Modeling Time Series With Auto-Regressive Markov Models

Alexis Dimitriadis

August 6, 1992

This paper is my M.A. thesis at the Department of Mathematics of Portland State University. It reviews the theory of Hidden Filter Hidden Markov Models and presents an extension, *Mixed State Hidden Markov Models,* developed jointly by Andrew Fraser and myself under his supervision. This manuscript version has only trivial differences from the original.

Contents

1 Introduction and generalities

This section introduces the problem of modeling time series, a number of successively more sophisticated viewpoints, and some terminology and notation.

A *time series* is a sequence of real numbers $\{y(1), y(2), ...\}$, which are understood to be the output of a process at times $t = 1, 2, ...$ The objective of time series modeling is to estimate, from a set of observations $y_1^T = \{y(1), y(2), ..., y(T)\}$, the likely values of the continuation of the process past the time T. This is accomplished by deriving some kind of approximation to the function y.

In general it is assumed that the process being modeled is stationary, that is, its characteristics do not change with time. Thus rather than consider a time series as a function from $\mathbb{N}$ or $\mathbb{Z}$ to $\mathbb{R}$, it is more appropriate to take an *auto-regressive* (AR) approach, in which the output $y(t)$ is viewed as a function of the history y_1^{t-1}. If it is assumed that $y(t)$ depends only on the vector $\mathbf{x}(t) \equiv y_{t-D}^{t-1}$ of the D most recent outputs, the goal of modeling is to find an optimal predictive function $f : \mathbb{R}^D \to \mathbb{R}$ among some chosen class of candidate functions. Thus a modeling strategy must include the selection of a function class rich enough to approximate the process well, but simple enough that a member function can be fully specified by a finite number of parameters, *and* that an algorithm for choosing an optimal function can be found.

The classic method of linear regression can be viewed in this light: the object is to fit a function $f : \mathbb{R}^1 \to \mathbb{R}$ given by $f(y) = ay + b$, such that the prediction for $y(t)$, $\hat{y}(t) = f(y(t-1))$, is optimal in the sense of minimizing the expected square error; linear regression is a formula that gives the optimal values of a and b, given a sequence of observations and the assumption that the error is normally distributed.

Similarly, multivariate linear regression fits a linear function $f : \mathbb{R}^D \to \mathbb{R}$ that would predict $\hat{y}(t) = f(y_{t-D}^{t-1})$. There are also formulas that optimize higher-order polynomial functions, etc.

A more elaborate method is *piecewise-linear* (PL) prediction. Rather than fit the same linear function over the entire domain, one divides the space of D-dimensional AR vectors into regions and fits a linear function to each region. Since any differentiable function can in principle be approximated arbitrarily closely by a piecewise-linear function, this approach is very powerful, but introduces the problem of choosing the best partitioning into regions.

Although any partitioning method could be used, better results are obtained if each region consists of history vectors that are "close together", so that the associated output can be predicted accurately with a single linear function. The approach we use involves *vector quantization*, choosing a set of

points (*centroids*) such that the total squared error $\sum_t d^2(n(\mathbf{x}(t)), \mathbf{x}(t))$ is as small as possible. Here $d(a, b)$ is the distance of the points a, b under a suitable metric, and $n(\mathbf{x})$ is the centroid whose distance from the vector $\mathbf{x}$ is minimal. The partition is then given by the relation $\mathbf{x}_1 \sim \mathbf{x}_2$ *iff* $n(\mathbf{x}_1) = n(\mathbf{x}_2)$. In this case there is no algorithm that can directly compute an optimal choice of centroids. *Lloyd iteration* replaces a set of centroids with a set for which the above sum is smaller. The process can be repeated until the choices stabilize or until the improvement in total error becomes negligible. Although the algorithm is widely used, it has not been proved that each iteration must result in lower total error.

Our work takes the *state space* approach: the time series is viewed as the outputs of a process that passes through unobserved (or, in our formulation, incompletely observed) states. A well-established method is to consider the time series as the output of a process described by a Hidden Markov Model (HMM). At any time t, the process is in state $s(t) \in S$, where S is typically a finite set. The sequence of states $s(1), s(2), \ldots$ constitutes a first-order Markov process. [1] At every transition from a state s_i to a state s_j, the process produces an output $y(t)$ whose value has a normal probability distribution. The mean and variance $\mu_{s_i,s_j}, \sigma_{s_i,s_j}$ of $y(t)$ given the transition $s_i \to s_j$ are parameters of the model. Fitting such a model involves choosing optimal output parameters and state transition probabilities. There are iterative algorithms that fit such models.

A combination of HMMs and piecewise linear AR models, in which the expected output is a function of the history vector as well as the state, was first described by Poritz[1] and dubbed Hidden Filter Hidden Markov Models. In Fraser and Dimitriadis[2], we studied this class of models and discussed, but did not develop in detail, a generalization which drops the assumption that the hidden states form a Markov chain but assumes that the *mixed states* $\psi(t) = (s(t), \mathbf{x}(t))$ form a Markov chain. This allows us to model a time series as the outputs of a Markov process with uncountably many possible states.

In the following sections I develop the theory behind the Hidden Filter and Mixed State types of HMM and the algorithms involved in fitting them from a somewhat different viewpoint than that of [2]. The development of the Mixed State HMM has not been published anywhere before.

I consider this paper an update and continuation, rather than an exposition, of [2]. In general I have aimed for completeness and, where possible, for consistency of presentation with [2]. This is my excuse for including from it substantial passages, sometimes verbatim. In particular section 5.4 and parts of sections 2 and 3 are excerpted from [2]. Section 6

[1] A sequence $s(1), s(2), \ldots$ constitutes a first-order Markov process if the transition probabilities at all times depend only on the most recent state, i.e., for all $t \in \mathbf{N}, s_i \in S$, $P(s(t) = s_i \mid s_1^{t-1}) = P(s(t) = s_i \mid s(t–1))$.

parallels the corresponding section in [2], since the forward backward algorithm can be adapted fairly directly to Mixed State models.

2 The state space approach to modeling

This section develops the notion of time series as the noisy output of a hidden dynamical system and recasts modeling as the problem of fitting certain probability distributions.

At the time t an n-step forecast is a probability density $P_{y(t+n)|y_1^t}$. in particular, the conditional density of a one-step forecast can be extracted from a joint density

$$P_{y(t+1)|y_1^t} \equiv \frac{P_{y_1^{t+1}}}{P_{y_1^t}}$$

(by definition) where $P_{y_1^t}$ is obtained from $P_{y_1^{t+1}}$ by integrating out $y(t+1)$, and further forecasts can be computed iteratively from $P_{y(t+1)|y_1^t}$; equivalently, the joint density can be written as a product of conditionals

$$P_{y_1^T} = P_{y(1)} \prod_{t=1}^{T-1} P_{y(t+1)|y_1^t}.$$

Thus forecasting, and time series modeling in general, can be cast as the problem of fitting high dimensional densities $(P_{y_1^t} : \mathbb{R}^t \to \mathbb{R}, \{t = 1, 2, 3, \ldots\})$ to a process. A standard simplifying assumption is that the process in question is n^{th} order Markov, meaning $P_{y(t)|y_1^{t-1}} = P_{y(t)|y_{t-n}^{t-1}}$. A state space approach enables one to build a model in which the observations alone are not a Markov process of any order, but the underlying *hidden* state space process is first order Markov.

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. The state space approach refines this direct method, using several *near* matches and refined notions of nearness to make forecasts. Rather than simply calculating the difference or squared difference between current measurements and historical measurements to determine the closeness of the conditions or contexts, it supposes that contexts are specified by points in an imperfectly observable state space. Thus to make a forecast we first determine which locations in state space are consistent with recent observations and then calculate the probabilities of subsequent observations conditioned on these states.

In the state space view, observations are functions of internal states. The classic analysis (which, according to Sorenson[3], originates with the ideas of Gauss) views time series as the noisy output of an unobserved, noisy system. In the equations

$$\begin{aligned} \psi(t{+}1) &= F(\psi(t)) + \eta_\psi(t) \\ y(t) &= G(\psi(t)) + \eta_y(t), \end{aligned} \tag{1}$$

ψ is a point in an unobserved vector state space, F describes the state space dynamics, y is an observation derived from the state by the function G, and η_ψ and η_y describe the dynamical noise and observation noise respectively. Given probability densities for η_ψ and η_y, specifying equation (1) specifies the conditional density functions $P_{\psi(t+1)|\psi(t)}$ and $P_{y(t)|\psi(t)}$.

Equation (1) implies that

$$P(\psi(t+1) \mid y_1^{t-1}, \psi_1^t) = P(\psi(t+1) \mid \psi(t)) \tag{2}$$

$$P(y(t) \mid y_1^{t-1}, \psi_1^t) = P(y(t) \mid \psi(t)) \tag{3}$$

Thus the sequence $\{\psi(t)\}$ is first order Markov,[2] and the hidden state $\psi(t)$ encapsulates all predictive information available in the history y_1^{t-1}, ψ_1^t. From now on we will consider processes that obey equations (2) and (3), including all processes described by (1), but dropping the requirement that the noise be independent of the state $\psi(t)$. For such processes, estimates of state space distributions $P_{\psi(t+1)|y_1^t}$ and forecasts

$$P(y(t{+}1) \mid y_1^t) = \int d\psi(t{+}1)\, P(y(t{+}1) \mid \psi(t{+}1))\, P(\psi(t{+}1) \mid y_1^t)$$

can be updated on the basis of each new observation in a sequence recursively; before $y(t)$ is considered, the procedure has calculated a density $P_{\psi(t)|y_1^{t-1}}$ in the hidden state space and the new observation is used to update the density estimate to

$$P_{\psi(t+1)|y_1^t} = \int d\psi(t) P_{\psi(t+1)|\psi(t),y(t)} P_{\psi(t)|y_1^{t-1}} = \int d\psi(t) \frac{P_{\psi(t+1)|\psi(t)}}{P_{y(t)|\psi(t)}} P_{\psi(t)|y_1^{t-1}}$$

If $\psi \in \mathbb{R}^m$, then at each step calculating the conditional distribution $P_{\psi(t+1)|y_1^t}$ involves the old $P_{\psi(t)|y_1^{t-1}}$ and $P_{\psi(t+1)|\psi(t),y(t)} : \mathbb{R}^{2m+1} \to \mathbb{R}$ but not $P_{\psi(t+1)|y_1^t} : \mathbb{R}^{m+t} \to \mathbb{R}$ directly.

[2] We will frequently make use of the property

$$P(A \mid B, C) = P(A) \Rightarrow P(A \mid B) = P(A) \tag{4}$$

I.e., whenever a set of conditions $\{B, C\}$ is independent of the probability of an event A, any subset of the conditions may itself be added or dropped without affecting the probability. Thus equation (2) implies that $P(\psi(t+1) \mid \psi_1^t) = P(\psi(t+1) \mid \psi(t))$.

Thus, using state space methods, one can build and evaluate high dimensional non-Markovian joint densities for an observable y on the basis of lower dimensional functions. It should be noted that $P_{y_1^T}$ is completely defined by $P_{\psi(t+1)|\psi(t))}$, $P_{y(t)|\psi(t)}$ and $P_{\psi(1)}$. Thus a process can be modeled by specifying just these functions.

3 Maximum likelihood and the EM algorithm

This section introduces the EM algorithm and shows the first steps of how an estimate of the characteristics of a process could be deduced from its output.[3]

Given a time series y_1^T, any stochastic model θ for which the associated probability $P_\theta(y_1^T)$ is non-zero could in principle have generated the output y_1^T. But for many models y_1^T will occur with very low probability, and their typical output will be nothing like y_1^T. The goal of modeling is to select a model θ for which the output y_1^T is as "typical" as possible, i.e., for which $P_\theta(y_1^T)$, the likelihood of the data given the model, is as large as possible. This is known as *maximum likelihood estimation*.

The EM (*estimate maximize*) algorithm iteratively adjusts model parameters θ to maximize the likelihood of observations $\mathbf{y}$. It operates on models which include unobserved data $\mathbf{s}$, $P_{\theta,\mathbf{y},\mathbf{s}}$. For our application, $\mathbf{y}$ is a sequence of observations $\{y(t)\}$ and $\mathbf{s}$ is a sequence of hidden states $\{\psi(t)\}$. Note, however, that the EM algorithm in its general form does not assume the unobserved data form a Markov chain.

The steps in any EM algorithm are:

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

The EM algorithm will work if $P_{\hat{\theta}}(\mathbf{y}) \geq P_\theta(\mathbf{y})$ with equality *iff* the likelihood is at a local maximum at θ. Brown[4] proceeds as follows:

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

[3]The first part of this section is excerpted, with only trivial changes, from [2]. The discussion there is in turn based on Brown[4]. Poritz[1] gives additional references.

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

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

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

$$\log P_{\hat{\theta}}(\mathbf{y}) = \left\langle \log P_{\hat{\theta}}(\mathbf{s},\mathbf{y}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta} - \left\langle \log P_{\hat{\theta}}(\mathbf{s}|\mathbf{y}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta} \tag{5}$$

The Gibbs inequality for two distributions P and Q says

$$\sum_x P(x) \log \frac{Q(x)}{P(x)} \leq 0 \tag{6}$$

In the current notation:

$$\langle \log P_{\theta_2}(x) \rangle_{\theta_1} \leq \langle \log P_{\theta_1}(x) \rangle_{\theta_1}$$

So

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

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

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

equation (5) yields

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

and the algorithm steps uphill. Since

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

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.

3.1 The maximization step

For a system described by equations (2) and (3), we can begin to see how step 2 of the EM algorithm can be carried out.

Let $\mathbf{y} \equiv y_1^T$, $\mathbf{s} \equiv \psi_1^{T+1}$. Then

$$\begin{aligned} P(\mathbf{y},\mathbf{s}) &= P(\psi(T{+}1), y(T) \mid \psi_1^T, y_1^{T-1}) P(\psi_1^T, y_1^{T-1}) \\ &= \prod_{t=1}^{T} P(\psi(t{+}1), y(t) \mid \psi_1^t, y_1^{t-1}) \times P(\psi(1)) \\ &= \prod_{t=1}^{T} P(\psi(t{+}1), y(t) \mid \psi(t)) \times P(\psi(1)) \end{aligned} \tag{8}$$

The task, then, is to maximize

$$\left\langle \log P_{\hat{\theta}}(\mathbf{y},\mathbf{s}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta} = \left\langle \log P_{\hat{\theta}}(\psi(1)) + \sum_{t=1}^{T} \log P_{\hat{\theta}}(\psi(t{+}1), y(t) | \psi(t)) \right\rangle_{\mathbf{s}|\mathbf{y},\theta} \tag{9}$$

In the next section a specific, computationally tractable model for time series is presented; section 5 discusses how expression (9) can be maximized for such models.

4 Computation with state space models

This section describes a concrete class of processes that can be described parametrically. The Hidden Filter and Mixed State types of HMM are defined.

We will now develop a computationally tractable special case of equations (2) and (3). We assume that we can decompose the hidden state $\psi(t)$ as the pair $(s(t), \mathbf{x}(t))$, where $\mathbf{x}(t)$ is the D-dimenstional history vector y_{t-D}^{t-1} and $s(t) \in S = \{s_1, s_2, ..., s_{\text{nstates}}\}$, a finite set. The state $\psi(t) \in S \times \mathbb{R}^D$ is referred to as a *mixed state.*

The process can be written in the form of equation (1) as

$$\begin{aligned} s(t+1) &= F_s(\psi(t)) + \eta_s(t) \;=\; q(\psi(t), t) \\ \mathbf{x}_1(t+1) &= F_1(\psi(t)) + \eta_{\mathbf{x}_1}(t) \;=\; f(\psi(t)) + \epsilon(\psi(t), t) \\ \mathbf{x}_i(t+1) &= F_i(\psi(t)) + \eta_{\mathbf{x}_i}(t) \;=\; \mathbf{x}_{i-1}(t) + 0 \qquad (i = 2, ..., D) \\ y(t) &= G(\psi(t)) + \eta_y(t) \;=\; f(\psi(t)) + \epsilon(\psi(t), t) \end{aligned} \tag{10}$$

I will refer to the class of all processes that fit equation (10) as *Autoregressive Markov processes*; here I will discuss two special cases of such

models, the Hidden Filter HMM first described by Poritz[1] and the Mixed State HMM, which as far as we know was introduced in [2] and is first described in detail in these pages.

The class of models that we call Mixed State HMMs corresponds to equations (10) with only the following additional assumption:

- The distribution $P(y(t), \mathbf{x}(t) \mid s(t) = s_i, s(t{+}1) = s_j)$ is multivariate normal with mean and covariance matrix specific to the transition $s_i \to s_j$. (11)

The class of Hidden Filter HMMs corresponds to equations (10) along with the additional assumptions:

1. The sequence of discrete states $\{s(t)\}$ forms a Markov process.
2. The distribution of each output $y(t)$ depends on the state $\psi(t)$ but is independent of all past *and future* discrete states. That is, $P(y(t)|\psi_1^t, s_{t+1}^T) = P(y(t)|\psi(t))$. From this it follows that $P(s(t{+}1) \mid \psi(t)) = P(s(t{+}1) \mid s(t))$.
3. The distribution $P(y(t), \mathbf{x}(t) \mid s(t) = s_i)$ is multivariate normal with mean and covariance matrix specific to the discrete state s_i. Thus given $s(t)$, the noise $\epsilon(\psi(t), t)$ is normally distributed and independent of $\mathbf{x}(t)$, and the prediction $f(\psi(t))$ is a linear function of $\mathbf{x}(t)$.

Thus a Mixed State HMM is a very general model of an AR Markov process, while the Hidden Filter HMM incorporates significant simplifying assumptions. Both types of model can be fully determined by a finite number of parameters and can be fitted reasonably efficiently, as will be shown in the following sections.

The remainder of this paper concentrates on Mixed State models. The equations appropriate to Hidden Filter models will be noted in section 5.6.

4.1 Parametrization of Mixed State models

A Mixed State model can be represented by the following parameters:

- The number of discrete states *nstates* and the dimension D of the history vector $\mathbf{x}$. Once chosen, these stay fixed.
- For each transition $s_i \to s_j$, the parameters of the distributions:

  $P(\mathbf{x}(t) \mid s(t{+}1) = s_j, s(t) = s_i)$ (typical $s_i \to s_j$ history vector),
  $P(y(t) \mid \mathbf{x}(t), s(t{+}1) = s_j, s(t) = s_i)$ (typical $s_i \to s_j$ output), and
  $P(s(t{+}1) = s_j \mid s(t) = s_i)$ (the overall probability of a transition).

The first is modeled as a multivariate normal (assumption (11)), the second as a normally distributed linear prediction (also assumption (11)), and the third is a constant. These are computed from the data and optimized iteratively.

- For each discrete state s_i, the probability $P(s(1) = s_i)$, a constant. This also is computed from the data.

These distributions are sufficient to calculate other probabilities of interest, as will be shown in the next section. The mathematics of parametrizing normal distributions will be discussed in section 5.2.

4.2 A menagerie of derived distributions

Using the above parameters, we have available[5]

$$P(\mathbf{x}(t{+}1)|s(t{+}1), s(t), \mathbf{x}(t)) = P(y(t)|s(t{+}1), s(t), \mathbf{x}(t))\ \delta(\mathbf{x}(t{+}1)_2^D - \mathbf{x}(t)_1^{D-1}) \quad (12)$$

$$P(\mathbf{x}(t) \mid s(t)) = \sum_j P(\mathbf{x}(t) \mid s(t{+}1){=}j, s(t)) P(s(t{+}1){=}j \mid s(t)), \quad (13)$$

and by use of the latter also

$$P(s(t{+}1){=}j \mid s(t){=}i, \mathbf{x}(t)) = \quad (14)$$
$$= \frac{P(\mathbf{x}(t) \mid s(t{+}1){=}j, s(t){=}i)}{P(\mathbf{x}(t) \mid s(t){=}i)} P(s(t{+}1){=}j \mid s(t){=}i)$$

Thus we can compute

$$P(\psi(t{+}1) \mid \psi(t)) =$$
$$= P(\mathbf{x}(t{+}1)|s(t{+}1), s(t), \mathbf{x}(t))\ P(s(t{+}1)|s(t), \mathbf{x}(t)) \quad (15)$$
$$= P(y(t)|s(t{+}1), s(t), \mathbf{x}(t))\ \delta(\mathbf{x}(t{+}1)_2^D - \mathbf{x}(t)_1^{D-1}) \quad (16)$$
$$\times \frac{P(\mathbf{x}(t) \mid s(t{+}1), s(t))}{P(\mathbf{x}(t) \mid s(t))} P(s(t{+}1) \mid s(t))$$

$$P(y(t) \mid \psi(t))) = \quad (17)$$
$$= \sum_j P(y(t) \mid s(t), s(t{+}1) = j, \mathbf{x}(t)) P(s(t{+}1) = j \mid s(t), \mathbf{x}(t))$$

[5]The notation $\delta(\mathbf{x})$, where $\mathbf{x}$ is a vector or real number, represents the Dirac delta, for which $\delta(\mathbf{x}) = 0$ for all $\mathbf{x} \neq 0$, but $\int \delta(\mathbf{x}) d\mathbf{x} = 1$.

In practice, it is more efficient to use

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

Finally, note that if we understand $\psi(t{+}1)$ to refer to the mixed state $(s(t{+}1), y(t), \mathbf{x}(t)_1^{D-1})$, we have

$$P(s(t{+}1) \mid \psi(t), y(t)) = P(\psi(t{+}1) \mid \psi(t), y(t)) = \frac{P(\psi(t{+}1) \mid \psi(t))}{P(y(t) \mid \psi(t))} \tag{18}$$

5 How to fit a better model

This section derives an explicit optimization algorithm for Mixed State Markov models. The arithmetic of modeling multivariate normal distributions is also presented.

We can now return to the optimization of $\left\langle \log P_{\hat{\theta}}(\mathbf{y}, \mathbf{s}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta}$ as given in equation (9), which is required in order to carry out the EM algorithm. Since in AR Markov models $y(t) = \mathbf{x}_1(t{+}1)$ is actually a coordinate of $\psi(t{+}1)$, we have for such models

$$P(y_1^T, \psi_1^{T+1}) = P(\psi(1)) \times \prod_{t=1}^{T} P(\psi(t{+}1) \mid \psi_1^t, y_1^{t-1}) \, \delta(y(t) - \mathbf{x}_1(t{+}1))$$

Note also that for such models, $P(\psi(t) \mid y_1^T) = P(s(t) \mid y_1^T) \, \delta(\mathbf{x}(t) - y_{t-D}^{t-1})$. For convenience we arrange for $\mathbf{x}(1)$ to be known (easily done by leaving the first few available outputs out of y_1^T) and assign it a probability of 1. Then y_1^T, s_1^{T+1} completely determine a mixed state sequence $\mathbf{s} = \psi_1^{T+1}$; the ensemble $\left\langle \log P_{\hat{\theta}}(\mathbf{y}, \mathbf{s}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta}$ can be replaced with

$$\left\langle \log P_{\hat{\theta}}(y_1^T, q) \right\rangle_{q|y_1^T,\theta},$$

which is taken over all sequences $q = s_1^{T+1}$ of discrete states, and we can replace equation (9) with

$$\begin{aligned} \left\langle \log P_{\hat{\theta}}(\mathbf{y}, \mathbf{s}) \right\rangle_{\mathbf{s}|\mathbf{y},\theta} &= \left\langle \log P_{\hat{\theta}}(\psi(1)) + \sum_{t=1}^{T} \log P_{\hat{\theta}}(\psi(t{+}1)|\psi(t)) \right\rangle_{q|y_1^T,\theta} \qquad (19) \\ &= \sum_q P_\theta(q|y_1^T) \log P_{\hat{\theta}}(\psi(1)) + \sum_{q,t} P_\theta(q|y_1^T) \log P_{\hat{\theta}}(\psi(t{+}1)|\psi(t)) \\ &\equiv A(\hat{\theta}) + B(\hat{\theta}) \end{aligned}$$

The second term of this can be converted to a sum over s and t. We begin by writing it as

$$B(\hat{\theta}) = \sum_t \sum_q \sum_i \delta_{q(t),i} \sum_j \delta_{q(t+1),j} P_\theta(q \mid y_1^T) \log P_{\hat{\theta}}(\psi(t{+}1)|\psi(t))$$

Since the only non-zero terms to this occur when $q(t) = i, q(t+1) = j$, we can replace $P_{\hat{\theta}}(\psi(t{+}1) \mid \psi(t))$ with $P_{\hat{\theta}}(\psi(t{+}1) = (j, \mathbf{x}(t{+}1)) \mid \psi(t) = (i, \mathbf{x}(t))$.

$$B(\hat{\theta}) = \sum_t \sum_i \sum_j \left(\sum_q \delta_{q(t),i} \delta_{q(t+1),j} P_\theta(q \mid y_1^T) \right) \log P_{\hat{\theta}}((j, \mathbf{x}(t{+}1)) \mid (i, \mathbf{x}(t)))$$

To simplify the quantity in parentheses, we note that since $s(t), s(t{+}1)$ are determined by q, we can write $\delta_{q(t),i}\delta_{q(t+1),j} = P_\theta(s(t){=}i, s(t{+}1){=}j \mid q, y_1^T)$. Thus we have

$$\begin{aligned} \sum_q \delta_{q(t),i}\delta_{q(t+1),j} P_\theta(q \mid y_1^T) &= \sum_q P_\theta(s(t) = i, s(t{+}1) = j, q \mid y_1^T) \\ &= P_\theta(s(t) = i, s(t{+}1) = j \mid y_1^T) \\ &\equiv w(j, i, t) \end{aligned}$$

where $w(j, i, t)$ is the total probability over all paths q that the state at time t is i and the state at time $t+1$ is j, given that the sequence of outputs is y_1^T, i.e.,

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

We can now write

$$B(\hat{\theta}) = \sum_{t,i,j} w(j, i, t) \log P_{\psi(t+1)|\psi(t),\hat{\theta}}((j, \mathbf{x}(t{+}1)) \mid (i, \mathbf{x}(t))) \tag{20}$$

It is a similar, and much simpler, matter to rewrite $A(\hat{\theta})$ as a sum over $s(1)$, and since we consider $\mathbf{x}(1)$ to be given,

$$\begin{aligned} A(\hat{\theta}) &= \sum_i P_\theta(s(t){=}i \mid y_1^T) \log P_{\hat{\theta}}(\psi(1) = (i, \mathbf{x}(1))) \\ &= \sum_i w(i, 1) \log P_{\hat{\theta}}(\mathbf{s}(1) = i), \end{aligned}$$

where $w(i, t) \equiv P_\theta(s(t) = i \mid y_1^T) = \sum_j w(j, i, t)$.

Thus for an AR Markov process, the EM algorithm has been reduced to the problem of maximizing the finite sum $A(\hat{\theta}) + B(\hat{\theta})$. Note that the number of summands in $B(\hat{\theta})$ is linear in t.

5.1 Fitting a Mixed State model: the naive algorithm

For a Mixed State HMM, equation (16) allows us to write

$$
\begin{aligned}
B(\hat{\theta}) = & \\
& \sum_{t,i,j} w(j,i,t) \log P_{\hat{\theta}}(y(t)|s(t{+}1){=}j, s(t){=}i, \mathbf{x}(t)) \\
& + \sum_{t,i,j} w(j,i,t) \log P_{\hat{\theta}}(\mathbf{x}(t) \mid s(t+1){=}j, s(t){=}i) \\
& + \sum_{t,i,j} w(j,i,t) \log P_{\hat{\theta}}(s(t{+}1){=}j \mid s(t){=}i) \\
& - \sum_{t,i} w(i,t) \log P_{\hat{\theta}}(\mathbf{x}(t) \mid s(t){=}i)
\end{aligned}
\tag{21}
$$

We now come to the one big hole in the development of Mixed State HMMs: we do not have an efficient algorithm that we have shown to maximize $A(\hat{\theta}) + B(\hat{\theta})$. Although it is possible to maximize this by global optimization of all parameters of the model $\hat{\theta}$, such optimization is very time-consuming. However, it is possible to optimize, separately, each of the terms

$$
\begin{aligned}
A(\hat{\theta}) &= \sum_{i} w(i,1) \log P_{\hat{\theta}}(\mathbf{s}(1) = i) \\
F(\hat{\theta}) &\equiv \sum_{t,i,j} w(j,i,t) \log P_{\hat{\theta}}((y(t)|s(t{+}1){=}j, s(t){=}i, \mathbf{x}(t)) \\
G(\hat{\theta}) &\equiv \sum_{t,i,j} w(j,i,t) \log P_{\hat{\theta}}(\mathbf{x}(t) \mid s(t+1){=}j, s(t){=}i) \\
H(\hat{\theta}) &\equiv \sum_{t,i,j} w(j,i,t) \log P_{\hat{\theta}}(s(t{+}1){=}j \mid s(t){=}i)
\end{aligned}
\tag{22}
$$

Doing so exhausts the free parameters of the model. We refer to the strategy of maximizing these and ignoring the denominator term $-\sum_{t,i} w(i,t) \log P_{\hat{\theta}}(\mathbf{x}(t) \mid s(t){=}i)$ as the "naive algorithm". It turns out that application of this naive algorithm yields essentially monotonic improvement in likelihood, with each model very close to the true optimal performance for that step in the process (which we estimated by global optimization of the parameters of $\hat{\theta}$).

Maximizing the above expressions requires the quantity $w(j,i,t) = P(s(t1){=}j, s(t){=}i \mid y_1^T)$. This can be efficiently computed by the *forward backward algorithm,* which is the subject of section 6. Given $w(j,i,t)$, the following sections show how to maximize the expressions of (22).

5.2 Modeling normal distributions

This rather parenthetical section covers the arithmetic of modeling multivariate normal distributions.

A multivariate normal distribution is parametrized by its mean $\overline{\mu}$ and covariance matrix $\mathbf{C}$. For the standard normal distribution (mean $\mathbf{0}$, unit covariance matrix), the probability density of a point $\mathbf{x}$ is given by

$$P_I(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2}\|\mathbf{x}\|^2} \tag{23}$$

For other normal distributions, the density is computed essentially by changing to a set of coordinates that make the distribution into a standard normal cloud and then applying formula (23), with the added complication of a rescaling factor equal to the determinant of the coordinate change.

If $\mathbf{T}$ is a matrix giving a coordinate change such that the map $f(\mathbf{x}) \equiv \mathbf{T}(\mathbf{x} - \overline{\mu})$ maps a set of vectors to a standard normal distribution, the probability density is given by

$$P(\mathbf{x}) = \frac{|\mathbf{T}|}{(2\pi)^{n/2}} e^{-\frac{1}{2}(T(\mathbf{x}-\overline{\mu}))^t \cdot (T(\mathbf{x}-\overline{\mu}))} \tag{24}$$

Let $\mathbf{A}$ be a collection of N column vectors with mean $\mathbf{0}$. Then its covariance matrix is $\mathbf{C} = 1/N\, \mathbf{A} \times \mathbf{A}^t$. In general, if a collection of vectors $\mathbf{X}$ has covariance matrix $\mathbf{C}$ and $\mathbf{T}$ is a coordinate change matrix, $\mathbf{TX}$ will have covariance matrix $1/N\, \mathbf{TX} \times (\mathbf{TX})^t = \mathbf{TCT}^t$. Thus if $\mathbf{TX}$ has identity covariance matrix, $\mathbf{T}^t\mathbf{T} = \mathbf{C}^{-1}$.

Accordingly, equation (24) is equivalent to the more common form

$$P(\mathbf{x}) = \frac{\left|\mathbf{C}^{-1}\right|^{1/2}}{(2\pi)^{n/2}} e^{-\frac{1}{2}(\mathbf{x}-\overline{\mu})^t\ \mathbf{C}^{-1}(\mathbf{x}-\overline{\mu})}$$

To model a normal distribution, it is thus sufficient to keep as parameters the mean $\overline{\mu}$ and either the inverse covariance matrix $\mathbf{C}^{-1}$ or the coordinate change matrix $\mathbf{T}$. We have used both methods at different times. I prefer to use the coordinate change matrix, since obtaining the inverse covariance matrix from it is trivial, while the converse is more complicated.

Fitting a multivariate normal

Given a collection of column vectors $\mathbf{X}$ as before, let $\mathbf{A}$ be the array of row vectors $(\mathbf{x} - \overline{\mu})^t$, for $\mathbf{x} \in \mathbf{X}$. Note that $\mathbf{X}$, $\mathbf{A}$ have the same covariance matrix.

The method of Singular Value Decomposition[6] converts the array $\mathbf{A}$ to $\mathbf{A} = \mathbf{UWV}^t$, where $\mathbf{W}$ is diagonal, $\mathbf{V}$ has orthonormal rows and columns, $\mathbf{U}$ has orthonormal columns, $\mathbf{U}^t\mathbf{U} = \mathbf{V}^t\mathbf{V} = \mathbf{V}\mathbf{V}^t = 1$. (If $\mathbf{U}$ is square, it is also row-orthonormal).

Then $\mathbf{A}^t\mathbf{A} = \mathbf{VWU}^t\,\mathbf{UWV}^t = \mathbf{V}\,\mathbf{W}^2\,\mathbf{V}^t$. Since $\mathbf{V}$ is orthogonal, we have $\mathbf{V}^{-1} = \mathbf{V}^t$. Then $\mathbf{A}(\sqrt{N}\,\mathbf{VW}^{-1})$ has covariance matrix

$$1/N\,(\mathbf{AVW}^{-1})^t(\mathbf{AVW}^{-1}) = 1/N\,\sqrt{N}\mathbf{W}^{-1}\mathbf{V}^t\mathbf{A}^t\sqrt{N}\mathbf{AVW}^{-1} = \mathbf{I}$$

Thus the coordinate change that takes $\mathbf{X}$ to a standard normal cloud has matrix

$$\mathbf{T} = (\sqrt{N}\mathbf{VW}^{-1})^t = \sqrt{N}\mathbf{W}^{-1}\mathbf{V}^t. \tag{25}$$

When given a set of weights w along with the vectors $\mathbf{X}$, we first calculate the weighted mean

$$\overline{\mu} = 1/\Big(\sum_{i=1}^{N} w_i\Big)\sum_i w_i\mathbf{X}_i, \tag{26}$$

where $\mathbf{X}_i$ is the i-th column vector of $\mathbf{X}$. Note that $(\mathbf{T}a\mathbf{x})^t\,(\mathbf{T}a\mathbf{x}) = a^2(\mathbf{Tx})^t(\mathbf{Tx})$. Form $\mathbf{A}$ by letting $\mathbf{A}_i = \sqrt{s}\cdot\sqrt{w}_i(\mathbf{X}_i - \overline{\mu})^t$, so that vector $\mathbf{X}_i$ contributes to the covariance with weight w_i and $s = N/\sum_i w_i$ makes the entire sample have weight N, as if no weights were used. Calculate $\mathbf{T}$ as above.

5.3 Fitting the typical vector term, $\mathbf{G}(\hat{\theta})$

The quantity

$$G(\hat{\theta}) = \sum_{t,i,j} w(j,i,t)\log P_{\hat{\theta}}(\mathbf{x}(t)\mid s(t+1){=}j, s(t){=}i)$$

can be maximized separately for each transition $i \to j$. By the Gibbs inequality, the normal distribution that maximizes $G(\hat{\theta})$ is the distribution with the mean and covariance matrix of the weighted sample of vectors $(w(j,i,t), \mathbf{x}(t))$. This is computed as described in section 5.2.

[6]See Section 2.9 of [5].

5.4 Fitting the output term, $\mathbf{F}(\hat{\theta})$

Given $w(j,i,t)$, $F(\hat{\theta})$ can be maximized separately for each transition $i \to j$. The normal distribution $P(y(t) \mid s(t{+}1){=}j, s(t){=}i, \mathbf{x}(t))$ is determined by its variance $\sigma_{i,j}$ and its mean, which is an affine linear function of $\mathbf{x}(t)$ given by $\hat{\mathbf{y}}(t) = \overline{y}_{i,j} + \mathbf{a}_{i,j} \cdot \mathbf{x}(t)$. Finding new values for $\overline{y}_{i,j}$, $\mathbf{a}_{i,j}$, and $\sigma_{i,j}$ is fairly standard linear fitting. Solve for $\mathbf{a}_{i,j}$ and $\overline{y}_{i,j}$ using Singular Value Decomposition[7] to minimize

$$\chi^2 = \sum_t \left\{ y(t)\sqrt{w(j,i,t)} - (\overline{y}_{i,j} + \mathbf{a}_{i,j} \cdot \mathbf{x}(t))\sqrt{w(j,i,t)} \right\}^2,$$

and set

$$\sigma_{i,j} = \sqrt{\frac{1}{\sum_t w(j,i,t)} \sum_t w(j,i,t)\,(y(t) - \hat{y}(t))^2}.$$

5.5 Fitting the state transition terms, $\mathbf{A}(\hat{\theta})$ and $\mathbf{H}(\hat{\theta})$

Since the a priori transition probabilities are the same for all values of t, we can write

$$H(\hat{\theta}) = \sum_i \sum_j \Big(\sum_t w(j,i,t) \Big) \log P_{s(t+1)|s(t),\hat{\theta}}(j \mid i)$$

We can maximize this quantity separately for each i: By the Gibbs inequality (equation (6)), $\sum_j P(j) \log Q(j)$ is maximized when $Q = P$. Thus $H(\hat{\theta})$ is maximized when for each i, we set

$$P_{s(t+1)|s(t),\hat{\theta}}(j \mid i) \propto \sum_t w(j,i,t)$$

Similarly we maximize $A(\hat{\theta})$ by setting

$$P_{\hat{\theta}}(\mathbf{s}(1) = i) \;=\; w(i,1)$$

5.6 Fitting Hidden Filter models

We now return to the Hidden Filter model defined in section 4. This section describes how the computations of the preceding sections apply to this class of models.

A Hidden Filter model is defined by the values of the following parameters (compare section 4.1).

[7] See equation 14.3.16 on page 535 of [5]

- The number of discrete states *nstates* and the dimension D of the history vector $\mathbf{x}$. Once chosen, these stay fixed.
- For each transition $s_i \to s_j$, the probability $P(s(t{+}1) = s_j \mid s(t) = s_i)$.
- For each discrete state s_i, the probabilities $P(s(1) = s_i)$ and $P(y(t) \mid s(t) = s_i, \mathbf{x}(t))$.

Of these the last is a normally distributed affine linear prediction, the others are constants. All are computed from the data and optimized iteratively.

For this class of models equation (16) reduces to

$$P(\psi(t{+}1) \mid \psi(t)) = P(s(t{+}1) \mid s(t))P(y(t)|s(t), \mathbf{x}(t))$$

and accordingly we can replace expression (21) with

$$\begin{aligned} B(\hat{\theta}) \quad = \quad & \sum_{t,i} w(i,t) \log P_{\hat{\theta}}(y(t)|s(t){=}i, \mathbf{x}(t)) \\ & + \sum_{t,i,j} w(j,i,t) \log P_{\hat{\theta}}(s(t{+}1){=}j \mid s(t){=}i) \end{aligned}$$

Each of these terms (and the term $A(\hat{\theta})$, which is the same as for the Mixed State models) can be maximized separately, as shown in the preceding sections. In this case, since we do not ignore any part of expression (9), the EM algorithm guarantees that $P_{\hat{\theta}}(y_1^T) \geq P_{\theta}(y_1^T)$.

6 The forward-backward algorithm

Here we show how to compute the last missing piece, the quantity $w(j,i,t)$, via the forward backward algorithm.[8]

The factor $w(j,i,t) = P(s(t+1) = j, s(t) = i|y_1^T)$, used throughout the optimizations of section 5, can be calculated with the help of the forward backward algorithm, which in turn depends on the αs and βs 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:

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

[8]This section is adapted, step for step, from the corresponding section in [2], where the computations appropriate for Hidden Filter HMMs are developed.

$\beta(s,t)$ 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:

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

If we let $W \equiv 1/P(y_1^T)$, we have the following identities:

$$w(s,t) = W\,\alpha(s,t)\,\beta(s,t) \tag{27}$$

$$\begin{aligned} w(j,i,t) &= W\,\beta(j,t{+}1)\,P(y(t+1) \mid \psi(t{+}1) = (j, \mathbf{x}(t{+}1)))\\ &\qquad \times P(s(t{+}1) = j \mid \psi(t) = (i, \mathbf{x}(t)), y(t))\,\alpha(i,t) \end{aligned} \tag{28}$$

The derivation of $w(s,t)$ is trivial. For $w(j,i,t)$, we note that

$$\begin{aligned} &P(y_{t+2}^T \mid s_1^{t+1}, y_1^{t+1})\\ &\quad = P(y_{t+2}^T \mid \psi_1^{t+1}, y(t{+}1))\\ &\quad = P(y_{t+2}^T \mid \psi(t{+}1), y(t{+}1))\\ &\quad = P(y_{t+2}^T \mid s(t{+}1), y_{t-D+1}^{t+1}) \end{aligned}$$

By equation (4), it then follows that

$$\begin{aligned} &P(y_{t+2}^T \mid s(t+1) = j, s(t) = i, y_1^{t+1})\\ &\quad = P(y_{t+2}^T \mid s(t+1) = j, y_1^{t+1})\\ &\quad = \beta(j, t+1). \end{aligned}$$

Thus we have

$$\begin{aligned} \overline{w}(j,i,t) &= P(s(t{+}1) = j, s(t) = i, y_1^T)/P(y_1^T)\\ &= W\,P(y_{t+2}^T \mid s(t{+}1) = j, s(t) = i, y_1^{t+1})\\ &\qquad \times P(y(t{+}1) \mid s(t{+}1) = j, s(t) = i, y_1^t)\\ &\qquad \times P(s(t{+}1) = j \mid s(t) = i, y_1^t) P(s(t) = i, y_1^t)\\ &= W\beta(j,t{+}1)\,P(y(t{+}1) \mid \psi(t{+}1) = (j, \mathbf{x}(t{+}1)))\\ &\qquad \times P(s(t{+}1) = j \mid \psi(t) = (i, \mathbf{x}(t)), y(t))\,\alpha(i,t) \end{aligned}$$

The quantities α and β can in turn be evaluated by the following recursion formulas.

$$\begin{aligned} &\alpha(j,t) = P(s(t) = j, y_1^t)\\ &\quad = \sum_i P(s(t) = j, s(t-1) = i, y_1^t) \end{aligned}$$

$$
\begin{aligned}
&= \sum_i P(y(t) \mid s(t) = j, s(t\text{–}1) = i, y_1^{t-1}) \\
&\quad \times P(s(t) = j \mid s(t\text{–}1) = i, y_1^{t-1}) P(s(t\text{–}1) = i, y_1^{t-1}) \\
&= \sum_i P(y(t) \mid \psi(t) = (j, \mathbf{x}(t)) \,) \, P(s(t) = j \mid \psi(t\text{–}1) = (i, \mathbf{x}(t\text{–}1)), y(t\text{–}1)) \, \alpha(i, t\text{–}1) \\
&= P(y(t) \mid \psi(t) = (j, \mathbf{x}(t)) \,) \sum_i P(s(t) = j \mid \psi(t\text{–}1) = (i, \mathbf{x}(t\text{–}1)), y(t\text{–}1)) \, \alpha(i, t\text{–}1)
\end{aligned}
$$

$$
\begin{aligned}
\beta(i,t) &= P(y_{t+1}^T \mid s(t) = i, y_1^t) \\
&= \sum_j P(y_{t+1}^T, s(t\text{+}1) = j \mid s(t) = i, y_1^t) \\
&= \sum_j P(y_{t+1}^T \mid s(t\text{+}1) = j, s(t) = i, y_1^t) P(s(t\text{+}1) = j \mid s(t) = i, y_1^t) \\
&= \sum_j \big(P(y_{t+2}^T \mid s(t\text{+}1) = j, s(t) = i, y_1^{t+1}) \\
&\quad \times P(y(t\text{+}1) \mid s(t\text{+}1) = j, s(t) = i, y_1^t) \\
&\quad \times \; P(s(t\text{+}1) = j \mid s(t) = i, y_1^t) \big) \\
&= \sum_j (\beta(j, t\text{+}1) \; P(y(t\text{+}1) \mid \psi(t\text{+}1) = (j, \mathbf{x}(t\text{+}1)) \,) \\
&\quad \times \; P(s(t\text{+}1) = j \mid \psi(t) = (i, \mathbf{x}(t)), y(t)))
\end{aligned}
$$

The recursion is initialized from the values

$$
\begin{aligned}
\alpha(s,1) &= P(s(1) = s, y(1)) = P(y(1) \mid \psi(1) = (s, \mathbf{x}(1)) \,) P(s(1) = s \mid \mathbf{x}(1)) \\
\beta(s,T) &= P(\emptyset \mid s(T) = s, y_1^T) = 1
\end{aligned}
$$

A convenient way to represent the matrices α and β in a way that avoids floating-point overflow is presented in [2], and easily adapts to the requirements of Mixed State models.

7 Acknowledgements etc.

The portion of the work described here that I am responsible for was supported by funding from the PSU Systems Science Ph.D. program and by an NSF grant to Andrew Fraser. I am also grateful to him for teaching me about time series; I would not know a thing about them if it were not for him.

I would also like to thank Mara Tableman, who taught me statistics and has been a great resource numerous times since then; and Tom Shuell and Ray Melton, who were reckless enough to ask me what on earth I was working on, leading to lengthy explanations that now form the first section of this paper.

References

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

[2] A. M. Fraser and A. Dimitriadis. Forecasting probability densities by using Hidden Markov Models with mixed states. In A. S. Weigend and N. A. Gershenfeld, editors, *Time Series Prediction: Forecasting the Future and Understanding the Past*, pages 264–281. Addison-Wesley, 1994.

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

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

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

[6] 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.

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