" the value of a single observable as a function of time.

What is a dynamical system?

A dynamical system consists of an abstract phase space or state space, whose coordinates describe the dynamical state at any instant; and a dynamical rule which specifies the immediate future trend of all state variables, given only the present values of those same state variables. Mathematically, a dynamical system is described by an initial value problem.

Dynamical systems are "*deterministic*" if there is a unique consequent to every state, and "*stochastic*" or "*random*" if there is more than one consequent chosen from some probability distribution (the "perfect" coin toss has two consequents with equal probability for each initial state).