Bayesian Online Prediction of Change Points

Diego Agudelo-España
MPI for Intelligent Systems

Sebastian Gomez-Gonzalez
MPI for Intelligent Systems

Stefan Bauer
MPI for Intelligent Systems

Bernhard Schölkopf
MPI for Intelligent Systems

Jan Peters
MPI for Intelligent Systems, TU Darmstadt

Abstract

Online detection of instantaneous changes in the generative process of a data sequence generally focuses on retrospective inference of such change points without considering their future occurrences. We extend the Bayesian Online Change Point Detection algorithm to also infer the number of time steps until the next change point (i.e., the residual time). This enables to handle observation models which depend on the total segment duration, which is useful to model data sequences with temporal scaling. The resulting inference algorithm for *segment detection* can be deployed in an online fashion, and we illustrate applications to synthetic and to two medical real-world data sets.

1 INTRODUCTION

An underlying assumption in time series models is often that the parameters of the data generative process remain fixed across a sequence of observations. Yet, it is not uncommon to encounter phenomena where this assumption does not hold. Switching models [5], also known as change point models, account for the time-variability in a time series model by splitting a sequence of observations into non-overlapping segments, each with fixed model parameters. Change point detection (CPD) algorithms aim to identify the boundaries of these segments, given the observed time series and an underlying predictive model (UPM) from which the segments of observations are generated.

Many of the previous approaches to CPD focus on the offline setting [10, 22], i.e., the detection algorithm treats the observations in batch mode, and they are focused on retrospective segmentation as opposed to predictive modeling. Online methods for CPD were proposed by

[1, 7] from a Bayesian perspective. The core idea behind Bayesian Online Change Point Detection (BOCPD) is to keep a probability distribution over the *run length* r_t, i.e., the elapsed time since the most recent change point (CP). When a new observation is available, the belief over the run length is updated accordingly. Importantly, as we explain in Section 2.2, the updated run length distribution enables one-step-ahead prediction in contrast with the offline approaches.

However, all these approaches have in common that they do not explicitly address inference of future CPs. We argue that this is of high practical importance for certain applications. Consider the scenario of medical time series, even though real-time classification or segmentation of time series is a challenging problem and already some solutions exist in many disciplines, online planning of interventions is often the ultimate goal and has received significantly less attention. For instance, in sleep research it was shown that acoustic stimulation has positive effects (for more details we refer to Section 5.2) only if the intervention is done in the up-shift of the slow wave. Thus, not only is online labeling required but likewise a prediction and uncertainty quantification of the current location within the current segment (i.e., slow wave phase). While the EEG time series used for sleep research might look rough and irregular at first, an endogenous circadian clock and regular sleep dynamics ensure that even in this case certain personalized duration patterns indeed exist.

We propose to extend the BOCPD framework along multiple directions in this work in order to predict future CPs. In particular,

- We show how to extend the original BOCPD algorithm to infer when the next CP will occur.

- We generalize such extension to perform *segment* detection. The term *segment* detection is deliberately preferred to emphasize that our approach not only infers the segment starting time but explicitly accounts

for its ending position as well.

- We point out the connections between Hidden semi-Markov Models (HSMMs) and BOCPD and show that the proposed model is equivalent to an HSMM which, however, supports online *segment* inference.

- The presented framework leads to a new class of UPMs that not only depend on the run length and the most recent observed data but also on the total *segment duration* d_t. This class of UPMs is suitable for modeling time series that share the same underlying functional shape but with a different time scale (e.g., ECG data while walking and running).

- We propose an algorithm to learn the model hyperparameters in a supervised fashion using maximum likelihood estimation (MLE). We provide implementations for both online inference and learning algorithms.

2 BACKGROUND

In the following we provide more details about the models we are building on top of (HSMMs and BOCPD) and point out the differences concerning inference and modeling assumptions.

2.1 HIDDEN SEMI-MARKOV MODELS

Hidden semi-Markov Models (HSMMs) represent a particular way to define a probability distribution over a data sequence $\mathbf{Y} = \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T$. As with Hidden Markov Models (HMMs), the observed data is assumed to be generated from a hidden sequence of discrete random variables $\mathbf{z} = z_1, z_2, \ldots, z_T$, where $z_t \in \{1, 2, \ldots, K\}$, through the so-called emission process. The semi-Markov property defines how the hidden variables are generated in a sequential manner. Namely, an HSMM explicitly models the number of consecutive time steps d_t that the hidden process remains in a particular state z_t through the duration model $p(d_t|z_t)$ [16]. HSMMs are fully parameterized by the transition model $p(z_t|z_{t-1})$ represented by the matrix $\mathbf{A} = \{A_{i,j}\}$, the probability mass function (p.m.f.) $p(z_1)$ of the initial hidden state $\boldsymbol{\pi} = \{\pi_i\}$, the duration model induced by matrix $\mathbf{D} = \{D_{i,d}\}$ and the emission parameters $\boldsymbol{\theta} = \{\theta_i\}$. Given that the i-th hidden state generates a segment of total duration d at time step t, the emission likelihood for the sequence of observations $\mathbf{Y}_{t,t+d-1} = \mathbf{y}_t, \mathbf{y}_{t+1}, \ldots, \mathbf{y}_{t+d-1}$ is

$$p(\mathbf{Y}_{t,t+d-1}|\boldsymbol{\theta}, z_t = i, d_t = d). \tag{1}$$

Note that the observations in the sequence $\mathbf{Y}_{t,t+d-1}$ are not necessarily conditionally independent among each other given the corresponding hidden variables. Moreover, they could even potentially depend on the total segment duration d. This allows to express more complex emission processes than with HMMs and to capture richer dependencies among distant observations.

2.2 BAYESIAN ONLINE CHANGE POINT DETECTION

Definition 1 (*run length*). Let the run length r_t be a non-negative discrete variable denoting the number of time steps elapsed at time t since the last CP. It holds that $r_t = 0$ at a CP.

The BOCPD [1] generative model assumes that the segments composing a sequence of observations $\mathbf{Y}_{1,T}$ form a product partition [3]. This means that every sequence block $\rho_k = [a_k, b_k]$ (i.e., segments for HSMMs) is generated from a fixed observation model $p(\mathbf{Y}_{\rho_k}|\boldsymbol{\omega}_k) = \prod_{t=a_k}^{b_k} p(\mathbf{y}_t|\boldsymbol{\omega}_k)$. The parameters $\boldsymbol{\omega}_k$ are drawn i.i.d. across segments from some fixed distribution parameterized by $\boldsymbol{\theta}$. Formally, the UPM represents the distribution $p(\mathbf{y}_{t+1}|r_t, \mathbf{Y}_{1,t}, \boldsymbol{\theta})$ which generates observations based on the current run length r_t and observed values $\mathbf{Y}_{1,t}$. Under this model a CP occurs when the underlying generative model changes and a new segment is generated. BOCPD relies on computing the run length posterior $p(r_t|\mathbf{Y}_{1,t})$ to find change points in an online manner. This posterior can be conveniently written in terms of the joint $p(r_t, \mathbf{Y}_{1,t})$, for which the following recursion holds:

$$p(r_t, \mathbf{Y}_{1,t}) = \sum_{r_{t-1}} p(r_t, r_{t-1}\mathbf{Y}_{1,t}) \tag{2}$$

$$= \sum_{r_{t-1}} p(\mathbf{y}_t|r_t, \mathbf{Y}^{r_t}) p(r_t|r_{t-1}) p(r_{t-1}, \mathbf{Y}_{1,t-1}).$$

The predictive distribution for $\mathbf{y}_t$ in (2) is assumed to be independent of the entire history of observations $\mathbf{Y}_{1,t}$ given the subset of observations $\mathbf{Y}^{r_t}$ since the last change point according to the run length r_t. The subset is empty if $r_t = 0$. The CP prior model $p(r_t|r_{t-1})$ is defined through a hazard function $H(r_{t-1}) \in [0,1]$ which parameterizes the two possible outcomes for r_t. Namely, $p(r_t = 0|r_{t-1}) = H(r_{t-1})$ or $p(r_t = r_{t-1} + 1|r_{t-1}) = 1 - H(r_{t-1})$.

Note that the hazard function $H(.)$ implicitly induces a distribution over the duration d of the segments composing an observation sequence. In the HSMM framework this is explicitly modeled through the duration distribution $p(d_t|z_t)$ introduced in Section 2.1. For instance, a constant hazard function $H(r) = c$ induces a geometric distribution over the segment duration d. The latter is indeed the case for the standard HMM.

The joint probability in (2) is not only useful for detecting change points, but can also be used to perform predictions about future observations. The predictive distribution for the next observation (shown in (3)) is derived by marginalizing over the current run length posterior, which is obtained from (2).

$$p(\mathbf{y}_t|\mathbf{Y}_{1,t-1}) = \sum_{r_t} p(\mathbf{y}_t|r_t, \mathbf{Y}^{r_t})p(r_t|\mathbf{Y}_{1,t-1}). \quad (3)$$

2.3 HSMMs vs BOCPD

Note that an HSMM (Section 2.1) with a single hidden state $K = 1$ leads to an equivalent generative model for BOCPD. One of the key differences is that HSMMs are traditionally parameterized in terms of the total segment duration, whereas the BOCPD parameterization is based on the run length through the hazard function instead. The equivalence between the hazard function and the duration model is well-understood—refer to [1] for details. However, the chosen parameterization has relevant implications for the UPMs, as discussed in Section 4.3.

Another key difference between HSMMs and BOCPD is related with how inference is performed and its ultimate goal. HSMMs have been mostly used for batch processing of time series and retrospective inference of hidden states through the celebrated Forward-Backward and Viterbi algorithms [20]. On the other hand, BOCPD focuses on the filtering setting (i.e., online updating) and on events which are relevant for the most recent observation. It does so by explicitly accounting for potentially incomplete segment realizations.

3 RELATED WORK

The original BOCPD model [1] has been extended by adding more structure to the generative model, and new algorithms for learning and online inference have likewise been proposed. [7] and [12] handle different classes of UPMs simultaneously to describe the segments of data between CPs. Furthermore, dependencies across segments can be modeled by assuming Markov structure over either different UPMs [9] or the generative parameters $\boldsymbol{\omega}_k$ [8]. UPMs based on Gaussian processes have been used to model temporal structure within segments [21]. More generally, non-exponential family UPMs can be used within the BOCPD framework without losing the original computational advantages by leveraging variational Bayes approximations [24]. Recently, BOCPD was made robust to outliers using β-Divergences [13]. Regarding hyperparameter learning, the offline [9] and online [25, 21, 4] settings have been considered from an unsupervised perspective. To the best of our knowledge none of these works either address the prediction of CPs

or handle the subsequently introduced duration-dependent emission models (Section 4.3).

More broadly, similar change point and segmentation models have also been used under a Bayesian treatment to model concept drift [2] (i.e., time-varying learning tasks) and as a meta-algorithm for sequence prediction and task-segmentation for piecewise stationary observation models [15].

4 BAYESIAN ONLINE PREDICTION OF CHANGE POINTS

We now extend the BOCPD inference procedure to predict future CP occurrences. First, we show how to perform such inference out-of-the-box in the original BOCPD model. Second, we consider an augmented model with a discrete number of different UPMs with Markovian dynamics (i.e., an HSMM) and extend the online inference procedure for this setting to estimate, in addition to the run length, the total segment duration, and the hidden state accounting for different UPMs. We refer to the augmented inference procedure as segment detection rather than CP detection since we jointly infer both the most recent (past) CP and the next (future) CP, which in turn define the segment enclosing boundaries. Third, we describe a new family of observation models that can be handled with the proposed approach which incorporate the total segment duration (Definition 3) into the emission process. We finally present a learning algorithm based on MLE and discuss computational aspects of the resulting inference procedure.

4.1 RESIDUAL TIME PREDICTION FOR BOCPD

Definition 2 (*residual time*). Let the residual time $l_t \in \mathbb{N} \cup \{0\}$ be the number of remaining observations after time t to reach the end of the segment active at t.

The residual time l_t can be expressed in terms of the run length as the event $(r_{t+1} > 0) \wedge (r_{t+2} > 0) \wedge \cdots \wedge (r_{t+l_t} > 0) \wedge (r_{t+l_t+1} = 0)$. Given observations $\mathbf{Y}_{1,t}$, the posterior over the residual time within the BOCPD framework can be expressed in terms of the run length posterior presented in (2) as

$$p(l_t|\mathbf{Y}_{1,t}) = \sum_{r_t} p(l_t|r_t)p(r_t|\mathbf{Y}_{1,t}), \quad (4)$$

$$\text{with} \quad p(l_t|r_t) = H(r_t + l_t) \prod_{\gamma=r_t}^{r_t+l_t-1} (1 - H(\gamma)),$$

where $H(\cdot)$ denotes the aforementioned hazard function. We highlight that $p(l_t|r_t)$ allows efficient offline compu-

tation since it does not depend on the observations given the run length r_t. Note the assumption of independence of the observation model with respect to the residual time and the segment duration.

It can be shown that for the particular case of a constant hazard $H(r) = c$ Equation 4 becomes $p(l_t|\mathbf{Y}_{1,t}) = p(l_t) = c(1-c)^{l_t}$; i.e., it is independent of the actual observations $\mathbf{Y}_{1,t}$. This suggests that online prediction of CPs is more meaningful for non-constant hazard settings.

4.2 BAYESIAN ONLINE SEGMENT DETECTION

Definition 3 (*segment duration*). Let the segment duration $d_t \in \mathbb{N}$ be the number of observations jointly emitted with the observation at time t.

As in the original formulation (Section 2.2), an observation sequence is composed of non-overlapping segments $\mathbf{Y}_{1,t} = \mathbf{Y}_{\rho_1}, \mathbf{Y}_{\rho_2}, \ldots, \mathbf{Y}_{\rho_S}$. But the parameters $\boldsymbol{\omega}_k$ governing the ρ_k segment are now sampled according to $\boldsymbol{\omega}_k|z_k \sim p(.|\boldsymbol{\theta}_{z_k})$, where the discrete random variable z_k indexes a particular distribution over $\boldsymbol{\omega}_k$. We also constrain the hidden variables $z_1, z_2, \ldots, z_t$ to follow a semi-Markov chain as described in 2.1.

Additionally, we reformulate the model in terms of two extra variables for d_t (Definition 3) and z_t, in addition to the run length r_t defined in Section 2.2.

Our goal is to compute the posterior $p(r_t, d_t, z_t|\mathbf{Y}_{1,t})$ in an online manner. It turns out that we can write a similar recursion to (2) for the joint $\gamma_t = p(r_t, d_t, z_t, \mathbf{Y}_{1,t})$:

$$\gamma_t = p(r_t, d_t, z_t, \mathbf{Y}_{1,t}) \tag{5}$$
$$= \sum_{r_{t-1}} \sum_{d_{t-1}} \sum_{z_{t-1}} p(r_t, d_t, z_t, r_{t-1}, d_{t-1}, z_{t-1}, \mathbf{Y}_{1,t})$$
$$= \sum_{r_{t-1}} \sum_{d_{t-1}} \sum_{z_{t-1}} p(r_t, d_t, z_t, \mathbf{y}_t | r_{t-1}, d_{t-1}, z_{t-1}, \mathbf{Y}_{1,t-1})$$
$$\times p(r_{t-1}, d_{t-1}, z_{t-1}, \mathbf{Y}_{1,t-1})$$
$$= \sum_{r_{t-1}} \sum_{d_{t-1}} \sum_{z_{t-1}} p(\mathbf{y}_t | r_t, d_t, z_t, \mathbf{Y}^{r_t}) \tag{6}$$
$$\times p(r_t, d_t, z_t | r_{t-1}, d_{t-1}, z_{t-1}) \tag{7}$$
$$\times \underbrace{p(r_{t-1}, d_{t-1}, z_{t-1}, \mathbf{Y}_{1,t-1})}_{\gamma_{t-1}}. \tag{8}$$

The UPM factor in (6) accounts for the generative process of observations and only takes into account the relevant observations $\mathbf{Y}^{r_t}$ according to the run length r_t. The factor accounting for the underlying semi-Markov dynamics in the latent discrete process is further expanded in Equation 9 using a similar notation to the one used in explicit-duration modeling of Markov models [6].

According to (5) and (8) γ_t can be written in terms of itself but with a temporal shift of one time step (γ_{t-1}). This fact enables its recursive estimation by processing one observation $\mathbf{y}_t$ at a time and updating the joint in (5) accordingly, i.e., online updating.

$$p(r_t, d_t, z_t | r_{t-1}, d_{t-1}, z_{t-1}) = \overbrace{p(d_t | r_t, z_t, d_{t-1})}^{\text{duration}}$$
$$\times \underbrace{p(z_t | r_t, z_{t-1})}_{\text{transition}} \times \underbrace{p(r_t | r_{t-1}, d_{t-1})}_{\text{run length}}, \tag{9}$$

$$p(d_t | r_t, z_t, d_{t-1}) = \begin{cases} p(d_t | z_t) & \text{if } r_t = 0 \\ \delta(d_t = d_{t-1}) & \text{otherwise,} \end{cases}$$

$$p(z_t | r_t, z_{t-1}) = \begin{cases} p(z_t | z_{t-1}) & \text{if } r_t = 0 \\ \delta(z_t = z_{t-1}) & \text{otherwise,} \end{cases}$$

$$p(r_t | r_{t-1}, d_{t-1}) = \begin{cases} \delta(r_t = 0) & \text{if } r_{t-1} = d_{t-1} - 1 \\ \delta(r_t = r_{t-1} + 1) & \text{otherwise.} \end{cases}$$

The $\delta(x = y)$ function is 1 iff x is equal to y and 0 otherwise. The initial state p.m.f. $p(z_1)$ is omitted for the sake of brevity but must be taken into account at the moment of defining the recursion base case γ_0.

The desired posterior distribution $p(r_t, d_t, z_t | \mathbf{Y}_{1,t})$ can be computed from (5) using the fact that $p(r_t, d_t, z_t | \mathbf{Y}_{1,t}) \propto p(r_t, d_t, z_t, \mathbf{Y}_{1,t})$. Given that r_t, d_t and z_t are all discrete quantities, the posterior normalization constant is computed straightforwardly as $c = p(\mathbf{Y}_{1,t}) = \sum_{r_t, d_t, z_t} p(r_t, d_t, z_t, \mathbf{Y}_{1,t})$. A similar recursion to (5) for $p(r_t, d_t, z_t | \mathbf{Y}_{1,t})$ can also be derived. The latter recursion behaves better in terms of numerical stability given that it is a p.m.f.; therefore, the probability mass is distributed over a discrete set, as opposed to the joint γ_t. This scaling procedure is well-known in the HMM literature [20].

Prediction for future observations is performed similarly to the BOCPD prediction in (3), but marginalizing over the new latent variables, in addition to the run length.

$$p(\mathbf{y}_t | \mathbf{Y}_{1,t-1}) = \sum_{r_t} \sum_{d_t} \sum_{z_t} p(\mathbf{y}_t | r_t, d_t, z_t, \mathbf{Y}^{r_t})$$
$$\times p(r_t, d_t, z_t | \mathbf{Y}_{1,t-1}). \tag{10}$$

4.3 DURATION DEPENDENT UNDERLYING PREDICTIVE MODELS

There are fundamental differences between the UPM contribution $p(\mathbf{y}_t | r_t, d_t, z_t, \mathbf{Y}^{r_t})$ derived in Equation 5) and $p(\mathbf{y}_t | r_t, \mathbf{Y}^{r_t})$, which is the analogous expression for BOCPD (Equation 2). Apart from the fact that we now account for a discrete number of observation models indexed by z_t, we also handle UPMs which not only depend

on the run length r_t but could also potentially depend on the total segment duration d_t. A dependence on d_t is present, for instance, in temporal settings where the emission process is invariant to different time scales. We give a particular example of such setting in Section 5.1. Notice that the UPM contribution of BOCPD is a special case of ours; i.e., it implicitly assumes a single hidden state and an emission model independent of the total segment duration d_t given the run length r_t.

Whether or not to make the UPM duration dependent is a design choice that must ultimately be informed by the data of the task at hand. We propose to use the likelihood as decision criteria; the duration is incorporated into the UPM if it leads to a higher marginal likelihood or improves the likelihood of a predictive task of interest (e.g., residual time prediction).

4.4 HYPERPARAMETER LEARNING

The described algorithm for change point online inference relies on the transition matrix $\mathbf{A}$, the duration matrix $\mathbf{D}$, observation model hyperparameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_z\}_{z=1}^{K}$ and initial state p.m.f. $\boldsymbol{\pi}$ (Section 2.1). Domain knowledge might be available to choose these hyperparameters, but it is usually limited for most real-world applications. We adopt a data-driven approach by using labeled observation sequences where the change point locations have been marked. We define an equivalent joint encoding of the variables r_t, d_t and z_t in terms of a set of binary random variables $\mathbf{S} = \{s_{i,t,d}\}$, where $s_{i,t,d} = 1$ if and only if the hidden state i generates the segment of observations $\mathbf{Y}_{t-d+1,t}$ (i.e., the segment of length d ending at time step t). The complete model likelihood can then be written as $\log p(\mathbf{Y}, \mathbf{S}) = \log p(\mathbf{Y}|\mathbf{S}) + \log p(\mathbf{S})$, where

$$\log p(\mathbf{S}) = \sum_{i,j=1}^{K} \sum_{d,d'=1}^{D} \sum_{t=d'+1}^{T} (s_{i,t-d',d})(s_{j,t,d'}) \log A_{i,j} +$$
$$\sum_{i=1}^{K} \sum_{d=1}^{D} \sum_{t=d}^{T} s_{i,t,d} \log D_{i,d} + \sum_{i=1}^{K} \sum_{d=1}^{D} s_{i,d,d} \log \pi_i,$$
$$(11)$$

$$\log p(\mathbf{Y}|\mathbf{S}) = \sum_{i=1}^{K} \sum_{d=1}^{D} \sum_{t=d}^{T} s_{i,t,d} \log p(\mathbf{Y}_{t-d+1,t}|\boldsymbol{\theta}_i, d),$$
$$(12)$$

and K, D, and T stand for the number of hidden states, maximum duration and sequence length respectively. It

can be shown from Equation (11) that

$$\pi_i^{\star} \propto \sum_{d=1}^{D} s_{i,d,d}, \quad D_{i,d}^{\star} \propto \sum_{t=1}^{T} s_{i,t,d},$$
$$A_{i,j}^{\star} \propto \sum_{t=2}^{T} \sum_{d=1}^{D} \sum_{d'=1}^{D} (s_{i,t-d',d})(s_{j,t,d'}), \quad (13)$$

are maximum-likelihood estimators for $\boldsymbol{\pi}$, $\mathbf{A}$ and $\mathbf{D}$. Similarly, we choose the UPM parameters $\boldsymbol{\theta}$ in such a way that (12) is maximized, which depends on the particular choice of UPM. Note that maximum a posteriori (MAP) estimators can also be derived by assuming suitable priors for the model parameters. We emphasize that having access to a subset of labeled sequences allows us to set the hyperparameters in a principled way.

The complete-data likelihood written in (11) and (12) could be used to estimate the parameters in an unsupervised fashion using the EM algorithm assuming the random variables $\mathbf{S}$ as unobserved. In such case, the estimators in (13) remain unchanged after replacing $s_{i,t,d}$ and $(s_{i,t-d',d})(s_{j,t,d'})$ for their corresponding expected values (i.e., $\mathbb{E}[s_{i,t,d}]$, $\mathbb{E}[(s_{i,t-d',d})(s_{j,t,d'})]$) with respect to its conditional distribution given the current values of the hyperparameters. These conditional expectations can be efficiently computed using forward-backward algorithms for HSMMs (e.g., α and β recursions in [16]).

4.5 COMPUTATIONAL COST

Note that a naive implementation of Equation (5) would be impractical. Using Dynamic Programming over the recursive structure of (5), in a similar way to HSMMs inference algorithms [16], we devised an update algorithm with complexity $\mathcal{O}(K^2 + D^3 K)$ per time-step for arbitrary UPMs. However, if we deal with UPMs whose likelihood function can be calculated incrementally through a set of sufficient statistics (i.e., exponential family likelihoods), then the cost per time-step reduces to $\mathcal{O}(K^2 + D^2 K)$ because $p(\mathbf{y}_t|r_t, d_t, z_t, \mathbf{Y}^{r_t})$ can be evaluated in constant time in such setting leveraging memoization. For a detailed description of the devised algorithms we refer to the Appendix[1].

It turns out that if the UPM is agnostic to the total segment duration $p(\mathbf{y}_t|r_t, d_t, z_t, \mathbf{Y}^{r_t}) = p(\mathbf{y}_t|r_t, z_t, \mathbf{Y}^{r_t})$, then the cost per online update is still $\mathcal{O}(K^2 + D^2 K)$. We leave as an open question whether it is possible to achieve a more efficient update to exactly infer the residual time l_t for this class of UPMs. An example of this type of UPM is presented in Section 5.2.
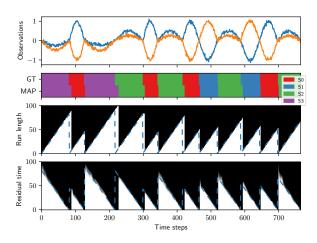
---

[1]Source code and data are available at a public repository: https://github.com/DiegoAE/BOSD
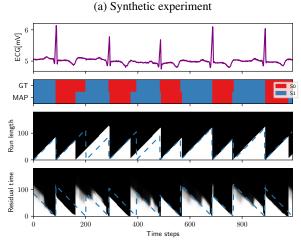
5 RESULTS

5.1 SYNTHETIC EXPERIMENT

We first illustrate the proposed inference procedure using a synthetic 2D data set. The data are generated using an HSMM with 4 hidden states and with some fixed initial p.m.f. $\boldsymbol{\pi}$, transition matrix $\mathbf{A}$, and duration matrix $\mathbf{D}$. The observation model for a hidden state k and for a particular segment duration of $d+1$ observations has the form $\mathbf{y}_t = (b_k \sin(t/d), c_k \sin(t/d)) + \boldsymbol{\epsilon}, \forall t \in \{0, 1, \ldots, d\}$, where $\{b_k, c_k\}$ denote state-specific constants and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ accounts for output noise with fixed variance σ^2. Note that the sine functions input t/d is always in the interval $[0, 1]$ regardless of the total segment duration. This constraint encodes the property that realizations of the same hidden state with different length share the same functional shape. However, those trajectories might exhibit different time scales (e.g., linear time warping). This is an example of an observation model which jointly depends on the hidden state and the total segment duration, as described in Section 4.3. A trajectory sampled from the described model is depicted in the top row of Figure 1a.

Note the alignment between the ground truth labels (GT) and the inferred most likely hidden state sequence (MAP) in the second row of Figure 1a. The run length and residual time inference are also plotted alongside. Remarkably, the run length inference has almost no uncertainty as it can be seen from the sudden color change from white (0 CDF mass) to black (1 CDF mass) around the blue-dashed line depicting the true run length. This high confidence occurs as a consequence of using the actual generative models to process the synthetic observations. The run length is sometimes overestimated (e.g., around time step 300), but it gets updated to the right value after observing the first few observations of a new segment.

The residual time posterior is also fairly confident but exhibits higher uncertainty during the first observations of a new segment (e.g., time step 700). However, note that it gets more confident after seeing a few more observations. This is a consequence of the observations being dependent on the total duration of the segment they belong to. Intuitively, the very first observations in a segment can give insights about its time scale which in turn reduces the uncertainty about the remaining time until the next CP. In both cases, the inferred posteriors are consistent with the true functions of run length and residual time.

(a) Synthetic experiment

(b) ECG segmentation

Figure 1: Inferences drawn from the proposed method in two different tasks. Both plots share the same structure where the **1st row** corresponds to the observations. The **2nd row** depicts the ground truth (GT) sequence of hidden states aligned with the sequence of states which maximizes the marginal filtering distribution (MAP)—each state is represented with a different color. **3rd and 4th rows**: the marginal run length CDF and the marginal residual time CDF are depicted as column vectors (white denotes 0 and black denotes 1). The corresponding true functions are plotted along as blue-dashed lines. Note how the true run length and remaining time functions lie in regions of high probability of the CDFs. In contrast to the experiment in Figure 2, the shown residual time posteriors are less uncertain after seeing a few observations of a segment thanks to the duration-dependent UPM. We refer to the Appendix for larger visualizations of all plots.
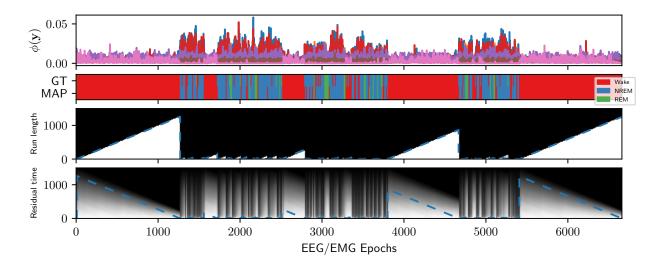
Figure 2: Mice sleep staging through Bayesian online segment detection. **First row**: shows frequency-based EEG/EMG features $\phi(\mathbf{y}_t)$ from a validation subject. **Second row**: the sequence of sleep states provided by an expert (GT) is aligned with the sequence of states which maximizes the marginal filtering distribution $p(z_t|\mathbf{Y}_{1,t})$ (MAP). The considered sleep states are wake (red), REM (green) and NREM (blue). We recover the GT with an average F1 score of 0.91 (See Table 1). In the **3rd and 4th** rows the marginal run length and marginal residual-time CDFs are plotted as columns for each epoch along ground truth functions (blue-dashed lines).

5.2 SLEEP CLASSIFICATION FROM EEG/EMG DATA

Sleep staging is a fundamental but labor intensive process and numerous sleep stage classification schemes have been proposed [19, 23]. Reliable and *online* sleep scoring in combination with stage dependent interventions potentially allows for acoustic stimulation [17], which has been shown to have positive effects, e.g., for declarative memory. However, interventions require the phase tracking of raw EEG signals in real-time, which is in general a difficult task [19].

In particular, sleep staging requires identifying different sleep states from time-varying measurements; typically electroencephalography (EEG) and electromyography (EMG). In mice, sleep researchers are usually interested in distinguishing among the following states: rapid eye movement sleep (REM), non-rapid eye movement sleep (NREM) and the wake status. The standard approach to stage the EEG/EMG time series is based on frequency-band features, and it has been traditionally applied by inspection of well-trained experts.

We consider a data set with three-channel measurements (2 EEG and 1 EMG) of 3 mice recorded at a frequency of 128Hz for 24h. The data are split into 4-second epochs yielding a sequence of 21600 epochs per subject. We use one mouse as test subject and the other two and their respective labels to train the model in a supervised fash-

ion using maximum likelihood estimation (MLE) (See Section 4.4).

We test our method with a maximum duration $D = 1500$ and with $K = 3$ hidden states. We choose an observation model which does not depend on the total segment duration for computational reasons given that it allows to handle large segment durations, as explained in Section 4.5. In particular, we assume the likelihood for a particular epoch $\mathbf{y}_t$ to be $p(\mathbf{y}_t|z_t, \phi) = \mathcal{N}(\phi(\mathbf{y}_t)|\boldsymbol{\mu}^{z_t}, \boldsymbol{\Sigma}^{z_t})$, where $\phi(.)$ denotes a frequency-band feature mapping and z_t denotes the corresponding hidden state. As features we use the amplitudes present in different frequency bands similarly to [23].

Table 1: Mice sleep state classification evaluation for the proposed method (BOSD) and FASTER.

	BOSD			FASTER		
STATE	PRECISION	RECALL	F1	PRECISION	RECALL	F1
WAKE	0.94	0.93	0.93	1.00	0.83	0.91
REM	0.90	0.91	0.91	0.76	0.98	0.86
NREM	0.81	0.87	0.84	0.57	0.37	0.45
AVG	0.91	0.91	0.91	0.87	0.86	0.85

In Table 1, we compare the sleep state classification performance with respect to FASTER [23], which is a fully-automated sleep staging methods designed for mice recordings. Notice that the occurrence of different sleep states is not balanced. For instance, the test sequence shown in Figure 2 has 11240 epochs labeled as wake,

8653 as NREM and only 1707 as REM. According to our experiments, FASTER does not perform well identifying the state NREM (F1 = 0.45), whereas our method gets a much higher score (F1 = 0.84) for the same class. Note that our algorithm performed online inference while FASTER considered the significantly easier offline case.

In the last two rows of Figure 2 we show the posterior CDFs over the run length and the residual time derived from the central object of our method $p(r_t, l_t, z_t | \mathbf{Y}_{1,t})$. In both cases, the probability mass is represented by a gray scale ranging from white (0 mass) to black (1 mass). The obtained run length inference is clearly much more confident than the corresponding residual time inference given the sharp pattern of its probabilities as opposed to the shadowed pattern obtained in the last row of Figure 2. This is mostly due to the fact that the residual time prediction is inherently a predictive task, whereas the run length estimation accounts for an event that has already happened and from which there must be more evidence about. The run length estimation tends to be particularly accurate for long segments as shown by the inference performed over the consecutive runs of the red color (wake state).

Regarding the residual time prediction, we highlight that the confidence fades out in a way that is consistent with the ground truth. Consider the residual time posterior for the three longest segments of the wake state and notice how the range of plausible values is much more constrained at the end of such segments. In fact, the true remaining time function is within 2 standard deviations of the expected residual time for all the considered epochs (not shown).

In contrast with the synthetic experiment, the observations themselves can not provide much insight about the residual time in this case because in this experiment the emission process does not depend on the total segment duration. This is why we do not get a more peaked posterior over the remaining time after seeing the first few observations of a new segment.

5.3 ECG ANALYSIS

The measurement of the duration cycle of ECG (Electrocardiogram) waves is a useful indicator to determine abnormalities (e.g., cardiac arrhythmia) in the heart function. Moreover, within any complete ECG cycle, there are events (e.g., R peak) which encode different stages of the heart cycle, and their locations are useful markers for diagnosing heart rhythm problems, myocardial damage and even in the drug development process [11]. We use one of the labeled ECG recordings (sel100) of the QT database [14] to showcase joint inference over the heart cycle stages (Diastole and Systole), the elapsed time on them, and the remaining time until the next stage transition in Figure 1b.

We leverage two properties of the ECG waves to fully exploit our inference procedure. Namely, bell-shaped duration models and consistent temporal patterns associated with the different heart stages. Furthermore, the functional shape of such patterns exhibit time scaling. This motivates the use of an UPM that takes the total duration into account, as explained in Section 4.3.

Formally, we model the ECG signal y_t in terms of a vector of N basis functions $\phi(x)$ (parameterized as a neural network) and their corresponding weights $\boldsymbol{\omega}$, with i.i.d. Gaussian noise added. In addition, a conjugate prior is assumed over the weights $\boldsymbol{\omega} \sim \mathcal{N}(\boldsymbol{\mu}^{z_t}, \boldsymbol{\Sigma}^{z_t})$, whose mean and covariance depend on the particular hidden state z_t associated with the segment. Given the run length r_t, the segment hidden state z_t and the total stage duration d_t, the observation model takes the form: $y_t = \phi(r_t/d_t)^\top \boldsymbol{\omega} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_{z_t}^2)$.

Note that the basis function vector is a function of the ratio $x_t = r_t/d_t$, therefore $0 \leq x_t < 1$ holds. Intuitively, this means that the shape of the functions generated by this emission process can be temporally stretched, and they can still be well represented by the same underlying model. A similar model is known in the robotics community as Probabilistic Movement Primitives [18].

In the top row of Figure 1b we depict the ECG waves for the last 5 heart cycles of the subject sequence under study. We use the first part of the sequence with its corresponding annotations to learn the basis functions $\phi(\cdot)$, the observation model parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}^{z_t}, \boldsymbol{\Sigma}^{z_t}, \sigma_{z_t}\}$, the transition matrix ($\mathbf{A}$) and the duration matrix ($\mathbf{D}$) in a supervised manner (Section 4.4). In the second row of Figure 1b, we show the segmentation resulting from the annotations (GT) on top of the inferred one (MAP) by maximizing the marginal state posterior. Regarding classification performance, we get precision $= 0.99$ and recall $= 0.81$ ($F1 = 0.89$) for S0 and precision $= 0.84$ and recall $= 0.99$ for S1 ($F1 = 0.91$). This shows that our method could be used to stage ECG waves in addition to detecting and predicting their change points. In the third and fourth rows, we illustrate the resulting inference over the run length and residual time after observing each ECG measurement y_t. The inference is consistent with the ground truth values (blue-dashed line) except for some segments located around the time step 100 and time step 300.

We emphasize that the used UPM depends on the total segment duration as opposed to the one used for the EEG in Section 5.2. The main consequence of this is that, in

the former case, the observations directly influence the residual time estimate, while in the latter case this happens only indirectly via the run length posterior. The residual time inference of the first S1 segment in Figure 1b serves as an example to show how, after observing the initial part of a segment, the uncertainty reduces abruptly around the true values. Comparing the last rows of Figures 2 and 1b, it is clear that in the duration agnostic case the estimates are more conservative, i.e., they have higher uncertainty, as opposed to the posteriors obtained in the duration dependent case where the inferred posteriors reduce their uncertainty early on in the segments.

6 CONCLUSIONS

We extend the Bayesian Online Change Point Detection algorithm to infer when the next change point will occur in a stream of observations. We further generalize the inference procedure to perform Bayesian Online Segment Detection (i.e., joint inference of enclosing CPs) which leads to a new family of observation models that were not supported so far: UPMs that depend on the total segment duration in addition to the run length. We illustrate the method using different observation models, with synthetic and real world medical data sets. We emphasize that the modularity of the proposed method enables its composition with existing extensions of BOCPD and more flexible UPMs, which we leave for future work together with strategies to speed up inference.

Acknowledgements

References

[1] R. P. Adams and D. J. MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

[2] S. Bach and M. Maloof. A bayesian approach to concept drift. In *Advances in neural information processing systems*, pages 127–135, 2010.

[3] D. Barry and J. A. Hartigan. A bayesian analysis for change point problems. *Journal of the American Statistical Association*, 88(421):309–319, 1993.

[4] F. Caron, A. Doucet, and R. Gottardo. On-line changepoint detection and parameter estimation with application to genomic data. *Statistics and Computing*, 22(2):579–595, 2012.

[5] S. Chiappa. Explicit-duration Markov switching models. *Foundations and Trends in Machine Learning*, 7(6):803–886, 2014.

[6] S. Chiappa and J. R. Peters. Movement extraction by detecting dynamics switches and repetitions. In *Advances in Neural Information Processing Systems*, pages 388–396, 2010.

[7] P. Fearnhead and Z. Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605, 2007.

[8] P. Fearnhead and Z. Liu. Efficient bayesian analysis of multiple changepoint models with dependence across segments. *Statistics and Computing*, 21(2):217–229, 2011.

[9] P. Fearnhead and D. Vasileiou. Bayesian analysis of isochores. *Journal of the American Statistical Association*, 104(485):132–141, 2009.

[10] Z. Harchaoui and O. Cappé. Retrospective mutiple change-point estimation with kernels. In *Statistical Signal Processing*, pages 768–772, 2007.

[11] N. P. Hughes, L. Tarassenko, and S. J. Roberts. Markov models for automated ECG interval analysis. In *Advances in Neural Information Processing Systems*, pages 611–618, 2004.

[12] J. Knoblauch and T. Damoulas. Spatio-temporal Bayesian on-line changepoint detection with model selection. In *Proceedings of the 35th International Conference on Machine Learning (ICML-18)*, pages 2718–2727, 2018.

[13] J. Knoblauch, J. E. Jewson, and T. Damoulas. Doubly robust bayesian inference for non-stationary streaming data with β-divergences. In *Advances in Neural Information Processing Systems*, pages 64–75, 2018.

[14] P. Laguna, R. G. Mark, A. Goldberg, and G. B. Moody. A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG. In *Computers in cardiology 1997*, pages 673–676. IEEE, 1997.

[15] K. Milan, J. Veness, J. Kirkpatrick, M. Bowling, A. Koop, and D. Hassabis. The forget-me-not process. In *Advances in Neural Information Processing Systems*, pages 3702–3710, 2016.

[16] K. P. Murphy. Hidden semi-Markov models (HSMMs). Technical report, 2002.

[17] H.-V. V. Ngo, T. Martinetz, J. Born, and M. Mölle. Auditory closed-loop stimulation of the sleep slow oscillation enhances memory. *Neuron*, 78(3):545–553, 2013.

[18] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann. Probabilistic movement primitives. In *Advances in Neural Information Processing Systems*, pages 2616–2624, 2013.

[19] A. Patanaik, J. L. Ong, J. J. Gooley, S. Ancoli-Israel, and M. W. Chee. An end-to-end framework for real-time automatic sleep stage classification. *Sleep*, 2018.

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

[21] Y. Saatçi, R. D. Turner, and C. E. Rasmussen. Gaussian process change point models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 927–934, 2010.

[22] D. Stephens. Bayesian retrospective multiple-changepoint identification. *Applied Statistics*, pages 159–178, 1994.

[23] G. A. Sunagawa, H. Séi, S. Shimba, Y. Urade, and H. R. Ueda. Faster: an unsupervised fully automated sleep staging method for mice. *Genes to Cells*, 18(6):502–518, 2013.

[24] R. D. Turner, S. Bottone, and C. J. Stanek. Online variational approximations to non-exponential family change point models: with application to radar tracking. In *Advances in Neural Information Processing Systems*, pages 306–314, 2013.

[25] R. C. Wilson, M. R. Nassar, and J. I. Gold. Bayesian online learning of the hazard rate in change-point problems. *Neural computation*, 22(9):2452–2476, 2010.