Change Modeling in Multivariate Streaming Time-Series

George D. Montañez
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

Building on recent advances in probabilistic temporal regularization for hidden Markov models, we develop an online learning extension for the inertial HMM framework, allowing for scaling to arbitrarily large datasets. In addition, we develop a robust delayed online prediction method, controlling for the trade-off between optimal and timely state prediction. Our method is tested on synthetic and real-world datasets, showing the effectiveness of our learning and prediction algorithms.

1 Introduction

Processing temporal information, aka time-series, is a crucial aspect of many AI systems. The main distinction between temopral data and static data is, in almost all time-series data, *change of value* from one time point to another is inevitable. There are two main sources that cause these changes. The first category of changes are the result of the normal progression of time-series behavior plus some level of uncertainty (or noise) over time, and therefore, are *expected*. This type of changes are typically modeled via dynamical equations and/or dynamic graphical models. The second type of changes, however, are unexpected due to the occurance of some (external) events, and are typically referred to as *anomalies*. Depending on how fundamental the impact of an anomaly is, it can be either volatile or persistent, which are respectively referred to as *outliers* and *change points* in the literature. There is an extensive volume of work in the literature to detect both outliers [Tsay, 1988; Chandola *et al.*, 2009; Galeano *et al.*, 2006] and change points [Kawahara *et al.*, 2007; Xie *et al.*, 2013; Liu *et al.*, 2013; Ray and Tsay, 2002].

As opposed to outliers, change points attribute to more profound and systematic changes in the underlying behavior of the data over time. Properly charectarizing this type of anomalies significantly affects the way that AI systems understand and interpret the data. As a result, in this paper, we focus on modeling this type of changes. By modeling persistent changes, we do not mean only detecting them but also recognizing the new behavior of the time-series after the change. From this perspective, the desired solution will be different from typical change point detection techniques. More precisely, we also want to recognize the *state* of behavior the data-generating system is operating in after change, which in general is not observed. To this end, a standard approach in Machine Learning is to incorporate latent space models such as Hidden Markov Models (HMM) and Dynamical Systems (DS). While these methods work well for many similar problems, they can result in high rate of false positives when it comes to detecting the persistent changes in time-series. The reason behind this observation is that a true transition between two states of behavior does not happen very frequently over time; in other words, after changing to a new state, the system tends to stay in that state for a while. Following [Montañez *et al.*, 2015], we refer to this property as the *inertial property*. The general HMM and DS are not well-equipped to capture this property, however.

To account for the inertial property in the input time-series, the straightforward approach is to directly encode this property into the model. To this end, Fox *et. al.* [Fox *et al.*, 2011] have proposed incorporating a non-parametric Bayes method to add *stickyness* to HMMs. The resulting method is called the *Hierarchical Dirichlet Process* HMM (HDP-HMM). Despite the nice theoretical formulation of HDP-HMM, in practice, HDP-HMM cannot properly handle time-series with dimension more than 10. This is because, due to the existence of many hyperparameters, the search for the best initialization of the model is exponentially expensive. More recently, Montañez *et. al.* [2015] have proposed *Inertial* HMM, which is much simpler yet way more effective in practice. Compared to HDP-HMM, Inertial HMM has only one tuning parameter and can handle moderate dimensional data; furthermore, learning for Inertial HMM is much faster in terms of time complexity. All of these makes Inertial HMM a practical solution for many real-world time-series change modeling problems.

Despite the nice theoretical and practical properties of HDP-HMM and Inertial HMM, both methods are batch frameworks, meaning that both learning and inference are done offline on batch time-series data. However, in many applications, the temporal information is consumed by the AI system in the streaming fashion. For instance, never-ending learning systems are required to learn from an act upon every single piece of information that is streamed into the system in real-time. In such scenarios, batch processing of the data is either infeasible or inefficient at best. Moreover, even for batch problems, depending on the computational resources,

the memory and CPU requirements can be prohibitive to process large-scale time-series data at once. In these cases, streaming the batch data into the system is one way of addressing the scalability problem.

As a result, in this paper, we propose an online framework based on the Inertial HMM to address the problem of change modeling in multi-variate streaming time-series. In particular, we propose (A) an online learning algorithm for Inertial HMM and (B) a robust inference algorithm for online detection of change points (i.e. the state transitions). The former provides the Inertial HMM with the ability to constantly update itself as it consumes the streaming data, while the latter is crucial in the sense that robust recognition of outliers vs. change-points in real-time is key to keeping the rate of false positives low. To evaluate our proposed framework, we have applied it to large-scale time-series which are fed to the model in the streaming fashion. The experimental results show the merits of the proposed methodology in terms of both accuracy and efficiency.

2 Problem Statement

Here is where we must state the problem we are trying to solve.

3 Inertial Hidden Markov Models

Montañez *et. al.* [2015] recently introduced the inertial HMM solution for learning temporally regularized hidden Markov models for segmentation and characterization of multivariate time series. The inertial HMM is a K-state hidden Markov model with a modified likelihood function that causes increased state persistence as a direct consequence of maximizing the likelihood function. Because of this, the inertial HMM allows for simple expectation maximization [Dempster *et al.*, 1977] training of the model.

3.1 Notation and Preliminaries

Here I will define α, β, ξ, γ, the 1-of-K notation for $\mathbf{z}_t$ and other things necessary to understand the next section.

3.2 Likelihood Function Modifications

Inertial HMMs propose redefining the likelihood function in one of two ways. The first is by applying a modified self-transition Dirichlet prior to the state transition matrix, which is scaled in proportion to sequence length in order to maintain consistent strength of regularization. This is referred to as the *MAP inertial HMM* by the authors. The second is to use pseudo-observations for temporal regularization, where the observations are added to the joint complete data likelihood through use of a set of binary indicator random variables. This second method is referred to as the *inertial pseudo-observation HMM*. The two methods lead to distinct, yet related, mathematical forms for the likelihood function and both allow for learning via expectation maximization. Furthermore, the authors report both methods to have similar performance on the tested datasets. Therefore, we extend the conceptually simpler MAP inertial HMM. Since the inertial regularization methods rely on standard EM learning, one can naturally incorporate online EM learning techniques into such systems.

3.3 Update Equation

For the MAP inertial HMM, the scale-free update equation for the state transition matrix A is defined as

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

where $\mathbb{1}(\cdot)$ denotes the indicator function, T the length of the time series and $\xi(z_{(t-1)j}, z_{tk}) = \mathbb{E}[z_{(t-1)j}z_{tk}]$. This modified update equation is what distinguishes the inertial HMM from a standard HMM, and thus requires derivation of a novel online update equation. We provide the required equations in the next section.

4 Online Learning of Inertial HMMs

We extend the work of Stenger *et al.* [2001] and Montañez *et al.* [2015] to provide an online learning algorithm for the regularized MAP inertial hidden Markov model, which allows scaling to arbitrarily large datasets. Theoretical justification for incremental online EM learning is given in [Neal and Hinton, 1999].

4.1 Parameter Update Equations

Define

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

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

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

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

$$\begin{aligned}\xi(\mathbf{z}_{t-1}, \mathbf{z}_t) &= P(\mathbf{z}_{t-1}, \mathbf{z}_t|\mathbf{X}) \\ &= \frac{\alpha(\mathbf{z}_{t-1})p(\mathbf{x}_t|\mathbf{z}_t;\phi)p(\mathbf{z}_t|\mathbf{z}_{(t-1)})\beta(\mathbf{z}_t)}{p(\mathbf{X})} \\ &= \frac{\alpha(z_{(t-1)i})p(\mathbf{x}_t|z_{tj};\phi)A_{ij}^{(t-1)}}{p(\mathbf{X})}\end{aligned} \tag{2}$$

where

$$\alpha(z_{tj}) = \left[\sum_{i=1}^{K}\alpha(z_{(t-1)i})A_{ij}^{(t-1)}\right]p(\mathbf{x}_t|z_{tj};\phi). \tag{3}$$

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

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

Given that $\beta(z_{T,k}) = 1$, we have

$$\gamma(z_{tk}) = \frac{\alpha(z_{tk})}{p(\mathbf{X})} \tag{4}$$

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

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

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

4.2 Scaling Factors

For the recursive update process, we see that the $\alpha(\cdot)$ values computed in Equation (3) will rapidly decrease towards zero and cause underflow issues, since at each step the previously computed value is multiplied by two small quantities. Since the same issue arises in standard hidden Markov models, we also make use of the same solution: we replace the $\alpha(\cdot)$ values with rescaled versions that remain at the order of unity for each step [Bishop, 2007].

We define a rescaled function, $\hat{\alpha}(\cdot)$, as

$$\begin{aligned}\hat{\alpha}(z_{ti}) &= p(z_{ti}|\mathbf{x}_1, \ldots, \mathbf{x}_t) \\ &= \frac{p(\mathbf{x}_1, \ldots, \mathbf{x}_t, z_{ti})}{p(\mathbf{x}_1, \ldots, \mathbf{x}_t)} \\ &= \frac{\alpha(z_{ti})}{p(\mathbf{x}_1, \ldots, \mathbf{x}_t)}.\end{aligned} \tag{5}$$

As defined, $\hat{\alpha}(\cdot)$ is a probability distribution over K states, and will thus remain well-behaved at each time step. If we further define $c_t := p(\mathbf{x}_t|\mathbf{x}_1, \ldots, \mathbf{x}_{t-1})$, by the chain rule we have

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_t) = \prod_{m=1}^{t} c_m$$

and

$$\alpha(z_{ti}) = \left(\prod_{m=1}^{t} c_m\right)\hat{\alpha}(z_{ti}). \tag{6}$$

From Equations (6) and (3), we obtain the following recurrence for $\hat{\alpha}(\cdot)$:

$$\begin{aligned}\hat{\alpha}(z_{tj}) &= \frac{1}{c_t}\left[\sum_{i=1}^{K}\hat{\alpha}(z_{(t-1)i})A_{ij}^{(t-1)}\right]p(\mathbf{x}_t|z_{tj};\phi) \\ &= \frac{1}{c_t}R_j(\mathbf{x}_t),\end{aligned} \tag{7}$$

where $c_t = p(\mathbf{x}_t|\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}) = \sum_{j=1}^{K} R_j(\mathbf{x}_t)$, and thus can be seen as a normalization constant.

We further derive expressions for the γ and ξ in terms of $\hat{\alpha}(\cdot)$, as

$$\begin{aligned}\gamma(z_{tk}) &= \frac{\alpha(z_{tk})}{p(\mathbf{X})} \\ &= \frac{\left(\prod_{m=1}^{t} c_m\right)}{\left(\prod_{m=1}^{T} c_m\right)}\hat{\alpha}(z_{tk}) \\ &= \frac{\hat{\alpha}(z_{tk})}{\left(\prod_{m=t+1}^{T} c_m\right)},\end{aligned} \tag{8}$$

and

$$\begin{aligned}\xi(\mathbf{z}_{t-1}, \mathbf{z}_t) &= \frac{\alpha(z_{(t-1)i})p(\mathbf{x}_t|z_{tj};\phi)A_{ij}^{(t-1)}}{p(\mathbf{X})} \\ &= \frac{\left(\prod_{m=1}^{t-1} c_m\right)}{\left(\prod_{m=1}^{T} c_m\right)}\hat{\alpha}(z_{(t-1)i})p(\mathbf{x}_t|z_{tj};\phi)A_{ij}^{(t-1)} \\ &= \frac{\hat{\alpha}(z_{(t-1)i})p(\mathbf{x}_t|z_{tj};\phi)A_{ij}^{(t-1)}}{\left(\prod_{m=t}^{T} c_m\right)}.\end{aligned} \tag{9}$$

Because t is always both the current and final time step, we have $t = T$ and these simplify to

$$\gamma(z_{tk}) = \hat{\alpha}(z_{tk}), \tag{10}$$

$$\xi(\mathbf{z}_{t-1}, \mathbf{z}_t) = \frac{1}{c_t}\left[\hat{\alpha}(z_{(t-1)i})p(\mathbf{x}_t|z_{tj};\phi)A_{ij}^{(t-1)}\right]. \tag{11}$$

4.3 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 $\hat{\alpha}(z_{1j}) = (1/c_1)\pi(z_{1j})p(\mathbf{x}_1|z_{1j};\phi)$ for each j, where c_1 is again the value that normalizes $\hat{\alpha}(\mathbf{z}_1)$. Using Equation 1 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.
9: **end for**

4.4 Robust Online Prediction

We now consider the problem of online prediction. If an observation at time t (the current time step) is an outlier, we cannot know whether 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 second observation is output. The first observation is used to represent all past history, via the Markov property, and the remainder of the window allows for "future" observations to affect "past" observations, via the backtracking maximization performed by the Viterbi algorithm. We can begin to output delayed state label predictions as soon as w observations arrive.

The length of the sliding window controls the trade-off between optimal state prediction (which occurs when w equals the length of all future and past observations) and the need for timely predictions. This parameter can be set using cross-validation when labeled state data is available.

5 Experiments

5.1 Datasets

Our synthetic data is generated from a two-state three-dimensional hidden Markov model with transition matrix

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

having equal start probabilities and emission parameters equal to $\mu_1 = (-1, -1, -1)^\top$, $\mu_2 = (1, 1, 1)^\top$, $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \text{diag}(3)$. Using this model, we generated one hundred time series of length 100,000.

The second dataset we constructed using real-world human accelerometer data [Altun *et al.*, 2010], collected using Xsens MTx™ units attached to the torso, arms and legs of human volunteers, resulting in forty-five dimensional signals. The signals were recorded for volunteers performing five different activities, such as playing basketball, jumping, walking on a flat surface, rowing and ascending stairs. The signals consist of accelerometer, gyroscope and magnetometer data, which we consider as a single 45D multivariate time series.

From this human activity data we generated one hundred multivariate time series, with varying number of segments and varying activities, using a five-state HMM with 90% probability of self-transition, 2.5% probability of non-self-transition (equal for all states), equal start probability and emissions generated by using the actual sensor data in serial fashion for the five activities, modulo the length of the stream. One hundred time series of 100,000 time ticks were generated in this manner.

5.2 Experimental Methodology

5.3 Results

6 Discussion

7 Related Work

8 Conclusions

Acknowledgments

GDM is supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1252522. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors alone and do not necessarily reflect the views of the National Science Foundation or any other organization.

References

[Altun *et al.*, 2010] Kerem Altun, Billur Barshan, and Orkun Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recogn.*, 43(10):3605–3620, October 2010.

[Bishop, 2007] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007. 627–629.

[Chandola *et al.*, 2009] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[Dempster *et al.*, 1977] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[Fox *et al.*, 2011] 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(2A):1020–1056, 2011.

[Galeano *et al.*, 2006] Pedro Galeano, Daniel Peña, and Ruey S Tsay. Outlier detection in multivariate time series by projection pursuit. *Journal of the American Statistical Association*, 101(474):654–669, 2006.

[Kawahara *et al.*, 2007] Yoshinobu Kawahara, Takehisa Yairi, and Kazuo Machida. Change-point detection in time-series data based on subspace identification. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 559–564. IEEE, 2007.

[Liu *et al.*, 2013] Song Liu, Makoto Yamada, Nigel Collier, and Masashi Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83, 2013.

[Montañez *et al.*, 2015] George D Montañez, Saeed Amizadeh, and Nikolay Laptev. Inertial Hidden Markov Models: Modeling change in multivariate time series. *The Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*, 2015.

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

[Ray and Tsay, 2002] Bonnie K Ray and Ruey S Tsay. Bayesian methods for change-point detection in long-range dependent processes. *Journal of Time Series Analysis*, 23(6):687–705, 2002.

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

[Tsay, 1988] Ruey S Tsay. Outliers, level shifts, and variance changes in time series. *Journal of forecasting*, 7(1):1–20, 1988.

[Xie *et al.*, 2013] Yao Xie, Jiaji Huang, and Rebecca Willett. Change-point detection for high-dimensional time series with missing data. *Selected Topics in Signal Processing, IEEE Journal of*, 7(1):12–27, 2013.