Inertial Hidden Markov Models: Modeling Behavior Change in Multivariate Time Series

George D. Montañez
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA USA
gmontane@cs.cmu.edu

Saeed Amizadeh
Yahoo Labs
Sunnyvale, CA USA
amizadeh@yahoo-inc.com

Nikolay Laptev
Yahoo Labs
Sunnyvale, CA USA
nlaptev@yahoo-inc.com

Abstract

Faced with the problem of characterizing systematic changes in multivariate time series in an unsupervised manner, we derive and test two methods of regularizing hidden Markov models for this task. Regularization on state transitions provide smooth transitioning among states, such that the sequences are split into broad, contiguous segments. Our methods are compared with a recent hierarchical Dirichlet process hidden Markov model (HDP-HMM) and a baseline standard hidden Markov model, of which the former suffers from poor performance on moderate-dimensional data and sensitivity to parameter settings, while the latter suffers from rapid state transitioning, over-segmentation and poor performance on a segmentation task involving human activity accelerometer data from the UCI Repository. The regularized methods developed here are able to perfectly characterize change of behavior the human activity data in roughly half of the real-data test cases, with accuracy of 94% and low variation of information. In contrast to the HDP-HMM, our methods provide simple, drop-in replacements for standard hidden Markov model update rules, allowing standard expectation maximization (EM) algorithms to be used for learning.

Introduction

"Some seek complex solutions to simple problems; it is better to find simple solutions to complex problems." - Soramichi Akiyama

Time series data arise in different areas of science and technology, describing the *behavior* of both natural and man-made systems over time. These behaviors are often quite complex with uncertainty, which in turn require us to incorporate sophisticated dynamics and stochastic models to model them. Furthermore, these complex behaviors can *change* over time due to some external event and/or some internal systematic change of dynamics/distribution. For example, consider the case of monitoring one's physical activity via an array of accelerometer body sensors over time. A certain pattern emerges on the time series of the sensors' readings while the person is walking; however, this pattern quickly changes to a new one as the person starts running. From the data anaylsis perspective, firstly it is important to detect these *change points* as they are quite often indicative of an "interesting" event or an anomaly in the system. Secondly, we are also interested to characterize the new *state* of the system (e.g. running vs. walking) which reflects its modus operandi. Change point detection methods () have been proposed to answer the first question while the classical Hidden Markov Models (HMM) can answer both.

One crucial observation in many real-world systems (natural and man-made), however, is that the behavior changes are typically infrequent; that is, the system takes some (unknown) time before it changes its behavior to a new modus operandi. For instance, in our earlier example, it is unlikely that a person changes between walking and running very frequently, making the durations of different activities over time relatively long and highly variable. We refer to this as the *inertial property*, alluding to the physical property of matter that ensures it will continue along a fixed course unless acted upon by an external force. Unfortunately, the classical HMMs are not equipped with sufficient mechanisms to capture this property and quite often result in a high rate of state transitioning and subsequently false positives in terms of detecting change points.

There are very few solutions in the literature to address this problem. In the context of Markov models, Fox *et al.* (Fox et al. 2011) have recently proposed the *sticky hierarchical Dirichlet process hidden Markov model (HDP-HMM)* which uses a Bayesian non-parametric approach with appropriate priors to promote self-transitioning (or *stickiness*) for HMMs. Albeit its neat theoretical foundations, HDP-HMM is not a practical solution in many real-world situations. In particular, the performance of HDP-HMM tends to break down as the dimentionality of the problem goes beyond 10. Moreover, due to iterative Gibbs sampling for its learning, HDP-HMM is computationally prohibitive. But, the most significant downside of HDP-HMM in practice originates from its non-parametric Bayesian nature: due to the existence of many hyperparameters, the search space for initial tuning is exponentially large which significantly affects the learning quality for a given task.

In this paper, we propose a regularization-based framework for HMMs called *Inertial HMM* to bias them toward the inertial property. Similar to HDP-HMM, our framework is based on theoretically sound foundations, yet much simpler and more intuitive than HDP-HMM. In particular, our framework has only two initial parameters for which we

have developed intuitive initilization techniques that significantly minimizes the effort needed for parameter tuning. Furthermore, as we show later, in practice, our proposed methodology boils down to upgraded update rules for standard HMMs. The main practical implication of this observation is that using our methodology, the standard HMM packages can be simply upgraded to support the inertial property yet preserve the computational efficiency of the standard HMM approach. By performing rigouros experiments on both synthetic and moderate dimensional real datasets, we show that not only are Inertial HMMs much faster than HDP-HMM, but the quality of detection is siginificantly better than that of the HDP-HMM, and therefore proposing that Inertial HMMs are far more practical choices compared to the state-of-the-art.

Problem Statement

Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ denote a d-dimensional multivariate time series, where $\mathbf{x}_t \in \mathbb{R}^d$. Given such a time series, we seek to segment $\mathbf{X}$ along the time axis into *segments*, where each segment corresponds to a subsequence $\mathbf{X}_{i\ldots i+m} = \{\mathbf{x}_i, \ldots, \mathbf{x}_{i+m}\}$ and maps to a predictive (latent) state $\mathbf{z}$, represented as a one-of-K vector, where $|\mathbf{z}| = K$ and $\sum_{i=1}^{K} z_{t,i} = 1$. For simplicity of notation, let $\mathbf{z}_t = k$ denote $z_{t,k} = 1$ and let $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ denote the sequence of latent states. Then for all $\mathbf{x}_t$ mapping to state k, we require that

$$\begin{aligned}&\Pr(\mathbf{x}_{t+1}|\mathbf{X}_{1\ldots t}, \mathbf{z}_t = k) = \Pr(\mathbf{x}_{t+1}|\mathbf{z}_t = k)\\ &= \Pr(\mathbf{x}_{t'+1}|\mathbf{z}_{t'} = k) = \Pr(\mathbf{x}_{t'+1}|\mathbf{X}_{1\ldots t'}, \mathbf{z}_{t'} = k).\end{aligned}$$

Thus, the conditional distribution over futures at time t conditioned on being in state k is equal to the distribution over futures at time t' conditioned on being in the same state. Thus, we assume conditional independence given state, and stationarity of the generative process.

We impose two additional complexity criteria on our model. First, we seek models with a small number of latent states, $K \ll T$, and second, we desire state transition sequences of low complexity such that the transitioning of states does not occur too rapidly. We refer to this as the *inertial transition* requirement, alluding to the physical property of matter that ensures it will continue along a fixed course unless acted upon by an external force.

The above desiderata must be externally imposed on our model, since simply maximizing the likelihood of the data will result in $K = T$ (i.e., each sample corresponds a unique state/distribution), and in general we may have rapid transitions among states. For the first desideratum, we choose the number of states in advance as is typically done for hidden Markov models (Rabiner 1989). For the second, we directly alter the probabilistic form of our model to include a parameterized regularization that reduces the likelihood of transitioning between different latent states.

Problem Input

As stated above, we are given a single d-dimensional multivariate time series of T time samples. Alternatively, the single time series can be thought of as a collection of d one-dimensional time series. As the generative story, we assume that at each time step t a state $\mathbf{z}_t$ is chosen, given the previous state $\mathbf{z}_{t-1}$, according to the transition probabilities governing states. A d-dimensional point-sample is then drawn according to the emission density for state $\mathbf{z}_t$, and the process repeats for $1 < t \leq T$.

Problem Output

The output of the process is a list of integer tuples (t, k), where $1 \leq t \leq T$ denotes the ending time of the segment and $1 \leq k \leq K$ the state that occurs during that segment.

Inertial Hidden Markov Models

Hidden Markov models (HMMs) are a class of long-studied probabilistic models well-suited for sequential data (Rabiner 1989). As a starting point for developing our inertial HMMs, we begin a standard K-state HMM with Gaussian emission densities. HMMs (locally) maximize the likelihood of the data, but typically do not guarantee slow inertial transitioning among states. The number of states must be specified in advance, but no other parameters need to be given, as the remaining parameters are all estimated directly from the data.

To accommodate the inertial transition requirement, we derive two different methods of enforcing state-persistence in HMMs. Both methods alter the probabilistic form of the complete data joint likelihood, which result in altered transition matrix update equations. The resulting update equations share a related mathematical structure and, as is shown in Section , have similar performance in practice.

We will next describe both methods and provide outlines of their derivations, with more detail being given to the derivation for the second method.

Maximum A Posteriori (MAP) Regularized HMM

Following (Gauvain and Lee 1994), we alter the standard HMM to include a Dirichlet prior on the transition probability matrix, such that transitions out-of-state are penalized by some regularization factor. A Dirichlet prior on the transition matrix $\mathbf{A}$, for the jth row, has the form

$$p(A_j; \eta) \propto \prod_{i=1}^{K} A_{jk}^{\eta_{jk}-1}$$

where the η_{jk} are free parameters and A_{jk} is the transition probability from state j to state k. The posterior joint density over $\mathbf{X}$ and $\mathbf{Z}$ becomes

$$P(\mathbf{X}, \mathbf{Z}; \theta, \eta) \propto \left[\prod_{i=1}^{K}\prod_{i=1}^{K} A_{jk}^{\eta_{jk}-1}\right] P(\mathbf{X}, \mathbf{Z} \mid \mathbf{A}; \theta)$$

and the log-likelihood is

$$\begin{aligned}\ell(\mathbf{X}, \mathbf{Z}; \theta, \eta) \propto &\sum_{i=1}^{K}\sum_{i=1}^{K}(\eta_{jk} - 1)\log A_{jk} + \log P(\mathbf{z}_1; \theta)\\ &+ \sum_{t=1}^{T}\log P(\mathbf{x}_t|\mathbf{z}_t; \theta) + \sum_{t=2}^{T}\log P(\mathbf{z}_t|\mathbf{z}_{t-1}; \theta).\end{aligned}$$

MAP estimation is then used in the M-step of the EM algorithm, to update the transition probability matrix. Maximizing, with appropriate Lagrange multiplier constraints, we obtain the update equation for the transition matrix,

$$A_{jk} = \frac{(\eta_{jk} - 1) + \sum_{t=2}^{T} \xi(z_{(t-1)j}, z_{tk})}{\sum_{i=1}^{K}(\eta_{ji} - 1) + \sum_{i=1}^{K}\sum_{t=2}^{T} \xi(z_{(t-1)j}, z_{ti})}. \quad (1)$$

where the $\xi(z_{(t-1)j}, z_{tk}) = \mathbb{E}[z_{(t-1)j} z_{tk}]$.

Given our prior, we can control the probability of self-transitions among states, but this method requires that we choose a set of K^2 parameters for the Dirichlet prior. However, since we are solely concerned about increasing the probability of self-transitions, we can reduce these parameters to a single parameter λ governing the amplification of self-transitions. We therefore define $\eta_{jk} = 1$ when $j \neq k$ and $\eta_{kk} = \lambda \geq 1$ otherwise, and the transition update equation becomes

$$A_{jk} = \frac{(\lambda - 1)\mathbb{1}(j = k) + \sum_{t=2}^{T} \xi(z_{(t-1)j}, z_{tk})}{(\lambda - 1) + \sum_{i=1}^{K}\sum_{t=2}^{T} \xi(z_{(t-1)j}, z_{ti})} \quad (2)$$

where $\mathbb{1}(\cdot)$ denotes the indicator function.

Inertial Regularization via Pseudo-observations

We now derive a second method of regularizing the state transitions, where we alter the HMM likelihood function to include a latent binary random variable, V indicating that a self-transition was chosen at random from among all transitions, according to some distribution. Thus, we view the transitions as being partitioned into two sets, self-transitions and non-self-transitions, and we draw a member of the self-transition set according to a Bernoulli distribution governed by parameter p. Given a latent state sequence $\mathbf{Z}$, with transitions chosen according to transition matrix $\mathbf{A}$, we define p as a function of both $\mathbf{Z}$ and $\mathbf{A}$. We would like p to have two properties: 1) it should increase with increasing $\sum_k A_{kk}$ (probability of self-transitions) and 2) it should increase as the number of self-transitions in $\mathbf{Z}$ increases. This will allow us to encourage self-transitions as a simple consequence of maximizing the likelihood of our observations.

We begin with a version of p based on a penalization constant $0 < \epsilon < 1$ that scales appropriately with the number of self-transitions. If we raise ϵ to a large positive power, the resulting p will decrease. Thus, we define p as ϵ raised to the number of non-self-transitions, B, in the state transition sequence, so that the probability of selecting a self-transition increases as B decreases. Using the fact that $B = (T-1) - \sum_{t=2}^{T}\sum_{k=1}^{K} z_{(t-1)k} z_{tk}$, we obtain

$$\begin{aligned} p = \epsilon^B &= \epsilon^{\sum_{t=2}^{T} 1 - \sum_{t=2}^{T}\sum_{k=1}^{K} z_{(t-1)k} z_{tk}} \\ &= \epsilon^{\sum_{t=2}^{T}\sum_{k=1}^{K} z_{(t-1)k} - \sum_{t=2}^{T}\sum_{k=1}^{K} z_{(t-1)k} z_{tk}} \\ &= \prod_{t=2}^{T}\prod_{k=1}^{K} \epsilon^{z_{(t-1)k} - z_{(t-1)k} z_{tk}}. \end{aligned} \quad (3)$$

Since ϵ is arbitrary, we choose $\epsilon = A_{kk}$, to allow p to scale appropriately with increasing probability of self-transition. We therefore arrive at

$$p = \prod_{t=2}^{T}\prod_{k=1}^{K} A_{kk}^{z_{(t-1)k} - z_{(t-1)k} z_{tk}}.$$

Thus, we define p as a computable function of $\mathbf{Z}$ and $\mathbf{A}$. Defining p in this deterministic manner is equivalent to choosing the parameter value from a degenerate probability distribution that places a single point mass at the value computed, allowing us to easily obtain a posterior distribution on V. Furthermore, we see that the function increases as the number of self-transitions increases, since $A_{kk} \leq 1$ for all k, and p will generally increase as $\sum_k A_{kk}$ increases. Thus, we obtain a parameter $p \in (0, 1]$ that satisfies all our desiderata. With p in hand, we say that V is drawn according to the Bernoulli distribution, Bern(p), and we observe $V = 1$ (i.e., a member of the self-transition set was chosen). To gain greater control over the strength of regularization, let λ be a positive integer and $\mathbf{V}$ be an λ-length sequence of pseudo-observations, drawn i.i.d. according to Bern(p). Since $P(V = 1|\mathbf{Z}; \mathbf{A}) = p$, we have

$$P(\mathbf{V} = \mathbf{1}|\mathbf{Z}; \mathbf{A}) = \left[\prod_{t=2}^{T}\prod_{k=1}^{K} A_{kk}^{z_{(t-1)k} - z_{(t-1)k} z_{tk}}\right]^{\lambda}$$

where $\mathbf{1}$ denotes the all-ones sequence of length λ.

Noting that $\mathbf{V}$ is conditionally independent of $\mathbf{X}$ given the latent state sequence $\mathbf{Z}$, we now consider the joint log-density over $\mathbf{X}$, $\mathbf{V}$, and $\mathbf{Z}$ parameterized by $\theta = \{\pi, \mathbf{A}, \phi\}$, which are the start-state probabilities, state transition matrix and emission parameters, respectively.

$$\begin{aligned} \ell(\mathbf{X}, \mathbf{V}, \mathbf{Z}; \theta) &= \log P(\mathbf{V}|\mathbf{Z}; \theta) + \log P(\mathbf{z}_1; \theta) \\ &\quad + \sum_{t=1}^{T} \log P(\mathbf{x}_t|\mathbf{z}_t; \theta) + \sum_{t=2}^{T} \log P(\mathbf{z}_t|\mathbf{z}_{t-1}; \theta). \end{aligned}$$

By defining $\ell_1 := \ell(\mathbf{X}, \mathbf{V} = \mathbf{1}, \mathbf{Z}; \theta)$ and noting that $P(\mathbf{z}_1; \theta) = \prod_{k=1}^{K} \pi_k^{z_{1k}}$, $P(\mathbf{x}_t|\mathbf{z}_t; \theta) = \prod_{k=1}^{K} P(\mathbf{x}_t; \phi_k)^{z_{tk}}$, and $P(\mathbf{z}_t|\mathbf{z}_{t-1}; \theta) = \prod_{k=1}^{K}\prod_{j=1}^{K} A_{jk}^{z_{(t-1)j} z_{tk}}$, we obtain

$$\begin{aligned} \ell_1 &= \sum_{t=2}^{T}\sum_{k=1}^{K} \lambda[z_{(t-1)k} - z_{(t-1)k} z_{tk}] \log A_{kk} \\ &\quad + \sum_{k=1}^{K} z_{1k} \log \pi_k + \sum_{t=1}^{T}\sum_{k=1}^{K} z_{tk} \log P(\mathbf{x}_t; \phi_k) \\ &\quad + \sum_{t=2}^{T}\sum_{k=1}^{K}\sum_{j=1}^{K} [z_{(t-1)j} z_{tk}] \log A_{jk}. \end{aligned}$$

Following Bishop (Bishop 2007), we define

$$\gamma(z_{tk}) = \mathbb{E}[z_{tk}] = \sum_{\mathbf{z}} \gamma(\mathbf{z}) z_{tk}$$

$$\xi(z_{(t-1)j}, z_{tk}) = \mathbb{E}[z_{(t-1)j} z_{tk}] = \sum_{\mathbf{z}} \gamma(\mathbf{z}) z_{(t-1)j} z_{tk}$$

to obtain

$$\mathbb{E}_{\mathbf{Z}}[\ell_1] = \sum_{t=2}^{T}\sum_{k=1}^{K}\lambda[\gamma(z_{(t-1)k}) - \xi(z_{(t-1)k}, z_{tk})]\log A_{kk} + \sum_{k=1}^{K}\gamma(z_{1k})\log\pi_k + \sum_{t=1}^{T}\sum_{k=1}^{K}\gamma(z_{tk})\log P(\mathbf{x}_t;\phi_k) + \sum_{t=2}^{T}\sum_{k=1}^{K}\sum_{j=1}^{K}\xi(z_{(t-1)j}, z_{tk})\log A_{jk}. \quad (4)$$

Using Lagrange multipliers, taking the derivative of (4) with respect to A_{jk} and setting to the result to zero, we obtain the regularized maximum likelihood estimate for A_{jk}:

$$A_{jk} = \frac{B_{j,k,T} + \mathbb{1}(j=k)C_{j,k,T}}{\sum_{i=1}^{K} B_{j,i,T} + C_{j,j,T}} \quad (5)$$

where

$$B_{j,k,T} = \sum_{t=2}^{T}\xi(z_{(t-1)j}, z_{tk}),$$
$$C_{j,k,T} = \lambda\left[\sum_{t=2}^{T}[\gamma(z_{(t-1)k}) - \xi(z_{(t-1)j}, z_{tk})]\right] \quad (6)$$

and $\mathbb{1}(\cdot)$ denotes the indicator function. The forward-backward algorithm can then be used for efficient computation of the γ and ξ values, as in unregularized HMMs.

Ignoring normalization, we see that

$$A_{jk} \propto \begin{cases} B_{j,k,T} + C_{j,j,T} & \text{if } j = k \\ B_{j,k,T} & \text{otherwise.} \end{cases}$$

Examining the $C_{j,j,T}$ term, we see that λ is a multiplier of additional mass contributions for self-transitions, where the contributions are the difference between $\gamma(z_{(t-1)j})$ and $\xi(z_{(t-1)j}, z_{tj})$. These two quantities represent, respectively, the expectation of being in a state j at time $t-1$ and the expectation of remaining there in the next time step. The larger λ or the larger the difference between arriving at a state and remaining there, the greater the additional mass given to self-transition.

Scale-Free Regularization In Equation 2, the strength of the regularization diminishes with growing T, so that asymptotically the regularized estimate and unregularized estimate become equivalent. While this is often desirable in other contexts, maintaining a consistent strength of inertial regularization becomes important with time series of increasing length, as is the case with online learning methods. Figure shows a regularized segmentation of human accelerometer data (discussed later in the Experiments section), where the regularization is strong enough to correctly segment the series into large, contiguous sections. If we then increase the number of data points in each section by a factor of ten while keeping the same regularization parameter setting, we see that the regularization is no longer strong enough, as is shown in Figure . Two of the sections have become splintered into small, choppy regions. Thus, the λ parameter is sensitive to the size of the time series.

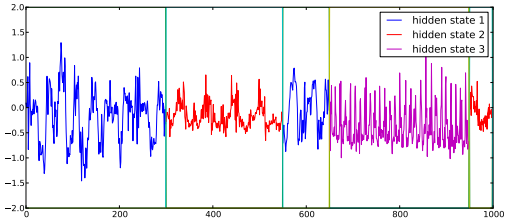

Figure 1: Human activities accelerometer data, short sequence. Vertical partitions correspond to changes of state.

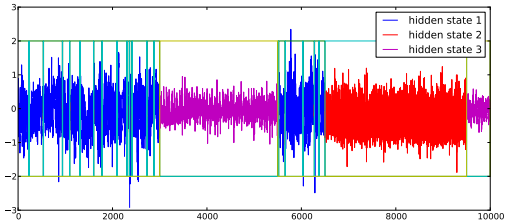

Figure 2: The long sequence human activities accelerometer data using regularization parameter from short sequence.

We desire models where the regularization strength is scale-free, having roughly the same strength regardless of how the time series grows. To achieve this, we define the λ parameter to scale with the number of transitions, namely $\lambda = (T-1)^{\zeta}$, and our scale-free update equation becomes

$$A_{jk} = \frac{((T-1)^{\zeta} - 1)\mathbb{1}(j=k) + \sum_{t=2}^{T}\xi(z_{(t-1)j}, z_{tk})}{((T-1)^{\zeta} - 1) + \sum_{i=1}^{K}\sum_{t=2}^{T}\xi(z_{(t-1)j}, z_{ti})}. \quad (7)$$

This preserves the effect of regularization as T increases, and ζ becomes our new regularization parameter, controlling the strength of the regularization. For consistency, we also re-parameterize Equation 6 using $\lambda = (T-1)^{\zeta}$.

Towards Parameter-Free Regularization The two methods of inertial transition regularization work well as long as the strength of regularization is provided. Here we seek to develop a version of the regularized HMM that does not require specification in advance of the regularization parameter. We accomplish this by making an assumption concerning the distribution of segment lengths. If we assume that most of the segment lengths are of roughly the same order-of-magnitude scale, then for a fixed K, we can automatically tune the regularization parameter.

We first define a range of possible regularization parameter values (such as $\lambda \in [0, 75]$), and perform a search on this interval for a value that gives *sufficient regularization*. 'Sufficient regularization is defined w.r.t the Gini ratio (Gini 1936; Wikipedia 2004), which is a measure of statistical dispersion often used to quantify income inequality. For a collection of observed segment lengths $L = \{l_1, \ldots, l_m\}$,

given in ascending order, the Gini ratio is estimated by

$$G(L) = 1 - \frac{2}{m-1}\left(m - \frac{\sum_{i=1}^{m} i l_i}{\sum_{i=1}^{m} l_i}\right).$$

Our assumption is that the true segmentation has a Gini ratio less than one-half, which corresponds to having more equality among segment lengths than not. One can perform a binary search on the search interval to find the smallest ζ parameter for which the Gini ratio is at least one-half. This increases the time complexity by a factor of $O(\log_2(R/\epsilon))$, where R is the range of the parameter space.

Online Learning of Inertial HMM parameters

Hidden Markov models traditionally use batch methods for learning model parameters, such as Baum-Welch EM. Since our inertial regularization methods rely on standard EM learning, we can naturally incorporate incremental EM learning techniques into our system. We thus extend the work of Stenger *et al.* (Stenger et al. 2001) to provide an online learning algorithm for our regularized MAP hidden Markov model, which allows scaling to arbitrarily large datasets. Theoretical justification for incremental online EM learning is given in (Neal and Hinton 1999).

Parameter Update Equations Define

$$D_{T,i} := ((T-1)^{\zeta} - 1) + \sum_{t=2}^{T}\sum_{k=1}^{K} \xi(z_{(t-1)i}, z_{tk}).$$

The recurrence for $D_{T,i}$ is then formulated as

$$D_{T,i} = D_{(T-1),i} + [(T-1)^{\zeta} - (T-2)^{\zeta}] + \sum_{k=1}^{K} \xi(z_{(T-1)i}, z_{Tk})$$

where T is the current time-step. Since T is both the current and final time-step, we have $\beta(z_{T,k}) = 1$ for $k = 1, \ldots, K$, and thus

$$\begin{aligned}\xi(\mathbf{z}_{t-1}, \mathbf{z}_t) &= P(\mathbf{z}_{t-1}, \mathbf{z}_t | \mathbf{X}) \\ &= \frac{\alpha(\mathbf{z}_{t-1}) p(\mathbf{x}_t | \mathbf{z}_t; \phi) p(\mathbf{z}_t | \mathbf{z}_{(t-1)}) \beta(\mathbf{z}_t)}{p(\mathbf{X})} \\ &= \frac{\alpha(z_{(t-1)i}) p(\mathbf{x}_t; \phi_j) A_{ij}^{(T-1)}}{\sum_{k=1}^{K} \alpha(z_{tk})}\end{aligned}$$

where

$$\alpha(z_{tj}) = \left[\sum_{i=1}^{K} \alpha(z_{(t-1)i}) A_{ij}^{(t-1)}\right] p(\mathbf{x}_t; \phi_j).$$

An efficient online update equation for the regularized transition matrix is then given by

$$\begin{aligned}A_{ij}^{(T)} = \frac{D_{(T-1),i}}{D_{T,i}} A_{ij}^{(T-1)} + \frac{\xi(z_{(T-1)i}, z_{Tj})}{D_{T,i}} \\ + \frac{\mathbb{1}(i=j)[(T-1)^{\zeta} - (T-2)^{\zeta}]}{D_{T,i}}\end{aligned}$$

Given that $\beta(z_{T,k}) = 1$, we have $\gamma(z_{tk}) = \alpha(z_{tk}) / \sum_{i=1}^{K} \alpha(z_{ti})$ for the incremental update. The corresponding incremental update equations for a Gaussian emission model (as reported in (Stenger et al. 2001)) are

$$\mu_j^{(T)} = \frac{\sum_{t=1}^{T-1} \gamma(z_{tj})}{\sum_{t=1}^{T} \gamma(z_{tj})} \mu_j^{(T-1)} + \frac{\gamma(z_{Tj})}{\sum_{t=1}^{T} \gamma(z_{tj})} \mathbf{x}_T$$

and

$$\begin{aligned}\mathbf{S}_j^{(T)} = \frac{\sum_{t=1}^{T-1} \gamma(z_{tj})}{\sum_{t=1}^{T} \gamma(z_{tj})} \mathbf{S}_j^{(T-1)} \\ + \frac{\gamma(z_{Tj})}{\sum_{t=1}^{T} \gamma(z_{tj})} \left(\mathbf{x}_T - \mu_j^{(T)}\right)\left(\mathbf{x}_T - \mu_j^{(T)}\right)'\end{aligned}$$

where $(\cdot)'$ denotes the matrix transpose operation and $\mathbf{S}_j$ is the covariance matrix for state j.

Initialization The process begins by batch-learning initial parameter estimates from a small portion of the time-series. These estimates are used for $\mathbf{A}^{(1)}$, $\mu^{(1)}$, $\mathbf{S}^{(1)}$ and $\pi(\mathbf{z}_t)$. For the α values, we initialize $\alpha(z_{1j}) = \pi(z_{1j}) p(\mathbf{x}_1; \phi_j)$ for each j. Using Equation 7 and the definition of $D_{T,i}$, we compute $D_{2,i} = \sum_{j=1}^{K} \xi(z_{1i}, z_{2j})$, and $A_{ij}^{(2)} = \xi(z_{1i}, z_{2j}) / D_{2,i}$.

The estimates are then updated for each new observation, using the update equations given above. Algorithm 1 outlines the order in which the various terms are computed.

Algorithm 1 Incremental Learning

1: Batch learn initial parameter estimates.
2: Compute $D_{2,i}$ and $A_{ij}^{(2)}$ for all i, j.
3: **for all** $T > 2$ **do**
4: Compute α values for observation at time T.
5: Compute $\xi(z_{(T-1)i}, z_{Tj})$ values for all i, j.
6: Compute $\gamma(z_{Tj})$ and $D_{T,i}$ values for all i, j.
7: Update $A_{i,j}^{(T)}$ using incremental update rule.
8: Update $\mu_j^{(T)}$ and $\mathbf{S}_j^{(T)}$ using incremental update rules.

Experiments

We perform two segmentation tasks on simulated and real multivariate time series data, using our scale- and parameter-free regularized inertial HMMs. For comparison, we present the results of applying a standard K-state hidden Markov model as well as the Bayesian hierarchical Dirichlet process hidden Markov model (sticky HDP-HMM) in (Fox et al. 2011). We performed all tasks in an unsupervised manner, with state labels being used only for evaluation.

Data

A simulated dataset was generated using a two-state HMM with 3-D Gaussian emissions, with transition matrix

$$\mathbf{A} = \begin{pmatrix} 0.9995 & 0.0005 \\ 0.0005 & 0.9995 \end{pmatrix},$$

equal start probabilities and emission parameters $\mu_1 = (-1, -1, -1)^\top$, $\mu_2 = (1, 1, 1)^\top$, $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \text{diag}(3)$. Using this model, we generated one hundred time series consisting of ten-thousand time points each. Figure shows an example time series from this simulated dataset.

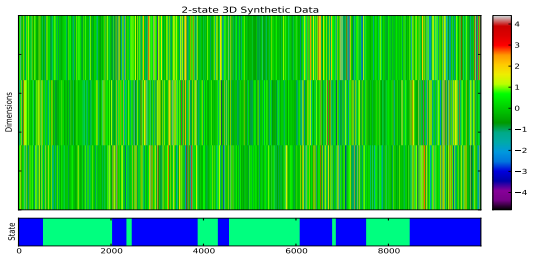

Figure 3: Simulated data example. Generated from two-state HMM with 3D Gaussian emissions and strong self-transitions.

The second dataset is generated from real-world forty-five dimensional human accelerometer data (Altun, Barshan, and Tunçel 2010) recorded for users performing five different activities, namely, playing basketball, rowing, jumping, ascending stairs and walking in a parking lot. The data were recorded from a single subject using five Xsens MTx™ units attached to the torso, arms and legs. Each unit had nine sensors, which recorded accelerometer (X, Y, Z) data, gyroscope (X, Y, Z) data and magnetometer (X, Y, Z) data, for a total of forty-five signals at each time point.

We generated one hundred multivariate times series from the underlying dataset, with varying activities (latent states) and number of segments. To generate these sets, the process was as follows. First, we uniformly chose the number of segments, between two and twenty. Then, for each segment, we chose an activity uniformly at random from among the five possible, and selected a uniformly random segment length proportion. The selected number of corresponding time points were extracted from the activity (keeping track of position in the sequence, and modulo the length of the sequence), rescaled to zero mean and unit variance, and appended to the output sequence. The final output sequence was truncated to ten thousand time points, or discarded if the sequence contained fewer than ten thousand points or fewer than two distinct activities. Additionally, prospective time series were rejected and replaced if they caused numerical instability issues for the algorithms tested, which occurred for some time series with many (or extremely short) segments. This process produced multivariate time series of fixed length, with varying number of segments, activities and segment lengths. The process was repeated to generate one hundred such time series of ten thousand time points each used in the quantitative analysis described in Section . An example of such generated data sequences is shown in Figure and the distribution of the time series according to number of activities and segments is shown in Figure .

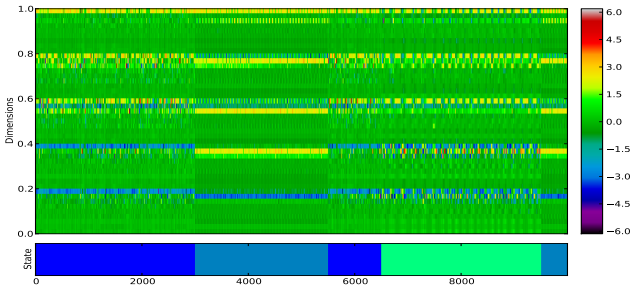

Figure 4: Human activities accelerometer data. Three state, 45-dimensional.

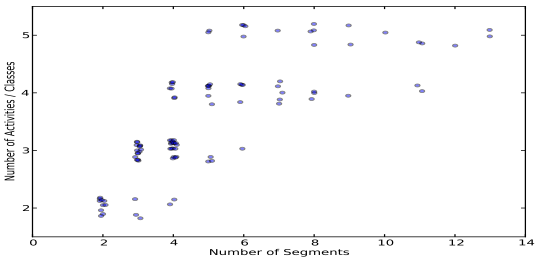

Figure 5: Distribution of Accelerometer Time Series Data.

Experimental Design

We compared performance of a standard K-state hidden Markov model with our batch-learned regularized HMMs on the two datasets described in the previous section. For the second dataset, we performed a quantitative analysis, treating the task as a multi-class classification problem, and measured the minimum zero-one loss under all possible permutations of output labels, to accommodate the fact that the output labels of an HMM may be a permuted mapping of the true labels. We measured the normalized variation of information (Meilă 2003) between the predicted state sequence and true state sequence, which is an information metric capturing the distance between two partitionings (clusterings) of a sequence. In addition to this, we considered the ratio of predicted number of segments to true number of segments, which gives us a sense of whether a method over- or under-segments data, and the absolute segment number ratio (ASNR), which is defined as ASNR $= \max(S_t, S_p)/\min(S_t, S_p)$, where S_t is the true number of segments in the sequence and S_p is the predicted number of segments. This value tells us how much a segmentation method diverges from the ground truth in terms of relative factor of segments. Lastly, we tracked the number of segments difference between the predicted segmentation and true segmentation and how many segmentations we done perfectly, giving the correct states at all correct positions.

To select parameters for the inertial regularized methods, we used the automated parameter selection procedure discussed in Section. To speed up evaluations, we only ran

the automated parameter selection process on ten randomly drawn examples, averaged the final ζ parameter value, and used the fixed value for all evaluations. The final ζ parameters are shown in Tables 2 and 1.

We also evaluated the sticky hierarchical Dirichlet process hidden Markov model (HDP-HMM) of Fox *et al.* (Fox et al. 2011) on both datasets. The publicly available HDP-HMM toolbox for MATLAB (Fox and Sudderth 2009) was used, with default settings for the priors. The Gaussian emission model with normal inverse Wishart (NIW) prior were used, and the truncation level L for each example was set to the true number of states, in fairness for comparing with the HMM methods developed here, which are also given the true number of states. The "stickiness" κ parameter was chosen in a data-driven manner by testing values of $\kappa = 0.001$, 0.01, 0.1, 1.0, 5.0, 10.0, 50.0, 100.0, 250.0, 500.0, 750.0 and 1000.0 for best performance over ten randomly selected examples each. The mean performance of the 500th Gibbs sample of ten trials was then taken for each parameter setting, and the best κ was empirically chosen. For the synthetic dataset, a final value of $\kappa = 10$ was chosen by this method. For the real human accelerometer data, a value of $\kappa = 100.0$ provided the best accuracy and relatively strong variation of information performance. These values were used for evaluation on each entire dataset, respectively.

To evaluate the HDP-HMM, we performed five trials one each example in the test dataset, measuring performance of the 1000th Gibbs sample for each trial. The mean performance was then computed for the trials, and the average of all one hundred test examples was recorded.

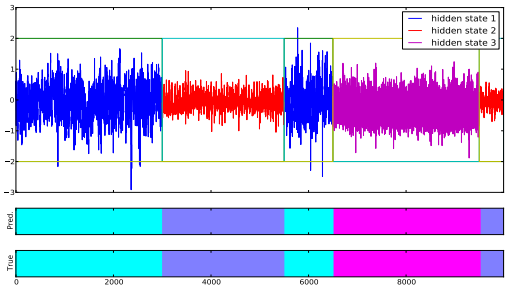

Figure 6: Example segmentation of human activities accelerometer data using inertial (MAP) HMM

Synthetic Data Results

Results for the synthetic dataset are shown in Table 1. Overall, the MAP regularized HMM had the strongest performance, with top scores on all metrics. The inertial pseudo-observation HMM also had strong performance, with extremely high accuracy and low variation of information. The standard HMM suffered from over-segmentation of the data (as reflected in the high SNR, ASNR, and SND scores), while the sticky HDP-HMM tended to under-segment the data. All methods were able to achieve fairly high accuracy on this simple, low-dimensional dataset.

Table 1: Results from quantitative evaluation on two-state, 3D synthetic data.

Method	Acc.	SNR	ASNR	SND	VOI	Perfect
Sticky HDP-HMM ($\kappa = 10$)	0.85	0.59	3.50	2.79	0.56	0/100
Standard HMM	0.87	172.20	172.20	765.91	0.62	0/100
MAP HMM ($\zeta = 2.28$)	0.99	0.96	1.13	0.51	0.07	2/100
Inertial PsO HMM ($\zeta = 8.19$)	0.99	0.87	1.43	1.15	0.14	1/100

Acc. = Average Accuracy (value of 1.0 is best)
SNR = Average Segment Number Ratio (value of 1.0 is best)
ASNR = Average Absolute Segment Number Ratio (value of 1.0 is best)
SND = Average Segment Number Difference (value of 0.0 is best)
VOI = Average Normalized Variation of Information (value of 0.0 is best)
Perfect = Total number of perfect/correct segmentations

Table 2: Results from quantitative evaluation on multivariate human accelerometer data.

Method	Acc.	SNR	ASNR	SND	VOI	Perfect
Sticky HDP-HMM ($\kappa = 100$)	0.60	0.75	4.68	5.03	0.95	0/100
Standard HMM	0.79	134.59	134.59	584.16	0.38	9/100
MAP HMM ($\zeta = 33.5$)	**0.94**	1.28	1.43	2.62	**0.14**	**48/100**
Inertial PsO HMM ($\zeta = 49.0$)	**0.94**	**1.03**	**1.29**	**1.29**	0.15	**48/100**

Human Activities Accelerometer Data Results

Results from the human accelerometer dataset are shown in Table 2. Both the MAP HMM and inertial pseudo-observation HMM achieved large gains in performance over the standard HMM model, with average accuracy of 94%. Furthermore, the number of segments was close to correct on average, with a value near one in both the absolute and simple ratio case. On average, the MAP HMM over-segmented by fewer than three segments, while the inertial pseudo-observation HMM over-segmented by fewer than two on average. Both methods were able to reproduce a perfect segmentation for 48 of the 100 test cases. The average normalized variation of information was low, at 0.14 and 0.15 for the MAP and pseudo-observation methods, respectively. Figure shows the segmentation results for the MAP regularized HMM on a typical sequence, displaying a single dimension of the multivariate time series for clarity. The regularized MAP HMM correctly segments the time series, as can be seen from the congruence between the true and predicted state transition histories (bottom of Figure).

In comparison, a standard hidden Markov model without inertial regularization achieved accuracy of 79%, but with average absolute SNR of 134.59 and average normalized variation of information value of 0.38. On average, the segmentations given by the standard HMM differed by 584.16 segments, compared to the fewer than three segments difference between the regularized HMMs and the ground truth sequences. The standard HMM was only able to produce perfect segmentations in 9% of the test cases. Thus, the inertial regularization produces drastic improvements for unsupervised segmentation of human accelerometer activity data.

Even more striking was the improvement over the sticky HDP-HMM of Fox *et al.* (Fox et al. 2011). The performance of that method was poor, with normalized variation of information near 1 (i.e., no correlation between the predicted and the true segment labels). The method tended to under-segment the data, often collapsing to a single uniform output state. This under-segmentation is reflected in the SNR ratio having a value below one. Problems for this method

may have resulted from the moderate dimensionality of the data, an issue suggested by Fox and Sudderth through private correspondence. The sticky HDP-HMM suffers from slow mixing rates as the dimensionality increases, and computation time explodes, being roughly cubic in the dimension. As a result, the one hundred test examples took several days of computation time to complete, whereas the inertial HMM methods took a few hours.

Discussion

Our results demonstrate the effectiveness of inertial regularization on HMMs for time series segmentation. Although derived in two independent ways, the MAP regularized and pseudo-observation inertial regularized HMM converge on a similar maximum likelihood update equation, and thus, have similar performance. Either version can be used for segmentation tasks, according to user preference.

The human activity task highlighted an issue with using standard HMMs for segmentation of time series with infrequent state changes, namely, over-segmentation. Incorporating regularization for state transitions provides a simple solution to this problem. Since our methods rely on changing a single update equation for a standard HMM learning method, they can be easily incorporated into HMM learning libraries with minimal effort. This ease-of-implementation gives a strong advantage over existing persistent-state HMM methods, such as the sticky HDP-HMM framework.

While the sticky HDP-HMM performed moderately well on the low-dimensional synthetic dataset, the default parameters produced poor performance on the real-world accelerometer data. It remains possible that different settings of hyperparameters may improve performance, but the cost of a combinatorial search through hyperparameter space combined with the lengthy computation time of the HDP-HMM method prohibited exhaustive exploration of the space. These results show, at very least, a strong dependence on hyperparameter settings for acceptable performance. In contrast, the inertial HMM methods presented here make use of a simple heuristic for automatically selecting the strength parameter ζ, which resulted in excellent performance on both datasets without the need for hand-tuning several hyperparameters.

One advantage of the sticky HDP-HMM over the methods presented here is the ability to segment a time series without knowing the number of states beforehand, whereas we require the number of states to be specified for our inertial methods. Thus, while the sticky HDP-HMM has poor performance on the two segmentation tasks, there exist tasks for which the HDP-HMM method may be better suited (e.g., when the correct number of states is unknown).

Related Work

Hidden Markov models for sequential data have enjoyed a long history, gaining popularity as a result of the widely influential tutorial by Rabiner (Rabiner 1989). Specific to the work presented here, the use of regularization for HMM parameters received a general treatment in (Gauvain and Lee 1994), for both transition and emission parameters. Our work details a more specific version of the regularization, useful for state persistence. Neukirchen and Rigoll (Neukirchen and Rigoll 1999) studied the use of regularization in HMMs for reducing parameter overfitting of emission distributions due to insufficient training data, but without an emphasis on inertial transitioning between states. Similarly, Johnson (Johnson 2007) proposed using Dirichlet priors on multinomial hidden Markov models as a means of enforcing sparse emission distributions.

In contrast, Fox *et al.* (Fox et al. 2011) develop a Bayesian sticky HMM to provide inertial state persistence. They present a method capable of learning a hidden Markov model without specifying the number of states or regularization strength beforehand, using a hierarchical Dirichlet process and truncated Gibbs sampling. Although our method requires the number of states to be specified in advance, their method requires a more complex approach to learning the model and suffers from poor performance for time series with more than ten dimensions. In contrast, our regularization only requires a small change to a single update equation, allowing drop-in regularization for standard Baum-Welch learning algorithms, and performs well on datasets of moderate dimensionality. Furthermore, several hyperparameters for the Bayesian priors must be chosen along with a truncation limit, thus not fully removing the need for specification of parameters, and in fact exacerbating it, since the sticky HDP-HMM requires more parameters than our methods. Our models only require the specification of two parameters, K and ζ, whereas the sticky HDP-HMM requires analogous truncation level L and κ parameters to be chosen, in addition to the hyperparameters on the model priors. We have shown that our inertial models are easily implemented, run efficiently, add almost no additional computation effort, and work well on data with moderate dimensions.

Conclusions

To segment multivariate time series data in an unsupervised manner, we derive two modified forms of hidden Markov model that effectively enforce state persistence. Although the methods derived here are simple, they perform well and are computationally efficient. Their simplicity is thus a feature and not a bug. We find that while the two inertial regularization methods differ in derivation and final mathematical form, their performance is similar on the datasets tested, allowing either to be used in practice.

We also derived incremental learning rules for the inertial MAP HMM, as well as presenting a simple heuristic method for automated regularization-strength parameter selection. Our experiments on synthetic and real-world datasets show the effectiveness of our methods, giving large improvements in performance over standard HMMs and the sticky HDP-HMM of Fox *et al.* (Fox et al. 2011).

The simplicity of the models pave the way for natural extensions, such as changing the form of the class conditional emission distributions to incorporate internal dynamics. Such extensions are the focus of future work.

References

Altun, K.; Barshan, B.; and Tunçel, O. 2010. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recogn.* 43(10):3605–3620.

Bishop, C. M. 2007. *Pattern Recognition and Machine Learning*. Springer.

Fox, E. B., and Sudderth, E. B. 2009. HDP-HMM Toolbox. `https://www.stat.washington.edu/~ebfox/software.html`. [Online; accessed 20-July-2014].

Fox, E. B.; Sudderth, E. B.; Jordan, M. I.; Willsky, A. S.; et al. 2011. A sticky HDP-HMM with application to speaker diarization. *The Annals of Applied Statistics* 5(2A):1020–1056.

Gauvain, J.-l., and Lee, C.-h. 1994. Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. *IEEE Transactions on Speech and Audio Processing* 2:291–298.

Gini, C. 1936. On the measure of concentration with special reference to income and statistics. In *Colorado College Publication*, number 208 in General Series, 73–79.

Johnson, M. 2007. Why doesn't EM find good HMM POS-taggers. In *In EMNLP*, 296–305.

Meilă, M. 2003. Comparing clusterings by the variation of information. In Schölkopf, B., and Warmuth, M., eds., *Learning Theory and Kernel Machines*, volume 2777 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 173–187.

Neal, R. M., and Hinton, G. E. 1999. Learning in graphical models. Cambridge, MA, USA: MIT Press. chapter A View of the EM Algorithm That Justifies Incremental, Sparse, and Other Variants, 355–368.

Neukirchen, C., and Rigoll, G. 1999. Controlling the complexity of HMM systems by regularization. *Advances in Neural Information Processing Systems* 737–743.

Rabiner, L. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2):257–286.

Stenger, B.; Ramesh, V.; Paragios, N.; Coetzee, F.; and Buhmann, J. 2001. Topology free hidden markov models: application to background modeling. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, 294–301 vol.1.

Wikipedia. 2004. Gini coefficient — Wikipedia, the free encyclopedia. [Online; accessed 8-June-2014].