Fast Forecasting of Time Series Data

Deepayan Chakrabarti
Center For Automated Learning and Discovery
deepay@cs.cmu.edu

February 5, 2001

Abstract

A time sequence is an ordered set of measurements of some observable over time. When the underlying dynamical system isn't fully understood, observed regularities in the time series can provide insights into the inner working of the system. Given a time series from the past, the aim of a prediction algorithm is to forecast the expected future of the time series by drawing analogies from previous behaviour, and thus, indirectly forecast the future behaviour of the system. I propose to work on methods for fast and efficient prediction of time sequences. Time sequences show up in a variety of natural and artificial systems, often under circumstances where the precise equations for the system are unknown. Thus, this area has wide applicability.

My approach uses a technique called *delay coordinate embedding*, in which the time series is broken up into equal-sized chunks which are used as indicators of the internal state of the system. Whenever a prediction needs to be made, the latest chunk is found, and the chunks "nearest" to it in "chunk space" are retrieved. By observing how the time series developed in each of these cases, its expected course in the current circumstances can be predicted. This involves several issues:

- Deciding the size of the chunks and their method of storage in a database
- Method for retrieving the best k-Nearest-Neighbour chunks
- Deciding the number k of nearest neighbours to be used, and a method of interpolation for combining the results from the k nearest chunks found
- Scalability to large databases (with Gigabytes of data)

As of now, there is already a prototype system which allows predictions on stock price data and on some transportation data. Chunk sizes were chosen on an ad hoc basis, and although the system is designed to be scalable, it has not been tested on such datasets. This project should result in the achievement of these aims.

1 Introduction

1.1 Problem Definition

Scientific research on physical systems is often based on the measurement of observables over time. Such data is the basis of characterization of the system and prediction of its future behaviour. If there are known underlying deterministic equations, they can be analyzed and system properties can be determined analytically or through simulations. But, in many real-world cases, the underlying equations for the system are unknown. Under such circumstances, regularities in the past have to be used as guides to understanding the system and forecasting the future.

Formally, the terminology involved is defined as follows:

- *Dynamical System:* This is a physical system which evolves over time.
- *System State:* This is the underlying state of the dynamical system under consideration, and is an unknown quantity.
- *Observable:* This is a function of the system state which can be be measured directly, or can be calculated from other measurements. If the system returns to the same state later, the observable will have the same value.
- *Time Series:* This is a list of values taken by the observable over time.

Given the time series data over some period, the problem is to predict the future evolution of the system. Some examples time series are shown in Figure 1.

1.2 Applications

The following gives a snapshot of the areas where time series have been shown up[4, 8]:

- *Physics:* Measurements of laser fluctuations.
- *Biology:* Physiological data, such as heart rate, blood oxygen concentration, and EEG state, of patients with sleep apnea.
- *Economics:* High-frequency currency exchange rate data, and daily stock quote data.
- *Astrophysics:* Intensity variations of white dwarf stars, prediction of solar activity.
- *Geophysics:* Measurements of magnetic storm data.
- *Transportation:* Highway usage-over-time data.

1.3 Deliverables

In this project, I plan to use a technique called *Delay Coordinate Embedding*, which is described in Section 3.1. The primary goal is to find formal methods of setting the various parameters that are needed in using this technique. Hence, the final product will be a working time series prediction system incorporating algorithms for setting of these

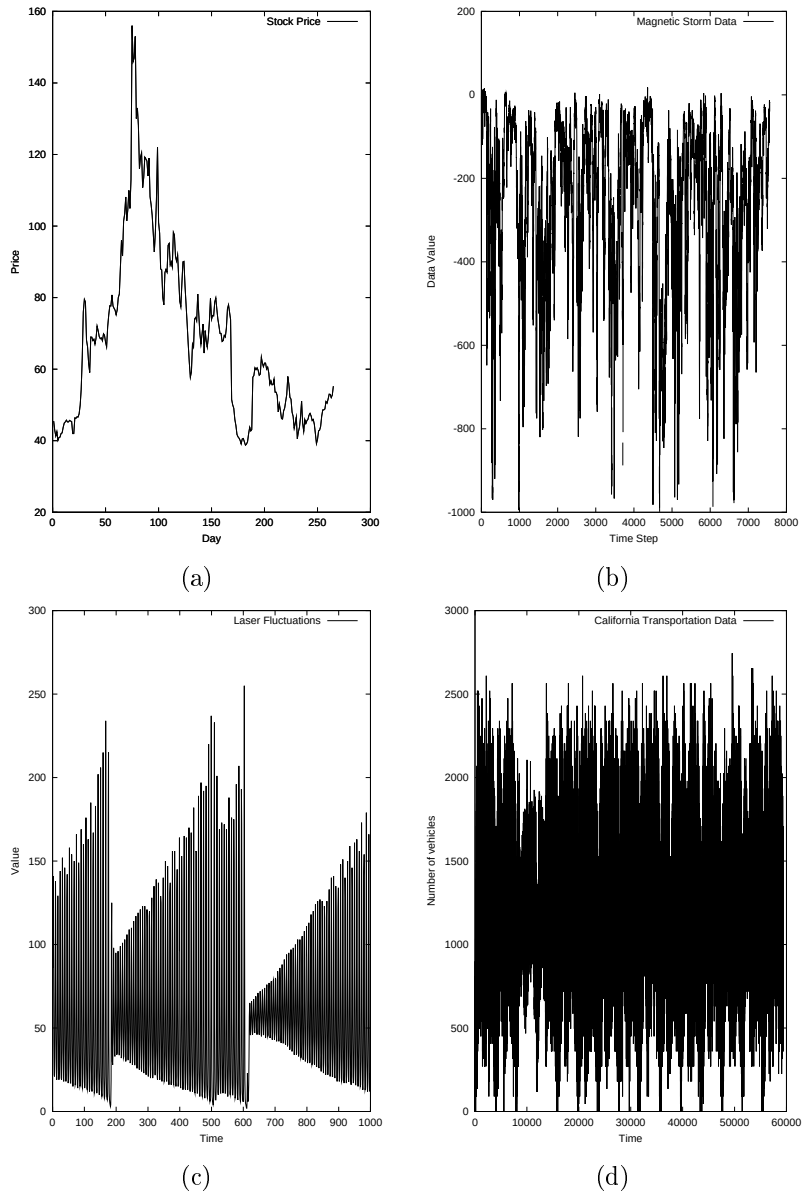


Figure 1: *Example time series*: Stock Price Fluctuations(a), Magnetic Storm Data(b), Laser Fluctuations(c) and California Transportation Data(d)

parameters. These algorithms will be run on the available time series as a preprocessing step, after which the system will be able to handle forecasting queries. The system will also have a validation system, which will be able to quantify the performance of the system in terms of computation costs and efficiency.

2 Related Work

Time-series prediction is a well-studied problem. Prior to the 1920s, forecasting was carried out by extrapolating the series through a global fit in the time domain. Then came the linear time series models, which attempted to describe the series as some linear function of internal and external variables. The advantages of linear models are twofold: they can be easily analyzed and understood in great detail, and, they are simple to implement. These models assume linearity and time invariance(ie., unchanging model coefficients over time) of the system. Brief descriptions of linear methods are given below. Reference texts for this are [6, 15].

One class of linear models is the Moving Average(MA) Models. These models assume that the system receives some external impulses which change the state of the system. Let the external input series be e_t and the output series be x_t.

$$x_t = \sum_{n=0}^{N} b_n e_{t-n} \tag{1}$$

The output is guaranteed to go to zero at N time steps after the input becomes zero. There is no feedback in the system. For the system to possess internal dynamics, some memory-property is needed. This is implemented by the following equation:

$$x_t = \sum_{m=1}^{M} a_m x_{t-m} + e_t \tag{2}$$

These are called the Autoregressive(AR) Models. The autocorrelation coefficient doesn't disappear after M steps, as opposed to MA models. Also, without any input, the output will finally either decay to zero, diverge, or oscillate periodically. Combining the previous two models gives the ARMA(M,N) model. Here, the equation for x_t is given by:

$$x_t = \sum_{m=1}^{M} a_m x_{t-m} + \sum_{n=0}^{N} b_n e_{t-n} \tag{3}$$

It has been widely used in all areas of time series analysis and discrete-time signal processing.

The downside of linear models is that they are capable of describing a time series *if and only if* the power spectrum is a useful characterization of the relevant features of the series. This can totally fail if nonlinearities lead to complicated power spectra. For example, natural systems are often driven by chaotic processes, which typically have a

continuous Fourier spectrum instead of a discrete range of frequencies. In such cases, other methods have to be used. Some such methods are described below.

One nonlinear prediction method is called the *delay coordinate embedding* method, and has been described in Section 3.1. Another approach has been through the use of Artificial Neural Networks [24, 25, 12, 23, 22]. Here, past values of time series are incorporated into the ANN. [22] describes a method where each network weight and neuron input is a vector instead of a scalar. The vector neuron input encodes previous values of the time series. The vector dot product $\mathbf{w}_{ij}^{l}.\mathbf{x}_{i}^{l}(k)$ is used instead of a scalar product (neuron i is connected to neuron j in layer l, k is the current time). A vector generalization of the backpropagation algorithm is used for learning from such networks. Another network structure for time series prediction is called the *Time-Delay Neural Network* [11, 21]. In these too, embedded time delays are incorporated into the network structure. A taxonomy of neural network structures for time-series processing is provided in [14]. But all these methods suffer from the traditional problems associated with neural networks: long training times and large number of parameters.

Hidden Markov Models(HMMs) have also been used [9]. There is a similarity between these models and the Delay-Coordinate Embedding technique. The notion of a hidden state in the former is similar to that of a *reconstructed state space* used in the latter technique (refer Section 3.1).

3 Approach

My proposed work is based on the delay-coordinate-embedding approach. One advantage of this approach is its firm mathematical basis [20, 18]. Another advantage is that this method avoids the need for a large number of free parameters, unlike such methods as ANNs, which leads to reduced time requirements, and perhaps more scalable algorithms. The thrust of my research will be on using concepts borrowed from several fields: chaos theory, fractal theory and spatial databases.

3.1 Delay Coordinate Embedding

The *state* of a deterministic system carries all the information necessary for predicting the behaviour of the system. Under minimal requirements on the smoothness of the underlying equations, the existence and uniqueness of the *state* holds. Thus, at any moment, the system has a state, and if a state is ever repeated later, then the future behaviour of the system will also be the same, within limits placed by noise. Thus, the internal system state can provide information for prediction.

Let the observation sequence be represented by the series x_t, which gives the value of the time series at time t. Let us define the following vector:

$$\mathbf{b} = [\mathbf{x_t}, \mathbf{x_{t-\tau}}, \mathbf{x_{t-2\tau}}, \ldots, \mathbf{x_{t-(m-1)\tau}}] \tag{4}$$

Here, τ is some real number greater than zero, and m is any integer greater than zero. The vector $\mathbf{b}$ is called a *delay coordinate vector* because its terms are the time-delayed data values from the time series. The technique of delay coordinate embedding is based on certain results [20, 18] which state that any structure that exists in the state space is carried over faithfully into the delay-coordinate-vector space, and that it is a one-to-one transformation. Thus, properties of the state space, such as equilibrium, periodicity, Lyapunov exponents (for measuring the degree of chaos) can all be observed in delay-coordinate space. Thus, the delay coordinate vector can be a useful representation of the internal state of the dynamical system. If, at two instances along the time series, the delay coordinate vectors are same (or very similar), then the data values following the two instances should be the same (or very similar). There is predictive power in finding this one-to-one map.

Hence, the problem of time series prediction is broken into two problems:

1. Finding the right length of the vector $\mathbf{b}$
2. Prediction of future values given the current delay coordinate vector

The first problem remains an art. For solving the second problem, the k-Nearest Neighbours $\mathbf{b}_1, \ldots, \mathbf{b}_k$ of the current state vector $\mathbf{b}$ are found, and the time-series values immediately succeeding these k instances (lets call them $p_1, \ldots, p_k$) are used to forecast the value after $\mathbf{b}$. One method is to just average over $p_1, \ldots, p_k$; another is to weight the p_i values by the exponential of the negative of the squared distance between $\mathbf{b}_i$ and $\mathbf{b}$. One particular implementation is given in [19]. Again, the determination of k is ad hoc.

3.2 Using Chaos

Recent times have witnessed an upsurge in the application of chaos theory to natural and artificial systems. Chaos lies logically between the domains of regular, predictable signals and totally unpredictable *stochastic* signals called *noise*. Chaos is irregular in time and is slightly predictable. It has structure in *phase space*. Chaotic strange attractors have been found in many environments. Thermal convection in the lower atmosphere was found to be chaotic [16, 13]. Laser fluctuations were found to be chaotic [3, 5, 2], as were the water volume fluctuations in the Great Salt Lake [17]! This signals the necessity of incorporating ideas from chaos theory in the prediction and analysis of natural systems. [1] is a good reference text for the application of chaos in time series prediction.

[1] suggests the use of a feature of chaotic systems called *False Nearest Neighbours* to determine the required length of the delay coordinate vector. Assuming a delay coordinate length of, say, k, the nearest neighbour (**vnn**) of some vector ($\mathbf{v}$) is found and their distance noted. Let the corresponding timestamps be T and TNN. Now the delay coordinate length is increased by 1, and the distance between the two vectors at

times T and TNN is found. If the distance has increased by a large value, it means that **vnn** was not the actual nearest neighbour to **v** in state space, but had arrived in the neighbourhood of **v** by *projection* from a higher dimension. This implies that the current dimension k was not enough to fully unfold the structure of the system. Such a vector (**vnn**) is called a *False Nearest Neighbour*, and its presence indicates that the proper dimension of the system is greater than the dimension being used currently.

This suggests a technique for finding the "right" length of the delay-coordinate vector. Starting with a low dimension, we calculate the total number of false nearest neighbours for all vectors formed by the time series. As dimension increases, the number of false nearest neighbours should fall off till it reaches some constant value (which might not be zero because of noise). The minimum dimension at which this number stops falling significantly is a good estimate for the length of the delay-coordinate vector. I plan to investigate the practical usefulness of this technique in my work.

3.3 Using Fractals

Another possible approach to the problem of time series forecasting can be used if the time series is self-similar, that is, the series looks similar whether we look at a short portion of the series or a long portion. In such a case, the fractal dimension of the time series can be used for prediction. This is an approach that has not been tried before, and if the assumption of self-similarity in time sequences is found to hold, then it could lead to a breakthrough in this area. Several time series, such as those for disk-accesses and network traffic, have been found to obey this assumption. Hence, it might be worthwhile to apply this technique to other forms of time series data.

The fractal dimension of the time series gives an estimate of the intrinsic dimensionality of the data. For example, if we consider points along the diagonal of a cube, the intrinsic dimensionality of these points is one (because they actually lie along a straight line), but the extrinsic dimensionality is three (because the individual points are expressed in 3D-coordinates). I plan to investigate any possible relationship between the fractal dimension and the optimal lag length. By observing how changing the lag length affects the fractal dimension of the corresponding vector space, the optimal lag length may be found. Also, this may provide clues as to which particular components of the lag vector are most influential in increasing the fractal dimension. Then, only the important components could be retained and the others discarded, leading to shorter lag vectors and thus, reducing dimensionality problems. This is analogous to the "feature selection" approach used in tackling several machine learning problems.

3.4 Database Issues

Finally, this system depends on making predictions on the basis of past regularities in the data. Thus, the data must be stored in a form making such similarity searches fast

and efficient. Storage of and similarity searches over delay coordinate vectors leads to several problems. One is the problem of dimensionality curse: the larger the dimension of the delay coordinate vector, the more acute the problem. Another problem is that of fast retrieval from the database. I have already implemented a time series similarity search prototype, which uses a spatial data access method called the R-Tree [10]. The dimensionality curse is tackled by the use of certain tricks, as detailed in [7]. Still, optimizing searches in the R-Tree for this particular problem is an open area. A formal method is needed to set the number of nearest neighbours to be used for prediction purposes.

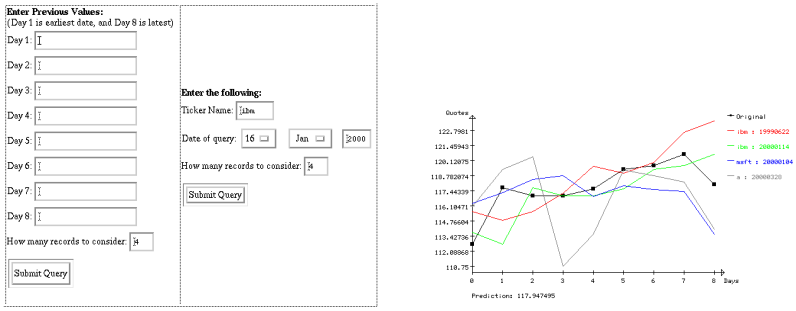
Thus, the main aim of my work is to find formal methods for choosing values for delay-vector length, number of nearest neighbours used for interpolation and selecting the important components of the delay-vector. The success of the work can be judged by comparing the results produced by using these parameters against those when the parameters are chosen on an ad hoc basis. I also plan to compare the results to those produced by an ARMA model. For this purpose, I intend to use the California-Transportation dataset, which I already have. If I can get some larger dataset, I also wish to try these techniques on it.

3.5 My Previous Work

I have worked in this area for one term and have already made some progress. A time series similarity-search prototype is already active, and can be accessed on the Internet at the URL http://snapdragon.cald.cs.cmu.edu/TSP. Currently, predictions can be performed on stock price data and California-city transportation data. Screenshots from the system are shown in Fig 2. The length of the delay coordinate vector is chosen on an ad-hoc basis currently. The dimensionality curse is handled as in [7]. The main objectives for this project is to use the methods outlined in the previous paragraphs to find "right" estimates for the delay vector length and the number of nearest neighbours to be used for prediction. Another aim is to scale up the system to very large datasets (in the order of Gigabytes of data). The proposed research would be the logical extension to the existing system.

4 Work Plan

Work	**Date**
False Nearest Neighbours	02/20
Fractal Dimensions	03/01
Delay-coordinate component selection	03/15
Choosing best number of nearest neighbours	04/01
Scalability for large databases	04/15



(a) (b)

Figure 2: *The current system*: The input screen(a) and the result screen (b)

5 Summary

Time series turn up in a variety of systems, and their analysis and prediction allows us to better understand the systems' internal functioning. In some cases, it might even help in finding the underlying dynamical equations. My approach involves the use of a technique called *Delay Coordinate Embedding*, under which the time series is broken up into chunks, and properties of these chunks are studied. These chunks give an indication of the internal state of the system. I already have a functional prototype incorporating this method, but several key components (such as determination of the length of the chunks) is carried out on an ad hoc basis. The aim of my project is to develop techniques to formalize these components, and also to find methods for efficient storage and retrieval of chunks. This will lead to a method of fast and efficient forecasting of a variety of time sequences.

References

[1] H. D. I. Abarbanel. *Analysis of Observed Chaotic Data.* Springer, 1995.

[2] H. D. I. Abarbanel, R. Brown, and J. J. Sidorowich an L. Sh. Tsimring. The analysis of observed chaotic data in physical systems. *Reviews of Modern Physics*, 65:1331–1392, 1993.

[3] H. D. I. Abarbanel, Z. Gills, C. Liu, and R. Roy. Quantum fluctuations in the operation of a chaotic laser. *Physical Review A*, 51:1175, 1995.

[4] A.S.Weigend and N.A.Gershenfeld, editors. *Time Series Prediction: Forecasting the Future and Understanding the Past.* Addison Wesley, 1993.

[5] C. Bracikowski and R. Roy. Chaos in a multimode solid-state laser system. *Chaos*, 1:49–64, 1991.

[6] C. Chatfield. *The Analysis of Time Series.* London: Chapman and Hall, 1989.

[7] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. *SIGMOD*, 1994.

[8] F. Fessant, S. Bengio, and D. Collobert. On the prediction of solar activity using different neural network models. *Annales Geophysicae*, 1995.

[9] A. M. Fraser and A. Dimitriadis. *Forecasting Probability Densities by Using Hidden Markov Models with Mixed States.* 1993.

[10] A. Guttman. R-trees: A dynamic index structure for spatial searching. *ACM SIGMOD*, pages 47–57, 1984.

[11] K. Lang and G. Hilton. A time-delay neural network architecture for speech recognition. Technical Report CMU-CS-88-152, Carnegie Mellon University, Pittsburgh, PA, 1988.

[12] A. Lapedes and R. Farber. Nonlinear signal processing using neural networks. Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM, 1987.

[13] E. N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20:130–141, 1963.

[14] Michael C. Mozer. *Neural Network Architectures for Temporal Sequence Processing*, pages 243–264. Addison Wesley, 1993.

[15] M. Priestley. *Spectral Analysis and Time Series.* London: Academic Press, 1981.

[16] B. Salzmann. Finite amplitude free convection as an initial value problem. *I. J. Atmos. Sci.*, 29:329–341, 1962.

[17] T. B. Sangoyomi. *Climatic Variability and Dynamics of Great Salt Lake Hydrology.* PhD thesis, Utah State University, 1993.

[18] M. Sauer, J. A. Yorke, and M. Casdagli. Embedology. *J. Stat. Phys.*, 65(3/4):579–616, 1991.

[19] T. Sauer. *Time Series Prediction by Using Delay Coordinate Embedding(Data Set A)*, pages 175–193. Addison Wesley, 1993.

[20] F. Takens. *Detecting Strange Attractors in Turbulence*, volume 898, pages 336–381. Berlin: Springer-Verlag, 1981.

[21] A. Waibel. Modular construction of time-delay neural networks for speech recognition. *Neur. Comp.*, 1(1):39–46, 1989.

[22] E.A. Wan. *Time Series Prediction by Using a Connectionist Network with Internal Delay Line*, pages 195–217. Addison Wesley, 1993.

[23] A. S. Weigend, B. A. Huberman, and D. E. Rumelhart. Predicting the future: A connectionist approach. *International Journal of Neural Systems*, 1:193–209, 1990.

[24] P. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioural Sciences*. PhD thesis, Harvard University, Cambridge, MA, 1974.

[25] P. Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neur. Net.*, 1:339–356, 1988.