In A. Weigend and N. Gershenfeld (eds.), Time Series Prediction: Forecasting the Future and Understanding the Past, 264-281. Addison-Wesley (1994)

Hidden Markov Models with Mixed States[1]

Andrew M. Fraser and Alexis Dimitriadis

Systems Science Ph.D. Program
Portland State University
Portland, Oregon 97207-0751
andy@sysc.pdx.edu

Abstract

We note similarities of the state space reconstruction ("Embedology") practiced in numerical work on chaos, state space methods of stochastic systems theory, and the hidden Markov models (HMMs) used in speech research. We review Baum's EM algorithm in general and the specific forward-backward algorithm that optimizes a class of HMM that has a mixed state space consisting of continuous and discrete parts. We then describe forecasts based on models fit to data set D.

1 Introduction

In the first part of this paper we hope to provide an intuitive explanation of hidden Markov model (HMM) methods that builds on the notion of state space reconstruction. Later, we provide enough details about the approach to enable a careful reader to develop new variants and to write his own programs. We begin with some thoughts on forecasting and state spaces. We were drawn to work on varieties of hidden Markov models for scalar time series from chaotic dynamics, by the similarity of the notion of *reconstructed state space* in the chaos literature to the notion of *hidden state* in the HMM literature. The approach provides forecasts that consist of probability densities instead of single guesses of future values. In section 6, such a forecast of data set D suggests that the approach is quite powerful.

The most direct method of forecasting is to search the past for times when conditions matched the patterns of recent observations and then guess that what happened before will happen again. While forecasts for discrete valued periodic sequences like $(0, 1, 2, 3, 0, 1, 2, 3, 0, 1, \ldots)$ are trivial, forecasting sequences like data set D, $(0.643, 0.558, 0.484, 0.434, 0.422, \ldots)$, is difficult because there is no segment in the recorded history that exactly matches recent observations. In such circumstances one may proceed by *guessing* how *close* conditions at various times in the past are to present conditions and then appropriately averaging near matches to make forecasts; here *closeness* corresponds to distance in *state space*, and conditional probability density functions for location in state space given observations implement the *guesses*.

[1] This is derived from the revised version that we submitted on April 17, 1993. The software that we used to insert figures into that version is not on our current system. In September of 2001, I (Andy Fraser) edited the document to include the figures so that I could make whole ps and pdf files for distribution.

The notion of state space has also been essential in the efforts over the past dozen years in which researchers have claimed that various experimental time series arise from chaotic dynamics. Characteristically scalar time series are converted to vector time series in a procedure called *state space reconstruction.* The simplest procedure is the use of *delay vectors.* Using a notation in which a sequence of scalar observations is denoted by $y_1^T \equiv (y(1), y(2), \ldots, y(T))$, we write delay vectors as $\mathbf{x}(t) \equiv y_{t-m}^{t-1}$, and observe that $\mathbf{x}_{m+1}^T$ can be obtained from y_1^T. Having reconstructed a vector time series, investigators generally argue that the dynamics are deterministic, i.e., $\exists\, \mathcal{F} : \mathbb{R}^m \to \mathbb{R}^m$ such that $\mathbf{x}(t+1) = \mathcal{F}(\mathbf{x}(t))\ \forall t$, and then estimate invariants such as dimensions, entropies, or Lyapunov exponents from experimental measurements.

While the practice of explaining scalar time series in terms of vector dynamics and using observed time series segments to specify locations in state space has a long history[2], the more recent literature in chaos usually cites Packard *et al.*[12] or Takens[18]. The idea is that an "original" state space variable $\mathbf{z}$ evolves via a diffeomorphism[3] F of some low dimensional manifold $\mathbf{Z}$ and gives rise to a scalar observable $y \in \mathbb{R}$

$$\mathbf{z}(t+1) = F(\mathbf{z}(t)) \tag{1a}$$

$$y(t) = g(\mathbf{z}(t)), \tag{1b}$$

and a reconstruction function $\phi : \mathbb{R}^w \to \mathbb{R}^m$ is used to map windows of observations y_{t-w}^{t-1} of size w to reconstructed vectors $\mathbf{x}(t) \in \mathbf{X} = \mathbb{R}^m$. Combining ϕ, F^{-1}, and g one can write $\Phi : \mathbf{Z} \to \mathbf{X}$ with

$$\mathbf{x}(t) = \Phi(\mathbf{z}(t)) = \phi\left(g(F^{-1}(z(t))), g(F^{-2}(z(t))), \ldots, g(F^{-w}(z(t)))\right).$$

Takens showed that if g and ϕ are differentiable and m is large enough, then it is a generic property that Φ is a diffeomorphism, and thus one expects coordinate invariant properties of trajectories and limit sets in $\mathbf{Z}$ to be the same as the properties of their images in $\mathbf{X}$. Although Takens' result is insensitive to the details of ϕ and g, when experimenters implemented the procedure, they found that their estimates of invariants varied with changes in ϕ and g.

This variability of invariants lead to a literature (recently called *Embedology*[16]) concerned with defining and finding *good reconstructions*[8]. The variability is usually explained by observing that the procedures used to estimate invariants converge in the limit of infinite amounts of noise free data, but that experimenters work with short noisy data sets instead. While it may be true that different applications lead to different optimal reconstructions, we suspect[4] that practical embedology will best be developed as an aspect of modeling techniques for optimizing likelihood or a variant such as MDL or AIC.

[2]Most control theory text books have a section titled *Observability* which addresses the question of whether or not sequences of observations uniquely determine locations in state space.

[3]A one to one differentiable function with a one to one differentiable inverse.

[4]This was suggested to us by Henry Abarbanel[1] and is similar to the notion of Casdagli *et al.*[4] that reconstruction and prediction are related.

For noisy or stochastic dynamics and observations, equation (1) is not appropriate. Instead, the ys are functions of a Markov process and are characterized by the conditional densities[5] $P_{\mathbf{z}(t+1)|\mathbf{z}(t)}$ and $P_{y(t)|\mathbf{z}(t)}$ with

$$P\left(\mathbf{z}(t+1)|\mathbf{z}_{-\infty}^{t}, y_{-\infty}^{t}\right) = P\left(\mathbf{z}(t+1)|\mathbf{z}(t)\right) \tag{2a}$$

$$P\left(y(t)|\mathbf{z}_{-\infty}^{t}, y_{-\infty}^{\infty}\right) = P\left(y(t)|\mathbf{z}(t)\right). \tag{2b}$$

Given these conditional density functions and a natural measure or stationary density μ with $\mu(\mathbf{z}') = \int P_{\mathbf{z}(t+1)|\mathbf{z}(t)}\left(\mathbf{z}'|\tilde{\mathbf{z}}\right)\mu(\tilde{\mathbf{z}})d\tilde{\mathbf{z}}$, forecasting formally reduces to iterating a recursion

$$P\left(y(T+1)|\, y_1^T\right) = \int P\left(y(T+1)|\mathbf{z}(T+1)\right) P\left(\mathbf{z}(T+1)|y_1^T\right)\, d\mathbf{z}(T+1) \tag{3a}$$

$$P\left(\mathbf{z}(T+1)|y_1^T\right) = \int P\left(\mathbf{z}(T+1)|\mathbf{z}(T)\right) P\left(\mathbf{z}(T)|y_1^T\right)\, d\mathbf{z}(T) \tag{3b}$$

$$P\left(\mathbf{z}(T)|y_1^T\right) = \frac{P\left(y(T)|\mathbf{z}(T)\right) P\left(\mathbf{z}(T)|y_1^{T-1}\right)}{P\left(y(T)|y_1^{T-1}\right)} \tag{3c}$$

starting with $P(\mathbf{z}(1)) = \mu$. The key idea is that in state space there is an evolving cloud of locations that are consistent with observations up to the time t, i.e., $P\left(\mathbf{z}(t)|y_1^t\right)$. If the conditional densities of equation (2) correspond to adding Gaussian noise to equation (1), they can be written as

$$P(\mathbf{z}(t+1)|\mathbf{z}(t)) = \left[\frac{\left|\Sigma^{-1}\right|}{\sqrt{2\pi}}\right]^n e^{\frac{-[\mathbf{z}(t+1)-F(\mathbf{z}(t))]\Sigma^{-1}[\mathbf{z}(t+1)-F(\mathbf{z}(t))]}{2}} \tag{4a}$$

$$P(y(t)|\mathbf{z}(t)) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{\frac{[y(t)-G(\mathbf{z}(t))]^2}{2\sigma^2}} \tag{4b}$$

where Σ^{-1} is an inverse covariance matrix. Further, if the functions F and G are linear, the recursion of equations (3) constitutes Kalman filtering[6].

The simplicity of the equations (3) obscures the following difficulties:

- The conditional densities that specify a stochastic process may be complex requiring descriptions with infinite numbers of parameters.
- The derived intermediate terms like $P\left(\mathbf{z}(t)|y_1^t\right)$ are not simply values, but functions.
- And most importantly, the conditional densities that specify a stochastic process must be estimated on the basis of observations alone.

[5] Our notation blurs the distinction between probability distributions of discrete variables and probability densities of continuous variables. For the probability at x, we use the notation $P(x)$, for the function, we use P_x, i.e., we use subscripts to specify a function and parentheses to denote the value of a function at a point. We resolve ambiguities such as $P(5.23)$ by using subscripts or $=$, e.g., $P_x(5.23)$ or $P(x{=}5.23)$. We attempt to balance opacity and imprecision in our notation.

[6] Sorenson[17] observes that the basic ideas go back to Gauss.

Rather than attempting to estimate the "true" stochastic process on the basis of observations, in view of these difficulties, we reconsider our pragmatic goals. For forecasting, we are interested only in the ys, the hidden $\mathbf{z}$s are just computational intermediates. Consequently, we consider a *model* of the process to be a sequence of probability density functions $P_{y_1^t} : \mathbb{R}^t \rightarrow \mathbb{R}, \{t = 1, 2, 3, \ldots\}$. In selecting model classes and fitting their parameters, our goal is to obtain a set $\left\{P_{y_1^t} : \forall t \in \mathbb{Z}^+\right\}$ or equivalently $\left\{P_{y(t+1)|y_1^t} : \forall t \in \mathbb{Z}^+\right\}$ that performs well in our application (forecasting) rather than discovering a "true" generating mechanism[7]. We are still exploring several model types and not having carefully accounted for free parameters in our comparisons, we simply use maximum likelihood methods.

2 Model Classes

In this section we begin by introducing basic discrete state discrete output HMMs. Then we turn to mixed state models, of which the hidden filter hidden Markov models HFHMMs[8] described in sections 4 and 5 are a special case.

Equation (2) describes a process in which the observable is a probabilistic function of an underlying Markov process. If the state variables $\mathbf{z}$ and the observations y are drawn from discrete finite sets, the process is what is called a hidden Markov model (HMM). Much HMM development work is motivated by applications in natural language. The original work was done in the Communications Research Division of the Institute for Defense Analysis in Princeton and the methods were reviewed at an open symposium in 1980. In the proceedings Ferguson[7] described HMMs:

> ...a Markov chain with state space $\mathcal{S}$, having S states ..., a finite output alphabet, $\mathcal{K}$, which we may take to be the integers $1, 2, \ldots, K$, and a collection of probability distributions, Explicitly, we need a transition matrix (a_{ij}), $i, j \in \mathcal{S}$, where
>
> $$a_{ij} = \text{Prob}\{\text{next state} = j \text{ given current state} = i\}$$
>
> and we need an output probability matrix $(b_j(k))$, $j \in \mathcal{S}, k \in \mathcal{K}$, where
>
> $$b_j(k) = \text{Prob}\{\text{observation} = k \text{ given current state} = j\}$$
>
> For completeness, we need an initial distribution on states, to get us started. Let $(a(i)), i \in \mathcal{S}$ be this distribution.

HMMs are useful because the probability distributions can be adjusted by the Baum-Welch, or forward-backward, algorithm to maximize the likelihood of a

[7] Our view of forecasting has been influenced by Williams' book on data compression[19] and indirectly by the work of Rissanen[15].

[8] We called these models autoregressive hidden Markov models (ARHMMs) until we found that Poritz[13] had already described them and called them HFHMMs.

given set of training data. We describe the version of the algorithm needed for HFHMMs in section 5.

The following points about discrete HMMs merit emphasis:

1. Although the hidden process is first order Markov, the output process may not be Markov of any order.

2. Even if the dynamics and observations (the functions F and G in equation (4)) are nonlinear, a discrete HMM can approximate the continuous case arbitrarily well by using large numbers of states S and possible output values K.

3. Larger numbers of training data are required as S and K are increased.

As an illustration of point 1 consider the process described by

$$a = \begin{bmatrix} .9 & .1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & .9 & .1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

which produces output strings with runs of about seven 1s interspersed with occasional 2s and 3s. In the output stream, the 2s and 3s alternate no matter how many 1s fall in between. Such behavior can not be captured by a simple Markov process of any order.

A discrete HMM fit to continuous observations of continuous dynamics (e.g. equation (4)) disregards useful properties of the data. The large number of training data in point 3 above can be reduced by preserving a measure of nearness; parameters for situations that do not occur in the training data can be fit on the basis of *interpolations* of nearby situations that do. To build HMM–like models that interpolate in this sense, we introduce the notion of a *mixed state* $\psi(t) = (s(t), \mathbf{x}(t))$ which consists of a discrete part $s \in \{1, 2, \ldots, n_{\text{states}}\}$ and a continuous part $\mathbf{x} \in \mathbb{R}^n$.

As a simplifying assumption, we let the continuous part be a deterministic function of past observations, i.e., $\mathbf{x}(t) = \text{Funct.}(y_1^{t-1})$. The mixed states are meant to summarize histories; thus we assume that $P(y(t)|\psi(t), y_1^{t-1}) = P(y(t)|\psi(t))$ and

$$P(y(t)|y_1^{t-1}) = \sum_{s(t)} P(y(t)|\psi(t))\, P(\psi(t)|y_1^{t-1}). \tag{5}$$

By putting all of the uncertainty about location into the discrete part of a mixed state, the integral in equation (3)a has been simplified to the sum in equation (5). While we doubt natural time series are actually generated by mixed state processes, we use them as models because they achieve such operational simplifications and their observable aspects, i.e., $\left\{P_{y(t+1)|y_1^t} : \forall t \in \mathbb{Z}^+\right\}$, provide high likelihood fits to complex behavior using relatively few free parameters.

In our discussions as we develop models and write programs to implement them, we have found sketches like those in fig. 1 helpful.

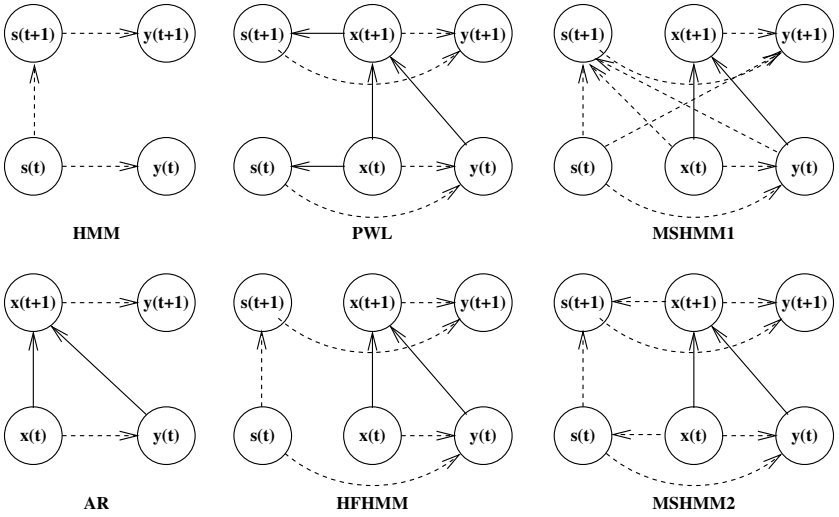

Figure 1: Chains of dependency in various model classes. Solid lines indicate deterministic dependence and dotted lines indicate stochastic influence. In each sketch time advances one step with the subsequent conditions appearing above the prior conditions. Given the depicted influences on a node, earlier values of all variables are irrelevant; thus the **HMM** sketch indicates $P(y(t)|s_1^t, y_1^{t-1}) = P(y(t)|s(t)$ and $P(s(t)|s_1^{t-1}, y_1^{t-1}) = P(s(t)|s(t\text{–}1)$. An **AR** model can be written as $P(y(t)|\mathbf{x}(t)) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(y(t)-\mathbf{a}\cdot\mathbf{x}(t))^2}{2\sigma^2}$ with $\mathbf{x}(t) = F(x(t\text{–}1), y(t\text{–}1))$ defined by $\mathbf{x}_1(t) = y(t\text{–}1)$ and $\mathbf{x}_i(t) = \mathbf{x}_{i-1}(t\text{–}1) : 1 < i \leq m$. In the sketch, the solid lines indicate the function F and the dotted lines indicate $P_{y(t)|\mathbf{x}(t)}$. In a piecewise linear (**PWL**) model, the history vector $\mathbf{x}$ determines the partition element s and thus the linear rule $\hat{y}(t) = \mathbf{a}_s \cdot \mathbf{x}(t)$, then $P_{y(t)|\mathbf{x}(t),s(t)}$ is Gaussian with mean $\hat{y}(t)$ and variance σ_s^2. **HFHMM**s are discussed in sections 4 and 6 and type 1 mixed state hidden Markov models (**MSHMM1**) are discussed in section 6.

3 Incomplete Data: The EM Algorithm

In this section we review the EM algorithm[9] which adjusts model parameters θ to maximize the likelihood of observations $\mathbf{y}$. It operates on models[10] which include unobserved data $\mathbf{s}$, $P_{\mathbf{y},\mathbf{s},\theta}$. In section 4 we describe a specific model for time series, and in section 5 a version of the EM algorithm tailored for that model is presented. The steps in any EM algorithm are:

[9]Our development follows the 1970 paper by Baum *et al.*[2]. In a 1977 paper Dempster Laird and Rubin[5] called the procedure the *estimate maximize* algorithm. We recommend Brown's dissertation[3, page 25] for clarity on the subject and Poritz[13] for a thorough bibliography and historical outline.

[10]For our application, $\mathbf{y}$ is a sequence of observations $(y(1), y(2), \ldots, y(T))$ and $\mathbf{s}$ is a sequence of discrete hidden states $(s(1), \ldots, s(T))$.

1. Guess a starting value of θ.
2. Choose $\hat{\theta}$ to maximize[11] $Q(\theta,\hat{\theta}) \equiv \left\langle \log P_{\mathbf{y},\mathbf{s},\hat{\theta}}(\mathbf{y},\mathbf{s}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta}$
3. Set $\theta = \hat{\theta}$.
4. If not converged, go to 2.

This procedure will work if,

$$Q(\theta,\hat{\theta}) > Q(\theta,\theta) \Rightarrow P_{\hat{\theta}}(\mathbf{y}) > P_{\theta}(\mathbf{y}) \tag{6a}$$

and

$$\max_{\hat{\theta}} Q(\theta,\hat{\theta}) = Q(\theta,\theta) \Rightarrow \max_{\hat{\theta}} P_{\hat{\theta}}(\mathbf{y}) = P_{\theta}(\mathbf{y}). \tag{6b}$$

The truth of the first implication (6a) is shown as follows:

$$P_{\hat{\theta}}(\mathbf{y}) = \frac{P_{\hat{\theta}}(\mathbf{s},\mathbf{y})}{P_{\hat{\theta}}(\mathbf{s}|\mathbf{y})}$$

$$\log P_{\hat{\theta}}(\mathbf{y}) = \log P_{\hat{\theta}}(\mathbf{s},\mathbf{y}) - \log P_{\hat{\theta}}(\mathbf{s}|\mathbf{y})$$

Note $\left\langle \log P_{\hat{\theta}}(\mathbf{y}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta} = \log P_{\hat{\theta}}(\mathbf{y})$ because $\mathbf{s}$ does not appear inside the $\langle\rangle_{\mathbf{s}|\mathbf{y},\theta}$. So

$$\log P_{\hat{\theta}}(\mathbf{y}) = \left\langle \log P_{\hat{\theta}}(\mathbf{s},\mathbf{y}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta} - \left\langle \log P_{\hat{\theta}}(\mathbf{s}|\mathbf{y}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta} \tag{7}$$

The Gibbs inequality[12] for two distributions P_{θ_1} and P_{θ_2} says

$$\sum_x P_{\theta_1}(x) \log \frac{P_{\theta_2}(x)}{P_{\theta_1}(x)} \leq 0 \quad \text{or} \quad \langle \log P_{\theta_2}(x) \rangle_{\theta_1} \leq \langle \log P_{\theta_1}(x) \rangle_{\theta_1}$$

So

$$\left\langle \log P_{\hat{\theta}}(\mathbf{s}|\mathbf{y}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta} \leq \left\langle \log P_{\theta}(\mathbf{s}|\mathbf{y}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta}$$

Now if $\hat{\theta}$ is chosen so that

$$\left\langle \log P_{\hat{\theta}}(\mathbf{s},\mathbf{y}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta} > \left\langle \log P_{\theta}(\mathbf{s},\mathbf{y}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta} \tag{8}$$

equation (7) yields the implication (6a)

$$P_{\hat{\theta}}(\mathbf{y}) > P_{\theta}(\mathbf{y})$$

and the algorithm steps uphill.

The second implication, (6b), does not always hold. Since

$$\left[\frac{\partial}{\partial\hat{\theta}} \left\langle \log P_{\hat{\theta}}(\mathbf{s},\mathbf{y}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta}\right]_{\hat{\theta}=\theta} = \left[\frac{1}{P_{\hat{\theta}}(\mathbf{y})} \frac{\partial}{\partial\hat{\theta}} P_{\hat{\theta}}(\mathbf{y})\right]_{\hat{\theta}=\theta},$$

[11] For discrete s, this notation means $\langle f \rangle_{\mathbf{s}|\mathbf{y},\theta} = \sum_{\mathbf{s}} P_{\mathbf{s}|\mathbf{y},\theta}(\mathbf{s}|\mathbf{y}) f(\mathbf{s})$.

[12] While many information theory texts attribute this inequality to Kullback and Leibler[11], it appeared 50 years earlier in chapter *XI Theorem II* of Gibbs[10].

and

$$\left[\frac{\partial^2}{\partial\hat{\theta}^2}\left\langle\log P_{\hat{\theta}}(\mathbf{s},\mathbf{y})\right\rangle_{\mathbf{s}|\mathbf{y},\theta}\right]_{\hat{\theta}=\theta} = \left[\frac{1}{P_{\hat{\theta}}(\mathbf{y})}\frac{\partial^2}{\partial\hat{\theta}^2}P_{\hat{\theta}}(\mathbf{y}) - \left\langle\left(\frac{\partial}{\partial\hat{\theta}}\log P_{\hat{\theta}}(\mathbf{s},\mathbf{y})\right)^2\right\rangle_{\mathbf{s}|\mathbf{y},\theta}\right]_{\hat{\theta}=\theta}$$

the critical points of $\langle\log P_\theta(\mathbf{s},\mathbf{y})\rangle_{\mathbf{s}|\mathbf{y},\theta}$ and $P_\theta(\mathbf{y})$ are the same, but maxima of the former may be saddle points of the latter. But since their basins of attraction are low dimensional stable manifolds, it is unlikely that the algorithm will get stuck at a saddle point of $P_\theta(\mathbf{y})$.

4 Hidden Filter HMMs (HFHMMs)

HFHMMs are a class of time series models to which the EM algorithm can be applied. They are diagrammed in fig. 1 and consist of a hidden first order Markov process on a set of discrete states $\{s_1, s_2, \ldots, s_{\text{nstates}}\}$ and associated with each discrete state is a linear autoregressive output process. The conditional transition probabilities from any state k are given by $P(s(t{+}1)|s(t){=}k)$, and the output distribution at time t given state $s(t) = k$ and history $\mathbf{x} = (y(t{-}1), y(t-2), \ldots, y(t-m))$ is

$$P_{y(t)|s(t),\mathbf{x}(t)}(y|k,\mathbf{x}) = \frac{1}{\sigma_k\sqrt{2\pi}}\exp -\frac{(y-\bar{y}_k-\mathbf{a}_k\cdot\mathbf{x})^2}{2\sigma_k^2}.$$

A model θ consists of all of the parameters for all of the states; for each state k the parameters are: the transition probabilities $P_{s(t+1)|s(t)}(j|k)$, and the output distribution parameters $\bar{y}_k$, $\mathbf{a}_k$, and σ_k. We impose the following constraints:

- Discrete state transitions are Markov and independent of prior outputs[13]

$$P(s(t)|s_1^{t-1}, y_1^{t-1}) = P(s(t)|s(t{-}1)), \tag{9}$$

  and $s(1)$ and $\mathbf{x}(1)$ are independent

$$P(s(1),\mathbf{x}(1)) = P(s(1))\,P(\mathbf{x}(1)). \tag{10}$$

  Using a notation in which $s_q(t)$ is the t^{th} element in the sequence of states q, i.e., $q \equiv (s_q(1), s_q(2), \ldots, s_q(T))$, we can write

$$P(q) = P(s_q(1))\prod_{t=2}^{T}P(s_q(t)|s_q(t{-}1)).$$

- Given $\mathbf{x}(t)$ and $s(t{-}1)$, earlier values of s and y are irrelevant,

$$P(y(t),s(t)|y_1^{t-1}, s_1^{t-1}) = P(y(t),s(t)|s(t{-}1),\mathbf{x}(t)), \tag{11}$$

  which with eqn. (9) implies

$$P(y(t),s(t)|s(t{-}1),\mathbf{x}(t)) = P(y(t)|s(t),\mathbf{x}(t))P(s(t)|s(t{-}1)). \tag{12}$$

[13] This may be appropriate if "the noise scale is at least as large as the discrete states", but it is a bad approximation for noise free deterministic dynamics.

Given y_1^T, a sequence of observations for training, the EM algorithm adjusts the model parameters θ to maximize the likelihood which can be evaluated as

$$P_\theta\left(y_1^T\right) = \sum_q P_\theta\left(y_1^T, q\right),$$

where the variable q runs over all possible state sequences. Step 2 of the algorithm prescribes selecting new parameters $\hat{\theta}$ to maximize the expected log likelihood $\left\langle \log P_{y_1^T,q,\hat{\theta}}(y_1^T, q) \right\rangle_{q|y_1^T,\theta}$, where the expectation is with respect to the conditional distribution $P_\theta(q|y_1^T)$ based on the old parameters θ. We assume $\mathbf{x}(1)$ is available[14], and use the assumptions to write:

$$\begin{aligned}
P_{\hat{\theta}}(y_1^T, q) &= P_{\hat{\theta}}(y(1), s_q(1)|\mathbf{x}(1)) \prod_{t=2}^{T} P_{\hat{\theta}}(y(t), s_q(t)|s_q(t\text{–}1), \mathbf{x}(t\text{–}1)) \\
&= P_{\hat{\theta}}(s_q(1)) \prod_{t=2}^{T} P_{\hat{\theta}}(s_q(t)|s_q(t\text{–}1)) \prod_{t=1}^{T} P_{\hat{\theta}}(y(t)|s_q(t), \mathbf{x}(t)) \\
\log P_{\hat{\theta}}(y_1^T, q) &= \log P_{\hat{\theta}}(s_q(1)) + \sum_{t=1}^{T-1} \log P_{\hat{\theta}}(s_q(t\text{+}1)|s_q(t)) \\
&\quad + \sum_{t=1}^{T} \log P_{\hat{\theta}}(y(t)|s_q(t), \mathbf{x}(t)) \\
&= \log P_{\hat{\theta}}(s_q(1)) + \sum_{t=1}^{T-1} \log P_{\hat{\theta}}(s_q(t\text{+}1)|s_q(t)) \\
&\quad - \sum_{t=1}^{T} \left\{ \log \sigma_{s_q(t)} + \frac{1}{2}\log 2\pi + \frac{\left(y(t) - \bar{y}_{s_q(t)} - \mathbf{a}_{s_q(t)} \cdot \mathbf{x}(t)\right)^2}{2\sigma_{s_q(t)}^2} \right\}
\end{aligned}$$

To proceed with the optimization formally, we need $P_\theta(q|y_1^T)$. If we use the notation $w(q) \equiv P_\theta(y_1^T, q)$, then $P_\theta(q|y_1^T) = w(q)/W$, where $W \equiv \sum_{q'} w(q')$. The number of terms in $\sum_{q'} w(q')$ depends exponentially on T, precluding a direct evaluation for Ts large enough to be interesting, but the sum *can* be evaluated by the *forward-backward algorithm* which is *linear* in T. We will describe the algorithm in section 5, but first we write out expressions for the required optimization.

$$\begin{aligned}
W \left\langle \log P_{\hat{\theta}}(y_1^T, q) \right\rangle_{q|y_1^T,\theta} &= \sum_q w(q) \log P_{s(1),\hat{\theta}}(s_q(1)) \\
&\quad + \sum_{q,t=1}^{T-1} w(q) \log P_{\hat{\theta}}(s_q(t\text{+}1)|s_q(t)) - \frac{W}{2}\log 2\pi \\
&\quad - \sum_{q,t=1}^{T} w(q) \left\{ \log \sigma_{s_q(t)} + \frac{\left(y(t) - \bar{y}_{s_q(t)} - \mathbf{a}_{s_q(t)} \cdot \mathbf{x}(t)\right)^2}{2\sigma_{s_q(t)}^2} \right\}
\end{aligned}$$

[14] We also drop $P_{\mathbf{x}(1)}(\mathbf{x}(1))$ in all calculations, i.e., set $P_{\mathbf{x}(1)}(\mathbf{x}(1)) = 1$.

and the maximization can be done separately by maximizing the term

$$F(\hat{\theta}) \equiv \sum_q w(q) \log P_{\hat{\theta}}(s(1)=s_q(1)) + \sum_{q,t} w(q) \log P_{\hat{\theta}}(s_q(t+1)|s_q(t)) \tag{13}$$

and minimizing the term

$$G(\hat{\theta}) \equiv \sum_{q,t} w(q) \left\{ \frac{\left(y(t) - \bar{y}_{s_q(t)} - \mathbf{a}_{s_q(t)} \cdot \mathbf{x}(t)\right)^2}{2\sigma^2_{s_q(t)}} + \log \sigma_{s_q(t)} \right\} \tag{14a}$$

$$\equiv \sum_{q,t} w(q) g(\hat{\theta}_{s_q(t)}, y(t), \mathbf{x}(t)). \tag{14b}$$

In equation (14) $\hat{\theta}_s$ refers to the parameters in the model that are associated with the state s, i.e., $\bar{y}_s$, $\mathbf{a}_s$, and σ_s. We convert the sum over q and t to a sum over s and t:

$$G(\hat{\theta}) = \sum_{s,t} \left\{ g(\hat{\theta}_s, y(t), \mathbf{x}(t)) \sum_q w(q) \delta_{s,s_q(t)} \right\} \tag{15a}$$

$$\equiv \sum_{s,t} g(\hat{\theta}_s, y(t), \mathbf{x}(t)) w(s,t). \tag{15b}$$

The function $w(s,t)$ introduced in equation (15) is the total probability, considering all possible paths q, that the system is in state s at time t and that the sequence of outputs y_1^T is produced by the model, i.e.,

$$w(s,t) \equiv P_\theta(s(t), y_1^T).$$

In section 5 we describe how to calculate $w(s,t)$ using the forward-backward algorithm. Given $w(s,t)$, finding new values for $\bar{y}_s$, $\mathbf{a}_s$, and σ_s is fairly standard linear fitting; solve for $\mathbf{a}_s$ and $\bar{y}_s$ by using the SVD method[15] to minimize

$$\chi^2 = \sum_t \left\{ y(t)\sqrt{w(s,t)} - (\bar{y}_s + \mathbf{x}(t) \cdot \mathbf{a}_s)\sqrt{w(s,t)} \right\}^2,$$

and, defining $W(s) = \sum_t w(s,t)$, set

$$\sigma_s = \sqrt{\frac{1}{W(s)} \sum_t w(s,t) \left(y(t) - \hat{y}(t)\right)^2}.$$

Introducing the notation

$$w(i,j,t) \equiv P_{s(t+1),s(t),y_1^T,\theta}(i,j,y_1^T),$$

and denoting the new discrete transition probabilities $f_{ij} \equiv P_{s(t+1)|s(t),\hat{\theta}}(i|j)$, we observe that optimizing eqn. (13) requires maximizing

$$F_{ij} \equiv \sum_{i,t} w(i,j,t) \log(f_{ij}) \quad \text{subject to:} \quad \sum_i f_{ij} = 1.$$

[15] See equation 14.3.16 on page 535 of Press *et al.*[14]

The Lagrange multiplier method yields

$$f_{ij} \propto \sum_t w(i,j,t).$$

Selecting the new $P_{s(1),\hat{\theta}}(s)$ is a similar problem, and the solution is given in equation (16) of the next section.

5 The Forward-Backward Algorithm

The forward-backward algorithm is an EM algorithm specifically for time series. The first steps of the algorithm are two passes through the time series: One "forwards" from $t = 1$ to $t = T$ to calculate αs, and the other "backwards" from $t = T$ to $t = 1$ to calculate βs. The factors $w(s,t)$ and $w(i,j,t)$ used in the previous section, can be evaluated in terms of these αs and βs which are defined as follows:

$\alpha(s,t)$ The probability, based on the model, of the observations up to time t and that the system is in state s at time t:

$$\alpha(s,t) \equiv P(y_1^t, s(t))$$

$\beta(s,t)$ The probability, based on the model, of the observations after time t given that the system is in state s at time t and given the previous observations y_1^t:

$$\beta(s,t) \equiv P(y_{(t+1)}^T | s(t), y_1^t)$$

These definitions and equations (11) and (12) yield

$$w(s,t) = \alpha(s,t)\beta(s,t),$$

$$w(i,j,t) = \alpha(j,t)\, P(y(t{+}1)|s(t{+}1){=}i, \mathbf{x}(t{+}1))\, P(s(t{+}1){=}i|s(t){=}j)\, \beta(i,t{+}1),$$

and the recursion formulas

$$\alpha(s,t) = \sum_j \alpha(s_j, t{-}1)\, P(s(t){=}s|s(t{-}1){=}s_j)\, P(y(t)|s(t){=}s, \mathbf{x}(t)),$$

$$\beta(s,t) = \sum_j \beta(s_j, t{+}1)\, P(s(t{+}1){=}s_j|s(t){=}s)\, P(y(t{+}1)|s(t{+}1){=}s_j, \mathbf{x}(t{+}1)).$$

Note:

1. For the new model, the initial state probabilities are

$$P_{s(1),\hat{\theta}}(s) \propto P_{s(1),y_1^T,\theta}(s, y_1^T) = \alpha(s,1)\beta(s,1) \tag{16}$$

subject to normalization.

2. The overall likelihood of the observations can be evaluated after a forward pass via:
$$P_\theta(y_1^T) = \sum_s \alpha(s,T)$$

3. The forward recursion is initialized by:
$$\alpha(s,1) = P(s(1){=}s)\, P(y(1)|s(1){=}s, \mathbf{x}(1))$$

4. The backward recursion is initialized by:
$$\beta(s,T) = 1$$

5.1 Programming Tricks

If
$$\gamma(t) \equiv \sum_s \alpha(s,t) = P(y_1^t),$$
and the process has an entropy rate h, then
$$\gamma(t) \approx e^{-ht},$$
and something must be done to prevent under(over)flow for even moderate values of t. The trick is: at each step in the forward recursion record only $a(t)$ and $c(t)$, and at each step in the backward recursion record only $b(t)$, where
$$c(t) \equiv \frac{\gamma(t)}{\gamma(t{-}1)} \quad \text{or} \quad \gamma(t) = \prod_{\tau=1}^{t} c(\tau),$$
$$a(j,t) = \frac{\alpha(j,t)}{\gamma(t)} = P(s(t){=}j|y_1^t),$$
and
$$b(s,t) = \frac{\beta(s,t)\gamma(t)}{\gamma(T)}.$$
Thus for any t
$$a(s,t)b(s,t) = \frac{\alpha(s,t)\beta(s,t)}{\gamma(T)} \propto w(s,t)$$
and
$$\frac{a(j,t)b(i,t{+}1)}{c(t{+}1)} P(y(t{+}1)|s(t{+}1){=}i, \mathbf{x}(t{+}1)) P(s(t{+}1){=}i|s(t){=}j) \propto w(i,j,t).$$

In each iteration of the forward-backward algorithm, there is a loop over discrete states in which the parameters for each state θ_s are reestimated. The new estimates are in part based on the weights $\{w(s,t) : t = 1,\ldots,T\}$. Because the Gaussians used in the models have tails that go on forever, $w(s,t) > 0, \forall t$. The times t with weights below a small threshold $w(s,t) < \epsilon$ have little effect on the new parameters θ_s, and discarding these times speeds the computations.

6 Forecasts of Data Set D

We have written a family of programs that construct and optimize HFHMMs. To seed the forward-backward algorithm we construct a HFHMM based on a partition of the space of autoregressive history vectors that we generate by Lloyd iteration[16]. Specifying a quantization vector v_s and metric $d_s()$ for each cell s defines the partition. An autoregressive history vector $\mathbf{x}$ is in the cell s which minimizes $d_s(v_s, \mathbf{x})$. In each cell, we set the metric proportional to the inverse covariance of the data in the cell. To construct the seed HFHMM, we associate a hidden state with each cell of the partition, initialize the parameters of $P_{y(t)|s(t),\mathbf{x}(t)}$ with a linear fit over training data that fall in the cell, and use relative frequencies of transitions between cells to estimate discrete state transition probabilities.

We used this procedure to fit the model that generated fig. 2*a*, a forecast of data set D. The model is the result of 70 passes of the forward-backward algorithm, each of which required about 45 minutes on a SPARCstation 2. We estimated the probability density that constitutes the forecast using a Monte Carlo method.

As the number of time steps is increased in very long forecasts, probability leaks away to exponentially larger values of y. Although this effect is subtle in fig. 2*a*, for simple chaotic systems it is dramatic[6]. Considering the Lyapunov exponents helps explain this defect. A discrete state sequence q, starting at the end of the observed data T and continuing for a forecast of τ steps $q \equiv (s_q(T{+}1), s_q(T{+}2), \ldots, s_q(T{+}\tau))$, specifies a sequence of linear maps, i.e., derivative information. For typical long sequences q, the magnitudes of the eigenvalues of the products of these maps will grow at exponential rates given by the Lyapunov exponents. For chaotic systems, at least one Lyapunov exponent is positive, and the composed maps are linearly, and hence globally, unstable. The problem is that the model is linear in the sense that

$$y^{T+\tau}_{T,q,\lambda\mathbf{x}(T)} = \lambda y^{T+\tau}_{T,q,\mathbf{x}(T)}$$

and

$$P_\theta(\lambda y^{T+\tau}_{T,q,\lambda\mathbf{x}(T)} | q, \lambda\mathbf{x}(T)) = P_\theta(y^{T+\tau}_{T,q} | q, \mathbf{x}(T))$$

where $y^{T+\tau}_{T,q,\mathbf{x}(T)}$ denotes the y sequence that maximizes $P_\theta(y^{T+\tau}_{T+1} | q, \mathbf{x}(T))$. Thus, in a HFHMM there are no nonlinearities to saturate diverging y values. The models described below address this weakness.

6.1 Output dependent state transitions

By allowing $\mathbf{x}$ values to influence the transition probabilities between discrete states, one can introduce the nonlinear saturation that HFHMMs miss. In such models, the sequence $(s(t), \mathbf{x}(t))$ still constitutes a Markov process, but the sequence of discrete states $s(t)$ alone does not.

The assumptions we now make are:

[16] For details on vector quantization, see Gersho and Gray's recent text[9]

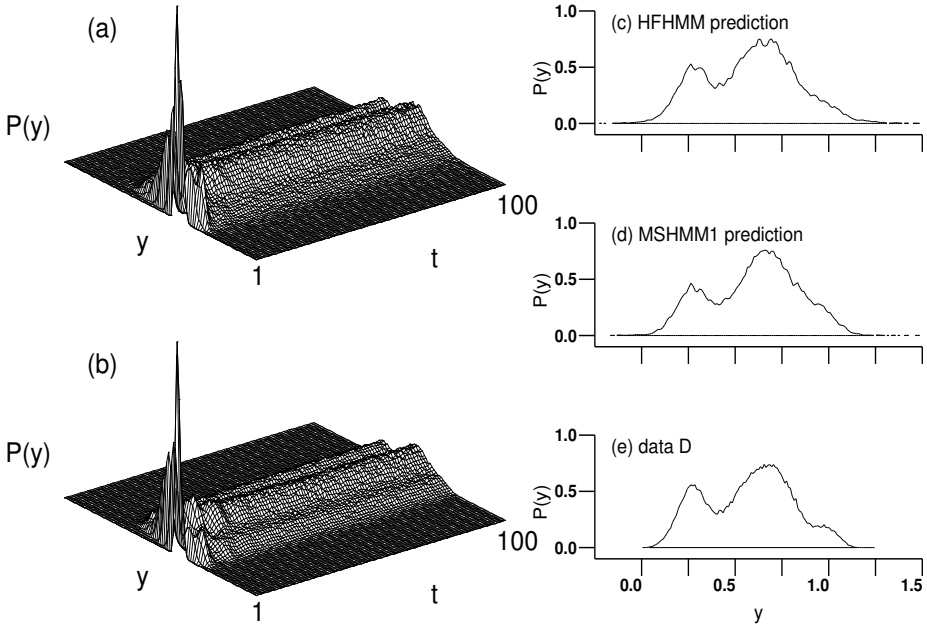

Figure 2: Plot a is a forecast of data set D generated by a HFHMM. The model has 25 discrete states and the autoregressive filters are 8^{th} order. A MSHMM1 with 20 discrete states and 8^{th} order autoregressive filters generated the forecast in plot b. Plots c and d illustrate the long-term behavior of the models. Each is a prediction of the distribution of y one hundred steps in the future. The predictions relax to distributions that are close to the overall distribution of data set D, which is plotted in e.

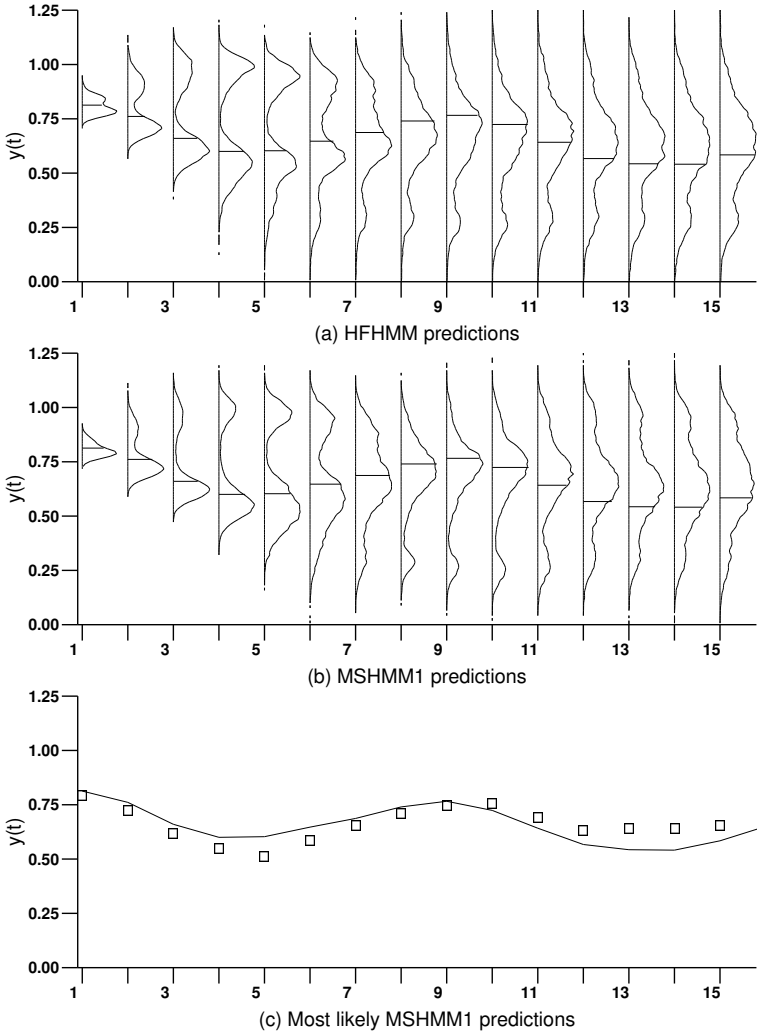

Figure 3: These plots illustrate the short-range behavior of the models. The first few time steps of the forecasts of figure 2 appear in plots *a* and *b*. For each step, the probability density is sketched and a horizontal bar indicates the *true* continuation data. The squares in *c* denote the peaks of the predictions in *b*, and the line connects the points of the true continuation data.

(a) $P(s(t{+}1) \mid s_1^t, y_1^t) = P(s(t{+}1)|s(t), y(t), \mathbf{x}(t))$

(b) $P(y(t) \mid s_1^t, y_1^{t-1}) = P(y(t)|s(t), \mathbf{x}(t))$

(c) $P(y(t), \mathbf{x}(t) \mid s(t), s(t{+}1))$ has a multivariate normal distribution.

(d) The entire process is stationary.

This type of model is referred to in the figures as MSHMM1. Simpler model types assume that $P(y(t) \mid s(t), \mathbf{x}(t))$ and $P(\mathbf{x}(t{+}1)|s(t{+}1), s(t))$ are normal and that $P(s(t{+}1) \mid s_1^t, y_1^t) = P(s(t{+}1)|s(t), \mathbf{x}(t{+}1))$.

Under the above assumptions $\psi(t) = (s(t), \mathbf{x}(t))$ is a Markov process. It is possible to compute all quantities of interest by maintaining, for each transition $s(t) \rightarrow s(t{+}1)$, the parameters of a multivariate normal distribution $P(\mathbf{x}(t) \mid s(t), s(t{+}1))$, a normally-distributed linear prediction $P(y(t)|s(t), s(t{+}1), \mathbf{x}(t))$, and the constant $P(s(t{+}1) \mid s(t))$.

The EM algorithm is applicable to this class of models as well. The transition probabilities in the mixed state space are

$$\begin{aligned} P(\psi(t{+}1) \mid \psi(t)) &= P(\mathbf{x}(t{+}1) \mid s(t{+}1), s(t), \mathbf{x}(t)) P(s(t{+}1) \mid s(t), \mathbf{x}(t)) \\ &= P(y(t) \mid s(t{+}1), s(t), \mathbf{x}(t)) \frac{P(\mathbf{x}(t) \mid s(t), s(t{+}1)) P(s(t{+}1) \mid s(t))}{P(\mathbf{x}(t) \mid s(t))} \end{aligned}$$

The EM algorithm is equivalent to maximizing $\langle \sum_t \log P(\psi(t{+}1) \mid \psi(t)) \rangle_{q|y_1^T}$. While we do not yet have working code that maximizes this, we have obtained good results using a naive algorithm which maximizes $\langle \sum_t \log P(y(t) \mid s(t{+}1), s(t), \mathbf{x}(t)) \rangle_{q|y_1^T}$, $\langle \sum_t \log P(\mathbf{x}(t) \mid s(t{+}1), s(t)) \rangle_{q|y_1^T}$, and $\langle \sum_t \log P(s(t{+}1) \mid s(t)) \rangle_{q|y_1^T}$, but ignores the denominator term, $-\langle \sum_t \log P(\mathbf{x}(t) \mid s(t)) \rangle_{q|y_1^T}$. Application of this naive algorithm yields essentially monotonic improvement in performance, with each model very close to the true optimal performance for that step in the process.

Predictions made by such a model trained on data set D appear in figs. 2*b*, 3*b*, and 3*c*. Even for very long forecasts there is no leakage of probability to ever larger ys.

7 Acknowledgments

This research was supported in part by NSF grant MIP-9113460 and by a contract from Radix Inc. Conversations with many other researchers including Henry Abarbanel, Ronald Hughes, Cory Myers, and Todd Leen have influenced this work.

References

[1] H.D.I. Abarbanel and J. Kadtke. Information theoretic methods for determining the minimum embedding dimension for strange attractors. *INLS/UCSD preprint*, May 1990. Unpublished.

[2] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occuring in the statistica analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[3] P.F. Brown. *The Acoustic-Modeling Problem in Automatic Speech Recognition.* PhD thesis, Carnegie Mellon University, Pittsburgh, 1987.

[4] M. Casdagli, S. Eubank, J.D. Farmer, and J. Gibson. State space reconstruction in the presence of noise. *Physica D*, 51D:52–98, 1991.

[5] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J.R. Stat. Soc. B*, 39:1–78, 1977.

[6] A. Dimitriadis and A.M. Fraser. Modeling double scroll time series. *Journal of Circuits, Systems and Computers*, 1993. Submitted December, 1992.

[7] J.D. Ferguson. Hidden Markov analysis: An introduction. In *Proc. of the Symposium on the Applications of Hidden Markov Models to Text and Speech*, pages 8–15, Princeton, 1980. IDA-CRD.

[8] A. M. Fraser. Phase space reconstructions from time series. *Physica D*, 34D:391–404, 1989.

[9] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer, Norwell MA, 1992.

[10] J.W. Gibbs. *Elementary Principles in Statistical Mechanics Developed with Especial Reference to the Rational Foundation of Thermodynamics.* Yale University Press, 1902. Republished by Dover in 1960.

[11] S. Kullback and R.A. Leibler. On information and sufficiency. *Ann. Math. Stat.*, 22:79–86, 1951.

[12] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R.S. Shaw. Geometry from a time series. *Phys. Rev. Lett.*, 45(9):712–716, Sept. 1 1980.

[13] A.B. Poritz. Hidden Markov models: A guided tour. In *Proc. IEEE Intl. Conf. on Acoust. Speech and Signal Proc.*, 1988.

[14] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Veterling. *Numerical Recipes in C.* Cambridge University Press, Cambridge, 1988.

[15] J. Rissanen. *Stochastic Complexity and Statistical Inquiry.* World Scientific, 1989.

[16] T. Sauer, J.A. Yorke, and M. Casdagli. Embedology. *J. Stat. Phys.*, 65:579–616, 1991.

[17] H.W. Sorenson. Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, pages 63–68, July 1970.

[18] F. Takens. Detecting strange attractors in turbulence. In D. A. Rand and L.S. Young, editors, *Dynamical Systems and Turbulence, Warwick, 1980, Lecture notes in mathematics vol. 898*, pages 366–381, Berlin, 1981. Springer.

[19] R.N. Williams. *Adaptive Data Compression.* Kluwer, Norwell MA, 1991.