Unsupervised Segmentation of Multivariate Time Series Through Inertial Hidden Markov Models

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

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

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

Abstract

Faced with the problem of segmenting a 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 state-of-the-art 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 the latter suffers from rapid state transitioning, over-segmentation and poor performance on a segmentation task involving human activity accelerometer data from the UCI Machine Learning Repository. The regularized methods developed here are able to perfectly segment the human activity data in roughly half of the 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. Lastly, we make progress towards tuning the regularization strength in an unsupervised manner and derive equations for online learning of the regularized HMM parameters.

1 Introduction

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

This is where the introduction goes. Want to give it a shot, Saeed or Nikolay?

2 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. 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.

We next discuss the input and output for our problem setup.

2.1 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 $1 < t \leq T$.

2.2 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.

3 Inertial Hidden Markov Models

3.1 Baseline: K-state Hidden Markov Model

As a baseline, a standard K-state HMM model with Gaussian emission densities is fit on the data. This model maximizes the likelihood of the data, but does not guarantee slow inertial transitioning among states. The number of states must be specified in advance, but no other parameters are needed. This gives us a starting point to improve on, allowing us to compare the regularized methods we develop.

3.2 Maximum A Posteriori (MAP) Regularized HMM

Following [5], 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

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

We then use MAP estimation in the M-step of the EM algorithm, to update the transition probability matrix. Maximizing, we obtain the update equation for the transition matrix, namely

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

Although we have a tractable, efficient algorithm for learning the regularized HMM and, given our prior, we can control the probability of self-transitions among states, 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

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

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

3.2.1 Scale-Free MAP Regularization

Astute readers may notice that the strength of the regularization diminishes with growing T, so that asymptotically the regularized estimates and unregularized estimates become equivalent. Figure 1 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 2. 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. Only one dimension of data is shown.

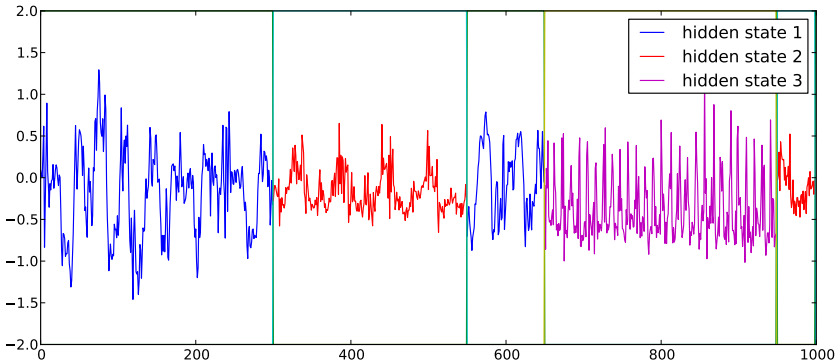
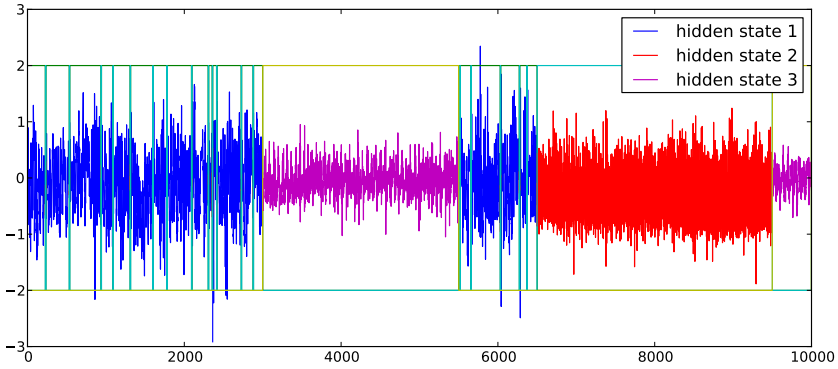


Figure 2: Human activities accelerometer data, long sequence. Regularization parameter from short sequence used here.

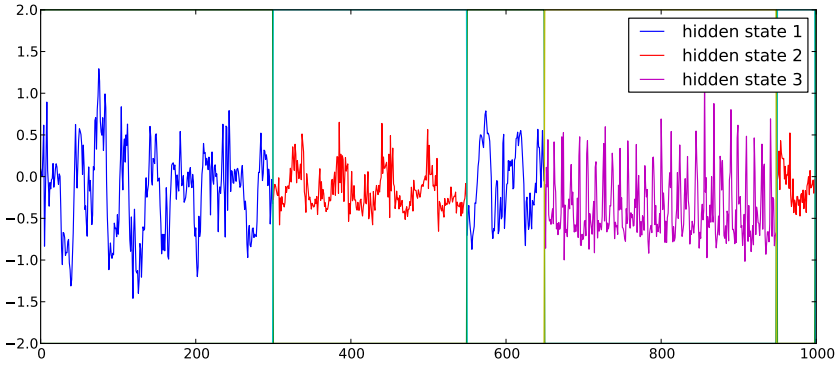


We desire a model where the regularization strength is scale-free, having roughly the same strength regardless of how the time series grows. To achieve this, we define the λ parameter to scale with the number of transitions, namely $\lambda = (T-1)^{\zeta}$, and our scale-free update equation becomes

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

This preserves the effect of regularization as T increases, and ζ becomes our new regularization parameter, controlling the strength of the regularization. Figures 3 and 4 shows the scale-free regularized result on both a short and long sequence, using the same regularization parameter. The parameter was chosen in reference to the short sequence, selecting the smallest parameter that still provided correct segmentation. As can be seen, the regularization strength is sufficient for the long sequence as well, illustrating the effectiveness of a scale-free parameterization.

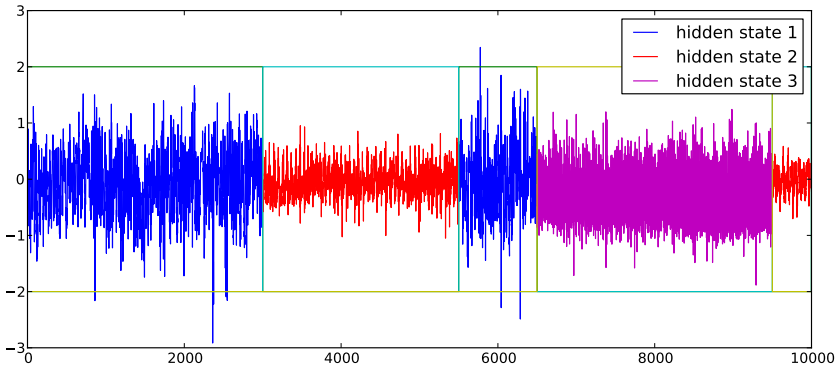
Figure 3: Human activities accelerometer data, short sequence, with minimum scale-free regularization parameter.



3.2.2 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 fictional observation, V. This is a binary random variable indicating that a transition was chosen at random from among all transitions, according to some distribution, and that the transition observed was a self-transition. 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 some 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

Figure 4: Human activities accelerometer data, long sequence, with minimum scale-free regularization parameter chosen in regards to short sequence. Correct segmentation is maintained despite the length of the time series increasing by an order of magnitude.



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 in the state transition sequence, so that the probability of selecting a self-transition increases as the number of non-self-transitions decreases. Using the fact that the number of non-self-transitions equals $(T-1) - \sum_{t=2}^{T}\sum_{k=1}^{K} z_{(t-1)k}z_{tk}$, we obtain

$$
\begin{aligned}
p &= \epsilon^{\text{NUM. OF NON-SELF-TRANSITIONS}} \qquad (2)\\
&= \epsilon^{(T-1)-\sum_{t=2}^{T}\sum_{k=1}^{K} z_{(t-1)k}z_{tk}}\\
&= \epsilon^{\sum_{t=2}^{T} 1-\sum_{t=2}^{T}\sum_{k=1}^{K} z_{(t-1)k}z_{tk}}\\
&= \epsilon^{\sum_{t=2}^{T}\sum_{k=1}^{K} z_{(t-1)k}-\sum_{t=2}^{T}\sum_{k=1}^{K} z_{(t-1)k}z_{tk}}\\
&= \epsilon^{\sum_{t=2}^{T}\sum_{k=1}^{K} z_{(t-1)k}-z_{(t-1)k}z_{tk}}\\
&= \prod_{t=2}^{T}\prod_{k=1}^{K} \epsilon^{z_{(t-1)k}-z_{(t-1)k}z_{tk}}. \qquad (3)
\end{aligned}
$$

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). Since $P(V = 1|\mathbf{Z}; \mathbf{A}) = p$, we have

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

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). Thus,

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

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

Clearly $\mathbf{V}$ is conditionally independent of $\mathbf{X}$ given the latent state sequence $\mathbf{Z}$. We now consider the joint density over $\mathbf{X}$, $\mathbf{V}$, and $\mathbf{Z}$, making use of our conditional independence assumption. We parameterize the joint density by $\theta = \{\pi, \mathbf{A}, \phi\}$, which are the start-state probabilities, state transition matrix and emission parameters, respectively. We have

$$P(\mathbf{X}, \mathbf{V}, \mathbf{Z}; \theta) = P(\mathbf{V}|\mathbf{Z}; \theta)P(\mathbf{z}_1; \theta)\left[\prod_{t=1}^{T} P(\mathbf{x}_t|\mathbf{z}_t; \theta)\right]\left[\prod_{t=2}^{T} P(\mathbf{z}_t|\mathbf{z}_{t-1}; \theta)\right]$$

and, taking the log of the likelihood,

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

Noting that

$$\begin{aligned}
P(\mathbf{z}_1; \theta) &= \prod_{k=1}^{K} \pi_k^{z_{1k}}, \\
P(\mathbf{x}_t|\mathbf{z}_t; \theta) &= \prod_{k=1}^{K} P(\mathbf{x}_t; \phi_k)^{z_{tk}}, \text{ and} \\
P(\mathbf{z}_t|\mathbf{z}_{t-1}; \theta) &= \prod_{k=1}^{K}\prod_{j=1}^{K} A_{jk}^{z_{(t-1)j}z_{tk}},
\end{aligned}$$

and substituting into $\ell(\mathbf{X}, \mathbf{V}, \mathbf{Z}; \theta)$ we obtain

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

Following Bishop [2], we define

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

to obtain

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

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}, namely

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

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

Ignoring normalization, we see that

$$A_{jk} \propto \begin{cases} \sum_{t=2}^{T} \xi(z_{(t-1)k}, z_{tk}) + \lambda \sum_{t=2}^{T} [\gamma(z_{(t-1)k}) - \xi(z_{(t-1)k}, z_{tk})] & \text{if } j = k \\ \sum_{t=2}^{T} \xi(z_{(t-1)j}, z_{tk}) & \text{otherwise.} \end{cases}$$

Thus, λ is a multiplier of additional mass contributions for self-transitions, where the contributions are the difference between $\gamma(z_{(t-1)k})$ and $\xi(z_{(t-1)k}, z_{tk})$. These two quantities represent, respectively, the expectation of being in a state k 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.

Similar to the MAP case, we let $\lambda = (T-1)^{\zeta}$ to maintain consistent regularization strength in the face of increasing sequence length, where ζ becomes our new regularization parameter.

3.2.3 Towards Parameter-Free Regularization

Our 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 $\lambda \in [0, 5]$), and perform a search on this interval for a value that gives sufficient regularization. "Sufficient regularization" is defined with regards to the Gini ratio [6, 13], 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.

We perform a binary search on our search interval, and for each parameter we train a model and evaluate the Gini ratio of the computed segment lengths. If the ratio is greater than one-half, we restrict ourselves to the upper interval and recurse. If the ratio is less than one-half, we store the current parameter value, reduce the parameter space to the lower interval and recurse. At the base of our recursion, we see if the interval width is smaller than some ϵ value, then return the parameter. On returning, we pass back the result from the recursion branch we took. As a special case, when the current parameter value has satisfied the Gini ratio condition but the left-branch recursion has not, we return the current parameter value rather than the child recursion value. Doing this guarantees that the algorithm will return the smallest regularization parameter value satisfying the Gini coefficient criterion, if such a value exists in the search space. Thus we have a parameter-free method of regularizing the segmentation task, at a cost of increasing runtime complexity by a factor of $O(\log_2(R/\epsilon))$, where R is the range of the parameter space.

4 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.* [12] 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 [9].

4.1 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

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

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

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

Also, because $\beta(z_{T,k}) = 1$, we have $\gamma(z_{tk}) = \alpha(z_{tk})/\sum_{i=1}^{K}\alpha(z_{ti})$. The corresponding incremental update equations for a Gaussian emission model (as reported in [12]) 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

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

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

4.2 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 1 and the definition of $D_{T,i}$, we compute

$$\begin{aligned}D_{2,i} &= \sum_{j=1}^{K}\xi(z_{1i}, z_{2j}), \\ A_{ij}^{(2)} &= \frac{\xi(z_{1i}, z_{2j})}{D_{2,i}}.\end{aligned}$$

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.

4.3 Robust Online Prediction

In keeping with our desire for slow state transitions, we now consider the problem of incremental prediction. If an observation at time t (the current time step) is an outlier, we cannot know whether

Algorithm 1

procedure INCREMENTAL LEARNING OF REGULARIZED HMM
    Batch learn initial parameter estimates from short segment of data.
    Compute $D_{2,i}$ and $A_{ij}^{(2)}$ for all i, j.
For $T > 2$:
    Compute α values for observation at time T.
    Compute $\xi(z_{(T-1)i}, z_{Tj})$ values for all i, j.
    Compute $\gamma(z_{Tj})$ and $D_{T,i}$ values for all i, j.
    Update $A_{i,j}^{(T)}$ using incremental update rule.
    Update $\mu_j^{(T)}$ and $\mathbf{S}_j^{(T)}$ using incremental update rules.

the model should remain in the same hidden state, treating the outlier as an anomaly, or transition to a new hidden state. To overcome this limitation, we propose delayed prediction of state labels using a sliding window of length w. As the window moves through the observation sequence, the Viterbi algorithm is performed on the section of data within the window and a prediction for the $(t-w/2)$th observation is output. This allows for "future" observations to affect "past" observations within the window, via the backtracking maximization performed by the algorithm. Assuming we batch-learned an initial segment of data longer than $w/2$, we can begin output delayed state label predictions as soon as incremental learning begins.

5 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) of Fox *et al.* [4]. We performed all tasks in an unsupervised manner, with state labels being used only for evaluation.

5.1 Data

TO DO: Need to describe how synthetic data was generated.

The second dataset is generated from real-world forty-five dimensional (45D) human accelerometer data [1] 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, we chose among the five different activities and chose anywhere from two to twenty segments. 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 6.2. An example of such generated data sequences is shown in Figure 5 and the distribution of the time series according to number of activities and segments is shown in Figure 6.

Figure 5: Human activities accelerometer data. Three state, 45-dimensional.

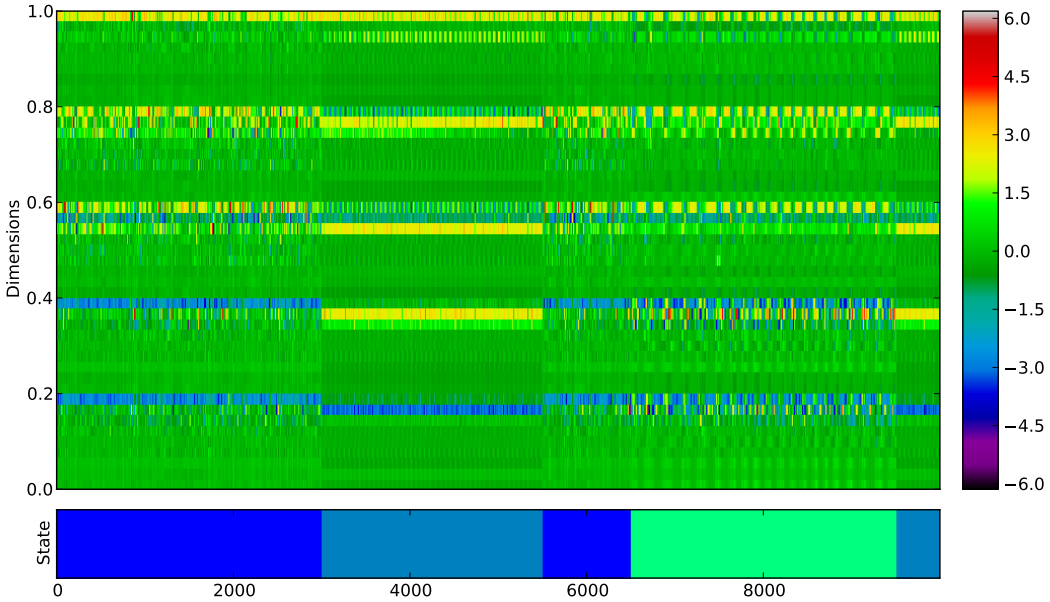
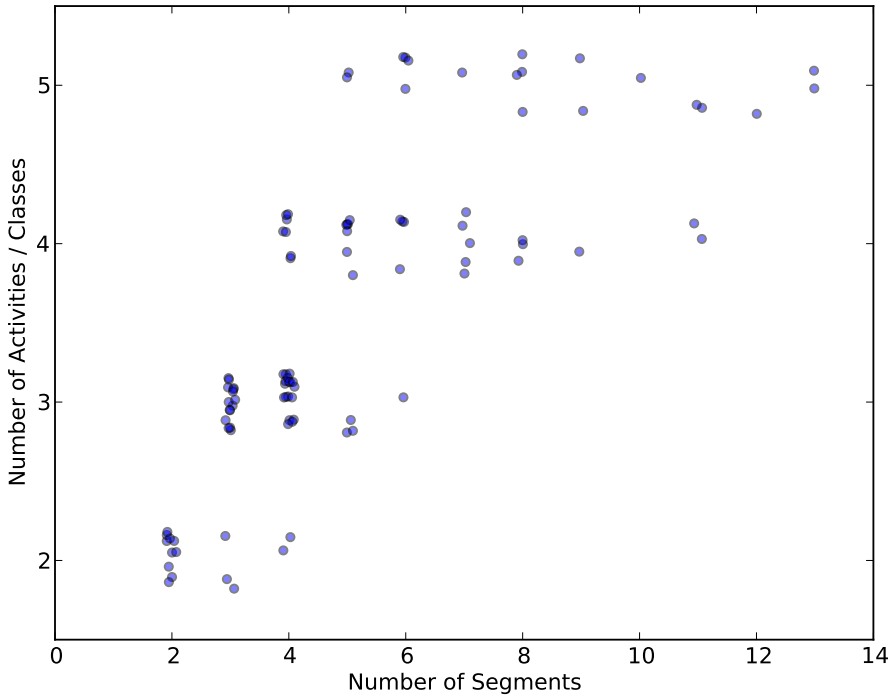


Figure 6: Distribution of Accelerometer Time Series Data (w/jitter).



5.2 Methodology

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 [8] between the predicted state sequence and true state sequence, which is an information metric capturing the symmetric two-way conditional entropy 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

$$\text{ASNR} = \frac{\max(S_t, S_p)}{\min(S_t, S_p)}$$

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 speed up evaluation, we used a fixed MAP regularization parameter for each set of tests ($\zeta = 74$), which is the largest regularization value capable of being used on the test system. As the parameter value decreases, we get performance more similar to the standard HMM, where performance is identical for $\zeta = 0$ and $\zeta = 1$, for MAP and pseudo-observation regularization, respectively.

We also evaluated the "sticky" hierarchical Dirichlet process hidden Markov model (HDP-HMM) of Fox *et al.* [4] on the one hundred time series human activity accelerometer dataset. The publicly available HDP-HMM toolbox for MATLAB [3] 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 $\kappa = 0.1$ (the default), 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, ... For the real human accelerometer data, we found a value of $\kappa = 100.0$ provided the best accuracy and relatively strong variation of information performance. These parameter 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.

6 Results

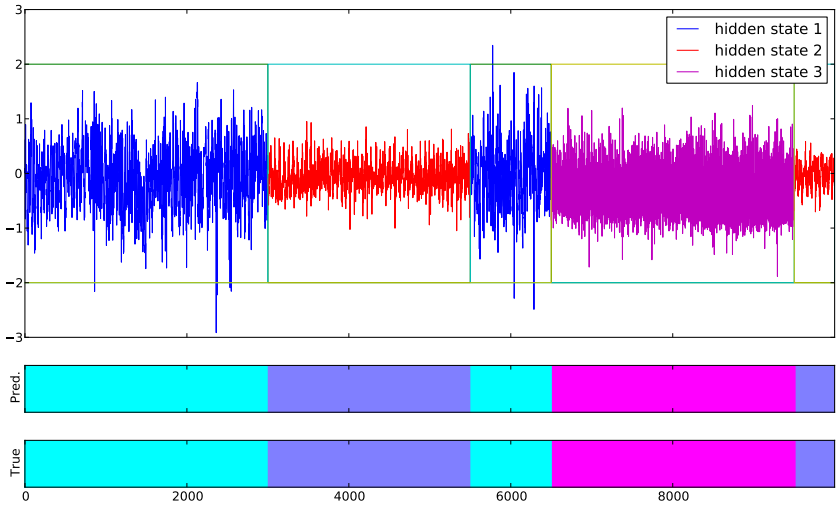
6.1 Simulated Data Results

TO DO: Will need to revisit this section once I have the final results from synthetic dataset.

6.2 Human Activities Accelerometer Data Results

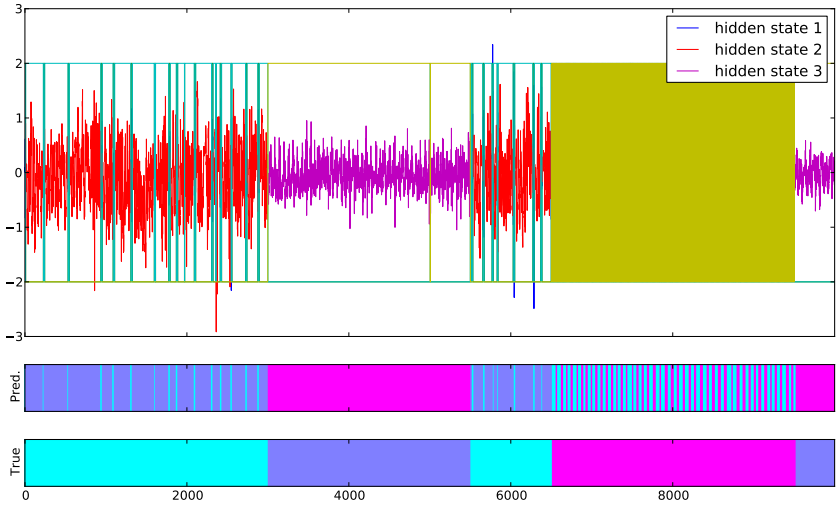
As an example of performance on the forty-five dimensional human activities accelerometer data, Figure 7 shows the segmentation results for the MAP regularized HMM on a typical sequence, displaying a single dimension of the multivariate time series for clarity. The regularized MAP HMM correctly segments the time series, as can be seen from the congruence between the true and predicted state transition histories (bottom of Figure 7). A final parameter value of $\zeta = 1.97$ was automatically discovered through the search process described in Section 3.2.3, and results are shown for that parameter value.

Figure 7: Segmentation of human activities accelerometer data using regularized MAP HMM.



In contrast, an unregularized HMM performs poorly on the same task, since simply seeking to maximize the log-likelihood of the observations may require that states often transition between neighboring points. The segmentation errors can be seen at the bottom of Figure 8.

Figure 8: Segmentation of human activities accelerometer data using a standard HMM.



Using the one hundred time series human activities dataset described in Section 5.1, we performed a quantitative analysis comparing the performance of standard and regularized MAP HMMs on the unsupervised segmentation task. The results are shown in Table 1.

TO DO: Discussion of results, once we have final results for all.

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.* [4]. The performance of that method was poor, with normalized variation of information near 1 (i.e., no correlation between predicted labels and the true segment labels). Problems for this method arose from the moderate dimensionality of the data, an issue confirmed by Fox and Sudderth through private correspondence. The sticky HDP-HMM suffers from slow mixing rates as the dimensionality increases,

Table 1: Results from quantitative evaluation on multivariate human accelerometer data.

Method	Accuracy	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**

Accuracy = 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

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.

7 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 recent sticky HDP-HMM framework of Fox *et al.* [4].

8 Related Work and Conclusions

Hidden Markov models for sequential data have enjoyed a long history, gaining popularity as a result of the widely influential tutorial by Rabiner [11]. Specific to the work presented here, the use of regularization for HMM parameters received a general treatment in [5], for both transition and emission parameters. Our work details a more specific version of the regularization, useful for state persistence. Neukirchen and Rigoll [10] 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 [7] proposed using Dirichlet priors on multinomial hidden Markov models as a means of enforcing sparse emission distributions.

In contrast, Fox *et al.* [4] 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 the methods presented here. 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 the inertial models are easily implemented, run efficiently, add almost no additional computation effort, and work well on data with over ten dimensions.

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 practically indistinguishable on the real-world data tested, allowing either to be used in practice. The simplicity of the models thus pave the way for natural modifications and extensions, such as changing the form of the class conditional emission distributions to incorporate internal dynamics. Such extensions are the focus of future work.

9 Acknowledgments

The authors would like to thank Emily Fox and Erik Sudderth for their discussions, feedback and assistance with use of the HDP-HMM toolbox.

References

[1] Kerem Altun, Billur Barshan, and Orkun Tunçel, *Comparative study on classifying human activities with miniature inertial and magnetic sensors*, Pattern Recogn. **43** (2010), no. 10, 3605–3620.

[2] Christopher M Bishop, *Pattern Recognition and Machine Learning*, Springer, 2007.

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

[4] Emily B Fox, Erik B Sudderth, Michael I Jordan, Alan S Willsky, et al., *A sticky HDP-HMM with application to speaker diarization*, The Annals of Applied Statistics **5** (2011), no. 2A, 1020–1056.

[5] Jean-luc Gauvain and Chin-hui Lee, *Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains*, IEEE Transactions on Speech and Audio Processing **2** (1994), 291–298.

[6] Corrado Gini, *On the measure of concentration with special reference to income and statistics*, Colorado College Publication, General Series, no. 208, 1936, pp. 73–79.

[7] Mark Johnson, *Why doesn't EM find good HMM POS-taggers*, In EMNLP, 2007, pp. 296–305.

[8] Marina Meilă, *Comparing clusterings by the variation of information*, Learning Theory and Kernel Machines (Bernhard Schölkopf and ManfredK. Warmuth, eds.), Lecture Notes in Computer Science, vol. 2777, Springer Berlin Heidelberg, 2003, pp. 173–187.

[9] Radford M. Neal and Geoffrey E. Hinton, *Learning in graphical models*, MIT Press, Cambridge, MA, USA, 1999, pp. 355–368.

[10] Christoph Neukirchen and Gerhard Rigoll, *Controlling the complexity of HMM systems by regularization*, Advances in Neural Information Processing Systems (1999), 737–743.

[11] Lawrence Rabiner, *A tutorial on hidden markov models and selected applications in speech recognition*, Proceedings of the IEEE **77** (1989), no. 2, 257–286.

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

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