

A literature review of Bayesian Hierarchical Hidden Markov Models applied to financial time series.

Luis Damiano, Brian Peterson, Michael Weylandt

2017-06-03

Abstract

This is the abstract. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras facilisis mi eu diam pellentesque, ac consequat enim ultricies. Duis convallis massa neque, sit amet molestie odio sagittis et. Duis aliquet dolor elementum, tempor justo in, aliquam libero. Ut non ligula gravida, sodales mauris sed, gravida enim. Vestibulum magna dui, congue et elementum vel, vestibulum et quam. Suspendisse sit amet felis sapien. Etiam scelerisque nisl ac gravida elementum. Curabitur laoreet erat ac lorem tempus ultrices. Integer ut placerat magna, eu fermentum nibh. Donec a dignissim dui, at elementum mauris.

1 Introduction

1

2 Markov chains

2

2.1 Definitions

Let $z_{1:T} = \{z_1, \dots, z_T\}$ be a sequence of regularly spaced observations of arbitrary length T . The index $t = 1, \dots, T$ reflects discrete time or space steps. The sequence of random variables has the Markovian property if the probability of moving to the next state depends only on the present state and not on the previous ones.

2.2 Transition function (joint distribution)

Let $p(z_t|z_{t-m:t-1})$ be the transition function, where m is finite. Assuming the conditional probabilities are well defined, the joint distribution of a Markov chain of order m , or a Markov chain with memory m , given the parameters θ can be derived with the chain rule of probability

$$p(z_{1:T}) = p(z_{1:m}) \prod_{t=m+1}^T p(z_t|z_{t-m:t-1}),$$

where conditioning on the fixed parameters θ was removed to increase readability.

For first-order Markov chains, by the chain rule of probability, the expression simplifies to

$$p(z_{1:T}) = p(z_1)p(z_2|z_1)p(z_3|z_2) \dots p(z_T|z_{T-1}) = p(z_1) \prod_{t=2}^T p(z_t|z_{t-1}).$$

¹This is the introduction. Neither the theory of hidden Markov models nor its applications to financial processes is new.

²Some brief intro. Since this is a preamble for HMM, this writing focus on finite-state Markov chain. It comes in many other flavours tho.

When the transition function is independent of the step index, the chain is called homogeneous, stationary, or time-invariant and the parameters are shared by multiple variables.

If the observed variable only takes one of K possible values, so that $z_t \in S = \{1, \dots, K\}$, the model is called a discrete-state or finite-state Markov chain. The possible values of z_t form a countable set S called the state space of the chain.

2.3 Transition matrix for a finite-state Markov chain (conditional distribution)

In the context of a finite-state Markov chain, the one-step transition matrix \mathbf{A} is a $K \times K$ stochastic matrix with elements $A_{ij} = p(z_t = j | z_{t-1} = i)$ with $i, j \in S$ that satisfies $\sum_j A_{ij} = 1$ for all rows i and $0 \leq A_{ij} \leq 1$ for all entries i, j . Each element specifies the probability of transition from i to j in one step. Given the constraints, the matrix has $K(K-1)$ independent parameters.

The n -step transition matrix $\mathbf{A}(n)$ has elements $A_{ij}(n) = p(z_{t+n} = j | z_t = i)$ representing the probability of getting from i to j in exactly n steps. By definition, $\mathbf{A}(1) = \mathbf{A}$. By the Chapman-Kolmogorov equations,

$$A_{ij}(n+s) = \sum_{l=1}^K A_{il}(n) A_{jl}(s),$$

the probability of transitioning from i to j in exactly $n+s$ steps is the probability of getting from i to l in n steps and then from l to j in s steps, summed up over all l . Since this is equivalent to matrix multiplication,

$$\mathbf{A}(n+s) = \mathbf{A}(n)\mathbf{A}(s),$$

multiple steps can be naturally computed by exponentiation

$$\mathbf{A}(n) = \mathbf{A}(1)\mathbf{A}(n-1) = \mathbf{A}(1)\mathbf{A}(1)\mathbf{A}(n-2) = \dots = \mathbf{A}^n.$$

2.4 State probabilities (marginal distribution)

Let $\pi_t(j) = p(z_t = j)$ be the probability that the random variable is in state j at the step t and $\boldsymbol{\pi}$ be a row vector called the state probability vector. Given the initial state distribution $\boldsymbol{\pi}_0$, the state probabilities for the next step can be computed by $\boldsymbol{\pi}_1 = \boldsymbol{\pi}_0 \mathbf{A}$.

A chain is said to have reached its stationary, invariant or equilibrium distribution when the following condition becomes true after many iterations

$$\boldsymbol{\pi} = \boldsymbol{\pi} \mathbf{A}.$$

This distribution does not always exist, but if does, the process cannot leave after entering this stage. There are different ways to prove if a chain is stationary, the most popular set of necessary conditions include irreducibility, recurrency and a limiting distribution that does not depend on the initial values, which in turns requires aperiodicity. The equilibrium is characterized by the the global balance equations. We refer to Doob (1953) for a detailed study on the existence of a stationarity distribution and its computation.

3 Hidden Markov Models

Real-world processes produce observable outputs characterized as signals. These can be discrete or continuous in nature, be pure or contaminated with noise, come from a stationary or non stationary source, among many other variations. These signals are modelled to allow for both theoretical descriptions and practical applications. The model itself can be deterministic or stochastic, in which case the signal is well characterized as a parametric random process whose parameters can be estimated in a well-defined manner.

Autocorrelation, a key feature in most signals, can be modelled in countless forms. While certainly pertinent to this purpose, high order Markov chains can prove inconvenient when the range of the correlation amongst the observations is long. A more parsimonious approach assumes that the observed sequence is a noisy observation of an underlying hidden process represented as a first-order Markov chain. In other terms, long-range dependencies between observations are mediated via latent variables. It is important to note that the Markov property is only assumed for the hidden states and not for the observations themselves.

3.1 Model specification

HMM involves two interconnected models. The state model consists of a discrete-time, discrete-state, first order hidden Markov chain $z_t \in \{1, \dots, K\}$ that transitions according to $p(z_t|z_{t-1})$. Additionally, the observation model is governed by $p(\mathbf{x}_t|z_t)$, where \mathbf{x}_t are the observations, emissions or output. The corresponding joint distribution is

$$p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}) = p(\mathbf{z}_{1:T})p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}) = \left[p(z_1) \prod_{t=2}^T p(z_t|z_{t-1}) \right] \left[\prod_{t=1}^T p(\mathbf{x}_t|z_t) \right].$$

It is a specific instance of the state space model family in which the latent variables are discrete. Each single time slice corresponds to a mixture distribution with component densities given by $p(\mathbf{x}_t|z_t)$, thus HMM may be interpreted as an extension of a mixture model in which the choice of component for each observation is not selected independently but depends on the choice of component for the previous observation. In the case of a simple mixture model for an independent and identically distributed sample, the parameters of the transition matrix inside the i -th column are the same, so that the conditional distribution $p(z_t|z_{t-1})$ is independent of z_{t-1} .

When the output is discrete, the observation model commonly takes the form of an observation matrix

$$p(\mathbf{x} = l|z_t = k, \boldsymbol{\theta}) = B(k, l).$$

Alternatively, if the output is continuous, the observation model is frequently a conditional Gaussian

$$p(\mathbf{x}_t|z_t = k, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

The latter is equivalent to a Gaussian mixture model with cluster membership ruled by Markovian dynamics, also known as Markov Switching Models (MSM). In this context, multiple sequential observations tend to share the same location until they suddenly jump into a new cluster.

3.2 Characteristics

By specification of the latent model, the density function of the duration τ in state i is given by

$$p_i(\tau) = (A_{ii})^\tau (1 - A_{ii}) \propto \exp(-\tau \ln A_{ii}),$$

which represents the probability that a sequence spends precisely τ steps in state i . The expected duration conditional on starting in that state is

$$\bar{\tau}_i = \sum_{\tau=1}^{\infty} \tau p_i(\tau) = \frac{1}{1 - A_{ii}}.$$

The density is an exponentially decaying function of τ , thus longer durations are more probable than shorter ones. In applications where this proves unrealistic, the diagonal coefficients of the transition matrix $A_{ii} \forall i$ may be set to zero and each state i is explicitly associated with a probability distribution of possible duration times $p(\tau|i)$ (Rabiner 1990).

One of the most powerful properties of HMM is the ability to exhibit some degree of invariance to local warping of the time axis. Allowing for compression or stretching of the time, the model accommodates for variations in speed.

3.3 Inference

There are several quantities of interest to be inferred from different algorithms. These are summarized in the following table. In this section, the discussion assumes that model parameters θ are known.

Table 1: Summary of the quantities that can be inferred and their corresponding algorithms.

Name	Quantity	Availability at	Algorithm	Complexity
Filtering	$p(z_t \mathbf{x}_{1:t})$	t (online)	Forward	?
Smoothing	$p(z_t \mathbf{x}_{1:T})$	T (offline)	Forwards-backwards	$O(K^2T)$ $O(KT)$ if left-to-right
Fixed lag smoothing	$p(z_{t-\ell} \mathbf{x}_{1:t}), \ell > 0$	$t + \ell$ (lagged)	?	?
State prediction	$p(z_{t+h} \mathbf{x}_{1:t}), h > 0$	t	?	?
Observation prediction	$p(x_{t+h} \mathbf{x}_{1:t}), h > 0$	t	?	?
MAP Estimation	\$ 1+2 \$	T	Viterbi encoding	$O(K^2T)$ $O(KT)$ if sparse
Probability of the evidence	$p(\mathbf{x}_{1:T})$	T	?	?

3.3.1 Filtering

A filter infers the belief state at a given step based on all the information available up to that point $p(z_t | \mathbf{x}_{1:t})$. It achieves better noise reduction than simply estimating the hidden state based on the current estimate $p(z_t | \mathbf{x}_t)$. The filtering process can be run online, or recursively, as new data streams in.

Filtered marginals can be computed recursively by means of the forward algorithm (Leonard E. Baum and Eagon 1967). Let $\psi_t(j) = p(\mathbf{x}_t | z_t = j)$ be the local evidence at step t and $\Psi(i, j) = p(z_t = j | z_{t-1} = i)$ be the transition probability. First, the one-step-ahead predictive density is computed

$$p(z_t = j | \mathbf{x}_{1:t-1}) = \sum_i \Psi(i, j) p(z_{t-1} = i | \mathbf{x}_{1:t-1}).$$

Acting as prior information, this quantity is updated with observed data at the step t using Bayes rule,

$$\begin{aligned}\alpha_t(j) &\triangleq p(z_t = j | \mathbf{x}_{1:t}) \\ &= p(z_t = j | \mathbf{x}_t, \mathbf{x}_{1:t-1}) \\ &= Z_t^{-1} \psi_t(j) p(z_t = j | \mathbf{x}_{1:t-1})\end{aligned}$$

where the normalization constant is given by

$$Z_t \triangleq p(\mathbf{x}_t | \mathbf{x}_{1:t-1}) = \sum_l p(\mathbf{x}_t | z_t = l) p(z_t = l | \mathbf{x}_{1:t-1}) = \sum_l \psi_t(l) p(z_t = l | \mathbf{x}_{1:t-1}).$$

This predict-update cycle results in the filtered belief states at step t . As this algorithm only requires the evaluation of the quantities $\psi_t(j)$ for each value of z_t for every t and fixed \mathbf{x}_t , the posterior distribution of the latent states is independent of the form of the observation density or indeed of whether the observed variables are continuous or discrete (Jordan 2003).

Let α_t be a K -sized vector with the filtered belief states at step t , $\psi_t(j)$ be the K -sized vector of local evidence at step t , Ψ be the transition matrix and $\mathbf{u} \odot \mathbf{v}$ is the Hadamard product, representing elementwise vector multiplication. Then, the bayesian updating procedure can be expressed in matrix notation as

$$\alpha_t \propto \psi_t \odot (\Psi^T \alpha_{t-1}).$$

In addition to computing the hidden states, the algorithm yields the log probability of the evidence

$$\log p(\mathbf{x}_{1:T} | \theta) = \sum_{t=1}^T \log p(\mathbf{x}_t | \mathbf{x}_{1:t-1}) = \sum_{t=1}^T \log Z_t.$$

Log domain should be preferred to avoid numerical underflow³.

Algorithm 1: Forward Algorithm

input : Transition matrix Ψ , local evidence vector ψ_t and initial state distribution π .

output: Belief state vector $\alpha_{1:T}$ and log probability of the evidence $\log p(\mathbf{x}_{1:T} = \sum_t \log Z_t)$.

```

1 def normalize( $\mathbf{u}$ ):
2    $Z = \sum_j u_j$ 
3    $v_j = u_j / Z$ 
4   return  $\mathbf{v}$ ,  $Z$ 

5  $\alpha_1, Z_1 = \text{normalize}(\psi_1 \odot \pi)$ 
6 for  $t = 2$  to  $T$  do
7    $\alpha_t, Z_t = \text{normalize}(\psi_t \odot (\Psi^T \alpha_{t-1}))$ 
8 return  $\alpha$ ,  $\sum_t \log Z_t$ 
```

3.3.2 Smoothing

A smoother infers the belief state at a given state based on all the observations or evidence $p(z_t | \mathbf{x}_{1:T})$. Although noise and uncertainty are significantly reduced as a result of conditioning on past and future data, the smoothing process can only be run offline.

³(Bishop 2006, Jordan (2003)) have more detail if we want to enlarge on potential numerical issues

Inference can be done by means of the forwards-backwards algorithm, also known as the Baum-Welch algorithm (Leonard E Baum 1972). Let $\gamma_t(j)$ be the desired smoothed posterior marginal,

$$\gamma_t(j) \triangleq p(z_t = j | \mathbf{x}_{1:T}),$$

$\alpha_t(j)$ be the filtered belief state at the step t as defined by equation ?? and $\beta_t(j)$ be the conditional likelihood of future evidence given that the hidden state at step t is j ,

$$\beta_t(j) \triangleq p(\mathbf{x}_{t+1:T} | z_t = j).$$

Then, the chain of smoothed marginals can be segregated into the past and the future components by conditioning on the belief state z_t ,

$$\gamma_t(j) = p(z_t = j | \mathbf{x}_{1:T}) \propto p(z_t = j, \mathbf{x}_{t+1:T} | \mathbf{x}_{1:t}) \propto p(z_t = j | \mathbf{x}_{1:t}) p(\mathbf{x}_{t+1:T} | z_t = j) \propto \alpha_t(j) \beta_t(j).$$

The future component can be computed recursively from right to left:

$$\begin{aligned} \beta_{t-1}(i) &= p(\mathbf{x}_{t:T} | z_{t-1} = i) \\ &= \sum_j p(z_t = j, \mathbf{x}_t, \mathbf{x}_{t+1:T} | z_{t-1} = i) \\ &= \sum_j p(\mathbf{x}_{t+1:T} | z_t = j) p(z_t = j, \mathbf{x}_t | z_{t-1} = i) \\ &= \sum_j p(\mathbf{x}_{t+1:T} | z_t = j) p(\mathbf{x}_t | z_t = j) p(z_t = j | z_{t-1} = i) \\ &= \sum_j \beta_t(j) \psi_t(j) \psi(i, j) \end{aligned}$$

Let β_t be a K -sized vector with the conditional likelihood of future evidence given the hidden state at step t . Then, the backwards procedure can be expressed in matrix notation as

$$\beta_{t-1} \propto \Psi(\psi_t \odot \beta_t).$$

At the last step, the base case is given by

$$\beta_T(i) = p(\mathbf{x}_{T+1:T} | z_T = i) = p(\emptyset | z_T = i) = 1.$$

Intuitively, the forwards-backwards algorithm passes information from left to right and then from right to left, combining them at each node. A straightforward implementation of the algorithm runs in $O(K^2T)$ time because of the $K \times K$ matrix multiplication at each step. There is a significant reduction if the transition matrix is sparse, for example a left-to-right transition matrix runs in $O(TK)$ time. Additional assumptions about the form of the transition matrix may ease the complexity further, for example reducing the time to $O(TK \log K)$ if $\psi(i, j) \propto \exp(-\sigma^2 |z_i - z_j|)$.

3.3.3 Fixed lag smoothing

A compromise between filtering and smoothing, it infers the belief state at a given point based on the information available up to that moment plus a fixed lag ℓ , that is $p(z_{t-\ell} | \mathbf{x}_{1:t})$. This approach yields better performance than filtering at the price of a delay, whose size can be tuned to trade off accuracy versus delay.

3.3.4 Two-slice marginal

I think this quantity is too closely related to the EM algo, which we won't be using I guess since we're going for a full bayes approach.

3.3.5 Backwards sampling

The smoothed posterior distribution of the hidden states is given by $\mathbf{z}_{1:T}^s \sim p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})$. While smoothing computes the sequence of the marginal distributions, additional information can be gathered by sampling from the posterior.

A naive sampling approach starts with the execution of the forwards-backwards algorithm to compute the two-slice smoothed marginal probabilities $p(z_{t-1,t}|\mathbf{x}_{1:T})$, continues with the computation of the conditionals $p(z_t|z_{t-1}, \mathbf{x}_{1:T})$ by normalizing, samples from the initial pair of states $z_{1,2}^* \sim p(z_{1,2}|\mathbf{x}_{1:T})$ and finally recursively samples the quantity of interest $z_t^* \sim p(z_t|z_{t-1}^*, \mathbf{x}_{1:T})$. This solutions requires a forwards-backwards pass as well as a forwards sampling pass.

Alternatively, it is possible to run a forward pass and perform sampling in the backwards pass. The joint posterior distribution can be written from right to left,

$$p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) = p(z_t|\mathbf{x}_{1:T}) = \prod_{t=T-1}^1 p(z_t|z_{t+1}, \mathbf{x}_{1:T}).$$

The state at a given point z_t can be sampled given future states,

$$z_t^s \sim p(z_t|z_{t+1:T}, \mathbf{x}_{1:T}) = p(z_t|z_{t+1}, \mathbf{x}_{1:t}) = p(z_t|z_{t+1}^s, \mathbf{x}_{1:t}),$$

where the sampling distribution is given by

$$p(z_t = i|z_{t+1} = j, \mathbf{x}_{1:t}) = p(z_t|z_{t+1}, \mathbf{x}_{1:t}) = \dots$$

At the last point, the base case is given by

$$z_T^s \sim p(z_T = i|\mathbf{x}_{1:T}) = \alpha_T(i).$$

The forwards filtering, backwards sampling algorithm forms the basis of blocked-Gibbs sampling methods for parameter inference.

3.3.6 Maximum a posteriori estimation

It is also of interest to compute the most probable state sequence,

$$\mathbf{z}^* = \arg \max_{\mathbf{z}_{1:T}} p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}).$$

The jointly most probable sequence of states can be inferred by means of MAP estimation. It is not necessarily the same as the sequence of marginally most probable states given by the maximizer of the posterior marginals (MPM),

$$\hat{\mathbf{z}} = (\arg \max_{z_1} p(z_1|\mathbf{x}_{1:T}), \dots, \arg \max_{z_T} p(z_T|\mathbf{x}_{1:T})),$$

which maximizes the expected number of correct individual states.

The MAP estimate is always globally consistent: while locally a state may be most probable at a given point, the Viterbi or max-sum algorithm decodes the most likely single plausible path Viterbi (1967). Furthermore, the MPM sequence may have zero probability, if it includes two successive states that are individually the most probable and yet their transition matrix element connecting them is zero. On the other hand, MPM is more robust since each node is estimated by averaging over its neighbours, rather than conditioning on a specific value of them.

The Viterbi algorithm is an adaptation of the forwards-backwards algorithm where the forward pass becomes a max-product and the backwards pass relies on a traceback procedure to recover the most probable path. Let $\delta_t(j)$ be the probability of arriving to the state j at step t given the most probable path was taken,

$$\delta_t(j) \triangleq \max_{z_1, \dots, z_{t-1}} p(\mathbf{z}_{1:t-1}, z_t = j | \mathbf{x}_{1:t}),$$

The most probable path to state j at step t consists of the most probable path to some other state i at point $t-1$, followed by a transition from i to j ,

$$\delta_t(j) = \max_i \delta_{t-1}(i) \psi(i, j) \phi_t(j),$$

where $\phi_t(j) = p(\mathbf{x}_t | z_t = j)$ is the ... ?

Additionally, the most likely previous state on the most probable path to j at step t is given by

$$a_t(j) = \arg \max_i \delta_{t-1}(i) \psi(i, j) \phi_t(j).$$

By initializing with $\delta_1 = \pi_j \phi_1(j)$ and terminating with the most probable final state $z_T^* = \arg \max_i \delta_T(i)$, the most probable sequence of states can be estimating using traceback

$$z_t^* = a_{t+1}(z_{t+1}^*).$$

It is advisable to work in the log domain to avoid numerical underflow,

$$\delta_t(j) \triangleq \max_{\mathbf{z}_{1:t-1}} \log p(\mathbf{z}_{1:t-1}, z_t = j | \mathbf{x}_{1:t}) = \max_i \log \delta_{t-1}(i) + \log \psi(i, j) + \log \phi_t(j).$$

As with the backwards-forwards algorithm, the time complexity of Viterbi is $O(K^2T)$ and the space complexity is $O(KT)$. If the transition matrix has the form $\psi(i, j) \propto \exp(-\sigma^2 \|\mathbf{z}_i - \mathbf{z}_j\|^2)$, implementation runs in $O(TK)$ time.

3.3.7 Prediction

Inference about the future belief states given the past observations requires computing $p(z_{t+h} | \mathbf{x}_{1:t})$ for the prediction horizon $h > 0$. The process is straightforward, the transition matrix is raised to the power of the prediction horizon and applied to the current belief state.

$$p(z_{t+h} | \mathbf{x}_{1:t}) = \boldsymbol{\alpha}_t \mathbf{A}^h.$$

WARNING, NOT SURE I'M GENERALIZING EQ. 17.42 FROM Murphy (2012) CORRECTLY.

Prediction about future observations involves the posterior predictive density,

$$p(\mathbf{x}_{t+h} | \mathbf{x}_{1:t}) = \sum_{z_{t+h}} p(\mathbf{x}_{t+h} | z_{t+h}) p(z_{t+h} | \mathbf{x}_{1:t}) = \sum_{z_{t+h}} p(\mathbf{x}_{t+h} | z_{t+h}) = \boldsymbol{\alpha}_t \mathbf{A}^h.$$

Since the influence of all available observations $\mathbf{x}_{1:t}$ is summarised in the K -sized vector $\boldsymbol{\alpha}_t$, prediction can be carried forward indefinitely with only a fixed amount of storage.

3.4 Parameter estimation

The parameters of the models are $\boldsymbol{\theta} = (\boldsymbol{\pi}_1, \mathbf{A}, \mathbf{B})$, where $\boldsymbol{\pi}_1$ is the initial state distribution, \mathbf{A} is the transition matrix and \mathbf{B} are the parameters of the state-conditional density function $p(\mathbf{x}_t | z_t = j)$. The form of \mathbf{B} depends on the specification of the observation model. Discrete observations may be characterized with an L -sized multinoulli distribution with parameters $B_{jl} = p(x_t = l, z_t = j)$ where $l \in \{1, \dots, L\}$, while continuous emissions may be modelled with a Gaussian distribution with parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ where $k \in \{1, \dots, K\}$.

Estimation can be run under both the maximum likelihood and bayesian frameworks. The former may be easily extended to regularized maximum likelihood with the introduction of the corresponding priors over the parameters. Although it is a straightforward procedure when the data is fully observed, in practice the latent states $\mathbf{z}_{1:T}$ are hidden. In principle, this is just another optimization problem that can be solved via standard numerical optimization methods. Analogous to fitting a mixture model, the most common approach is the application of the Expectation-Maximization (EM) algorithm Dempster, Laird, and Rubin (1977) to find either the maximum likelihood or the maximum a posteriori estimates. In the context of HMM, this is also known as the Baum-Welch algorithm Leonard E. Baum et al. (1970).

The algorithm considers the unobserved internal state trajectory as missing data and decouples the learning problem into two parts: a temporal assignment subproblem and a statistical learning subproblem, which consists of fitting parameters to the next-state and output mapping defined by the estimated trajectory.

Should I develop the EM algo? It's trivial and it's on any book. I've already cited the original source.

Additionally, estimation can be done in a fully bayesian fashion. In terms of variational Bayes EM, MacKay (1997) proposes a method based on the optimization of an ensemble which approximates the entire posterior probability distribution. In turns, Beal (2003) presents a unified variational Bayesian framework which approximates these computations in models with latent variables using a lower bound on the marginal likelihood. As for Markov Chain Monte Carlo methods, block Gibbs sampling can be applied as shown in Frühwirth-Schnatter (2006). Briefly, samples are drawn from the density $p(\mathbf{z}_{1:T} | \mathbf{x}_{1:T}, \boldsymbol{\theta})$ by means of forwards-filtering, backwards-sampling, and then the parameters are sampled from their posteriors conditional on the sampled latent paths.

3.4.1 Implementation in Stan

While we're implementing the code during the following months, we'll probably have to work out a few implementation details like marginalization, vectorization and other speed-up tricks, limitations by stan data structures, priors and tricks to improve convergence, etc. These adaptations for the inference and/or estimation steps should be noted and explained in here. This would be a great contribution, since there's little written about that apart from a few pages in stan man.

3.4.2 Variations

There are numerous variants of the HMM. Some of them impose constraints on the form of the transition matrix.

In the left-to-right or Barkis model, where $A_{ij} = 0 \ \forall \ j < i$, the underlying state sequence stays the same or increases as time increases.

4 Input-Output Hidden Markov Models

The Input-Output Hidden Markov Model (IOHMM) is an architecture proposed by Bengio and Frasconi (1995) to map input sequences, sometimes called the control signal, to output sequences. Similarly to HMM, the model is meant to be especially effective when it comes to learning long term memory, that is when input-output sequences span long points. On the other hand, it differs from HMM, which is part of the unsupervised learning paradigm, since it is capable of learning the output sequence itself instead of just the output sequence distribution. IOHMM is a probabilistic framework that can deal with general sequence processing tasks such as production, classification, or prediction.

4.1 Model specification

As with HMM, IOHMM involves two interconnected models.

$$\begin{aligned} z_t &= f(z_{t-1}, \mathbf{u}_t) \\ \mathbf{x}_t &= g(z_t, \mathbf{u}_t) \end{aligned}$$

The first line corresponds to the state model, which consists of a discrete-time, discrete-state hidden states $z_t \in \{1, \dots, K\}$ whose transition depends on the previous state z_{t-1} and the input vector $\mathbf{u}_t \in \mathbb{R}^M$. Additionally, the observation model is governed by $g(z_t, \mathbf{u}_t)$, where $\mathbf{x}_t \in \mathbb{R}^R$ is the vector of observations, emissions or output. The corresponding joint distribution,

$$p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T} | \mathbf{u}_t),$$

can take many forms. In the most frequent parametrization when all the inputs and the outputs are continuous, the state model involves a logistic regression whose parameters depend on the previous hidden states

$$p(z_t | \mathbf{x}_t, \mathbf{u}_t, z_{t-1} = i) = \text{logit}^{-1}(\mathbf{w}_i \mathbf{u}_t),$$

and the observation model is built upon the Gaussian density with parameters depending on the current state

$$p(\mathbf{x}_t | \mathbf{u}_t, z_t = j) = \mathcal{N}(\mathbf{x}_t | \mathbf{b}_j \mathbf{u}_t, \mathbf{\Sigma}_j).$$

IOHMM is fully probabilistic. Hidden states are assumed to follow a multinomial distribution that depends on the input sequence. The transition probabilities $\Psi_t(i, j) = p(z_t = j | z_{t-1} = i, \mathbf{u}_{1:t})$, which govern the state dynamics, are driven by the control signal as well.

As for the output sequence, the local evidence at time t now becomes $\psi_t(j) = p(\mathbf{x}_t | z_t = j, \boldsymbol{\eta}_t)$, where $\boldsymbol{\eta}_t = \mathbb{E} \langle \mathbf{x}_t | z_t, \mathbf{u}_t \rangle$ can be interpreted as the expected location parameter for the probability distribution of the emission \mathbf{x}_t , conditional on the input vector \mathbf{u}_t and the hidden state z_t .

The actual form of the emission density $f_{\mathbf{x}}(\mathbf{x}_t, \boldsymbol{\eta}_t)$ can be discrete or continuous. In case of sequence classification or symbolic mutually exclusive emissions, it is possible to set up the multinomial distribution by running the softmax function over the estimated outputs of all possible states. Alternatively, when approximating continuous observations with the Gaussian density, the target is estimated as a linear combination of these outputs.

4.2 Inference

4.2.1 Filtering

Filtered marginals can be computed recursively by adjusting the forward algorithm to consider the input sequence:

$$\begin{aligned}
\alpha_t(j) &\triangleq p(z_t = j | \mathbf{x}_{1:t}, \mathbf{u}_{1:t}) \\
&= \psi_t(j) \sum_{i=1}^K p(z_t = j | z_{t-1} = i, \mathbf{u}_t) p(z_{t-1} = i | \mathbf{u}_{1:t-1}) \\
&= \psi_t(j) \sum_{i=1}^K \Psi_t(i, j) p(z_{t-1} = i | \mathbf{u}_{1:t-1}) \\
&= \psi_t(j) \sum_{i=1}^K \Psi_t(i, j) \alpha_{t-1}(i).
\end{aligned}$$

4.2.2 Smoothing

Similarly, inference about the smoothed posterior marginal can be computed adjusting the forwards-backwards algorithm to consider the input sequence in both components $\alpha_t(j)$ and $\beta_t(j)$. The future component now becomes

$$\begin{aligned}
\beta_{t-1}(j) &\triangleq p(\mathbf{x}_{t:T} | z_{t-1} = j, \mathbf{u}_{t:T}) \\
&= \psi_t(j) \sum_{i=1}^K \Psi_t(i, j) \beta_t(i).
\end{aligned}$$

4.3 Parameter estimation

The parameters of the models are $\boldsymbol{\theta} = (\boldsymbol{\pi}_1, \boldsymbol{\theta}_h, \boldsymbol{\theta}_o)$, where $\boldsymbol{\pi}_1$ is the initial state distribution, $\boldsymbol{\theta}_h$ are the parameters of the hidden model and $\boldsymbol{\theta}_o$ are the parameters of the state-conditional density function $p(\mathbf{x}_t | z_t = j, \mathbf{u}_t)$. The form of $\boldsymbol{\theta}_h$ and $\boldsymbol{\theta}_o$ depend on the specification of the model. State transition may be characterized by a logistic regression with parameters \mathbf{w}_k for $k \in \{1, \dots, K\}$, while emissions may be modelled with a linear regression with Gaussian noise \mathbf{B}_k and $\boldsymbol{\Sigma}_k$ for $k \in \{1, \dots, K\}$.

Estimation can be run under both the maximum likelihood and bayesian frameworks. Although it is a straightforward procedure when the data is fully observed, in practice the latent states $\mathbf{z}_{1:T}$ are hidden. The most common approach is the application of the EM algorithm to find either the maximum likelihood or the maximum a posteriori estimates. The application of sigmoidal functions, for example the logistic or softmax transforms for the hidden transition model, requires numeric optimization via gradient ascent or similar methods for the M step.

References

- Baum, Leonard E. 1972. “An Inequality and Associated Maximaization Technique in Statistical Estimation for Probabilistic Functions of Markov Process.” *Inequalities* 3: 1–8.
- Baum, Leonard E., and J. A. Eagon. 1967. “An Inequality with Applications to Statistical Estimation

- for Probabilistic Functions of Markov Processes and to a Model for Ecology.” *Bulletin of the American Mathematical Society* 73 (3). American Mathematical Society (AMS): 360–64. doi:10.1090/s0002-9904-1967-11751-8.
- Baum, Leonard E., Ted Petrie, George Soules, and Norman Weiss. 1970. “A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains.” *The Annals of Mathematical Statistics* 41 (1). Institute of Mathematical Statistics: 164–71. doi:10.1214/aoms/1177697196.
- Beal, Matthew J. 2003. “Variational Algorithms for Approximate Bayesian Inference.”
- Bengio, Yoshua, and Paolo Frasconi. 1995. “An Input Output Hmm Architecture.”
- Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.
- Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. “Maximum Likelihood from Incomplete Data via the Em Algorithm.”
- Doob, Joseph L. 1953. *Stochastic Processes*. Vol. 7. 2. Wiley New York.
- Frühwirth-Schnatter, Sylvia. 2006. *Finite Mixture and Markov Switching Models*. Springer New York. doi:10.1007/978-0-387-35768-3.
- Jordan, Michael I. 2003. “An Introduction to Probabilistic Graphical Models.” preparation.
- MacKay, David J. C. 1997. “Ensemble Learning for Hidden Markov Models.”
- Rabiner, Lawrence R. 1990. “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.” In *Readings in Speech Recognition*, 267–96. Elsevier. doi:10.1016/b978-0-08-051584-7.50027-9.
- Viterbi, A. 1967. “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm.” *IEEE Transactions on Information Theory* 13 (2). Institute of Electrical; Electronics Engineers (IEEE): 260–69. doi:10.1109/tit.1967.1054010.