Learning Time Series Segmentation Models from Temporally Imprecise Labels

Roy J. Adams and **Benjamin M. Marlin**
University of Massachusetts, Amherst

Abstract

This paper considers the problem of learning time series segmentation models when the labeled data are subject to temporal uncertainty or noise. Our approach augments the semi-Markov conditional random field (semi-CRF) model with a probabilistic model of the label observation process. This augmentation allows us to estimate the parameters of the semi-CRF from timestamps corresponding roughly to the occurrence of transitions between segments. We show how exact marginal inference can be performed in the augmented model in polynomial time, enabling learning based on marginal likelihood maximization. Our experiments on two activity detection problems show that the proposed approach can learn models from temporally imprecise labels, and can successfully refine imprecise segmentations through posterior inference. Finally, we show how inference complexity can be reduced by a factor of 40 using static and model-based pruning of the inference dynamic program.

1 INDRODUCTION

Structured prediction frameworks (e.g. conditional random fields [7]) are well-established approaches that often improve on independent prediction models when applied to problems with structured output spaces. In this paper, we consider the problem of learning and inference in structured prediction models for time series segmentation when the labels are subject to temporal imprecision. In this setting, supervision is provided in the form of timestamps that roughly correspond to segment boundaries.

This problem is an instance of weakly supervised learning [5] motivated by real-world data analytic challenges that arise in the area of mobile health (mHealth) research. A central problem in mHealth research is learning accurate models for detecting health behaviors like eating, smoking, and sleeping from mobile sensor data [14, 18]. A key challenge in such problems is the high cost of obtaining accurately annotated data. mHealth researchers often must estimate the parameters of detection models from limited amounts of data collected in a lab setting where subjects perform scripted activities. This collection process allows researchers to observe subject behavior in detail, but severely limits the amount of data that can be gathered. Further, such data can differ systematically from data collected under real-world scenarios, leading to a lab-to-field generalization gap [10].

An alternative to gathering data in the lab is having subjects self-report their activities in the field, but this suffers from a variety of problems from a modeling perspective including limited frequency of the reports and recall bias. In both lab and field settings, annotations are generally provided in continuous time and may be subject to temporal imprecision. If ignored, such temporal imprecision in the labels may lead to the degraded performance of models trained on these labels. This is the primary problem we address in this paper.

This work makes three main contributions. First, we propose a framework for estimating the parameters of discrete time series segmentation models from temporally imprecise, continuous-time labels. Our approach augments a conditional random field (CRF) model with a probabilistic model of the label observation process. We focus on two classes of CRF models: the semi-Markov CRF (semi-CRF) [15], which models sequence segmentations, and the *hierarchical nested segmentation* (HNS) model [2], an extension of the semi-CRF that models activities composed of repeated short duration events such as eating. Our proposed observation model can account for both temporal imprecision and missingness in the labels. We evaluate this framework on sleep and smoking detection problems using data gathered in both lab

and field settings, demonstrating improved performance compared with ignoring label imprecision.

Second, we enable the synthesis of self-report and wearable sensor data. To the best of our knowledge, current methods for synthesizing these two types of observations are ad hoc and domain specific (e.g. [11]). In this work, we combine sensor data with imprecise continuous-time observations of activity segment boundaries by performing posterior inference in the proposed observation model. This leads to improved predictive performance over treating test-time observations as ground truth.

Finally, we enable the practical application of the proposed framework to long sequences. The model that we present supports exact inference via dynamic programming, but the complexity scales quadratically in the length of the input sequence. We achieve a 40 times speedup by applying a combination of static and model-based pruning techniques, while matching the performance of a model trained on hand-aligned labels.

2 RELATED WORK

In this section, we briefly describe related work on weakly supervised learning in the independent classification and structured prediction settings.

Weakly supervised classification: Reducing the cost of acquiring labeled data is a fundamental problem in supervised learning. This can often be achieved by lowering the quality of labels in some way. For example, *multiple instance learning* generalizes supervised learning by allowing for sets (or "bags") of instances to be labeled instead of single instances. It is assumed that a positive bag contains at least one positive instance [9]. Similarly, the *label proportions* framework provides the proportion of each type of label for a group of instances [13]. These approaches avoid the need to label individual instances.

More closely related problems include learning independent classifiers in the presence of label noise [6], and learning independent sequence labeling models from temporally imprecise labels [1]. In both of these frameworks, the true instance labels are assumed to be unobserved. In the label noise framework, noisy instances of the labels are observed, while in the temporally imprecise labels framework, timestamps roughly corresponding to positive instances are observed. Approaches to both problems exist that are based on models of the noisy labeling process that marginalize over the unobserved instance labels during learning. The main difference between these models and the model presented in this work is that these models assume the true labels are independent given the features. In this paper, we consider a more complex structured prediction setting, which in turn requires more complex observation models and inference algorithms.

Weakly supervised structured prediction: There has also been significant research in the area of weakly supervised structured prediction, particularly for computer vision applications. Various standard weak supervision frameworks, such as multiple instance learning, have been extended to structured prediction. [16] extend the multiple instance SVM framework to structured SVMs by considering an image to be a bag of pixels or overlapping sub-windows. [4] extend multiple instance learning to an auto-regressive HMM. While applicable to the problem considered in this paper, these methods would require discarding temporal information that was shown to be valuable in [1].

Another common approach is to assume that only a subset of the label variables in the model are exactly observed. This can be handled by marginalizing out the unobserved variables [17]; however, this framework cannot incorporate auxiliary observations such as continuous observation timestamps. [8] incorporate domain knowledge in the form of constraints on the marginal label distributions. These constraints can be enforced on unlabeled data, allowing for weak supervision. [12] use a similar constraint based approach where image tags are used to form constraints on the set of possible image segmentations. The approach in this work can be interpreted as using observation timestamps to place soft constraints on the set of segmentations; however, by using soft constraints, we can explicitly model the notion of temporal proximity.

3 NOTATION AND BACKGROUND

Many mHealth detection problems involve inferring activity segments from sensor data. Past work has shown improved performance when using conditional random field-based structured models to infer such segmentations [2]. We begin by the defining notation used for the input sequences and output structures in this type of problem. We then briefly review the Semi-Markov CRF model that this work extends.

3.1 Notation

We assume that the input data consists of N multivariate time series that we will call **sessions**. Each session contains a set of time-aligned signals gathered from one or more sensors. Seperate sessions may correspond to data from different subjects data or to data from the same subject collected at different times. We assume that each session n has been discretized into a sequence of L_n potentially overlapping sub-windows and that a feature

vector $\mathbf{x}_{ni} \in \mathbb{R}^D$ has been extracted for each sub-window i. We refer to each sub-window i as an **instance**. Further, each instance i in session n is associated with a timestamp t_{ni} which may correspond to the start, end, or other point of interest associated with instance i. We refer to the complete sequence of feature vectors $\mathbf{x}_n = \{\mathbf{x}_{ni}\}_{i=1,...,L_n}$ as the **input sequence** and the complete sequence of timestamps $\mathbf{t}_n = \{t_{ni}\}_{i=1,...,L_n}$ as the **timestamp sequence**. Where it does not cause ambiguity, we will drop the session index n. We use the notation $\mathbf{x}_{j:k} = \{\mathbf{x}_i\}_{i=j,...,k}$ to refer to the subsequence of $\mathbf{x}$ beginning at j and ending at k (this applies to any sequence).

In this work, our goal is to learn a model that produces a labeled segmentation of the input sequence $\mathbf{x}$. We represent such a segmentation as a sequence $\mathbf{y} = \{y_s\}_{s=1,...,S}$ of segments where a segment $y_s = (c_s, j_s, k_s)$ is a tuple containing a label $c_s \in \mathcal{C}$, a start position $j_s \in \{1, ..., L\}$, and an end position $k_s \in \{1, ..., L\}$. To ensure only valid segmentations, we assume $j_1 = 1$, $k_S = L$, and $k_s = j_{s+1}$ for all $1 \leq s \leq S - 1$. Our goal, then, is to learn the distribution $p(\mathbf{y}|\mathbf{x}, \mathbf{t})$. We will parameterize this distribution as a semi-Markov CRF.

3.2 Semi-Markov Conditional Random Fields

The semi-CRF [15] associates each segment y_s with a feature function $\mathbf{f}(y_s, c_{s-1}, \mathbf{x}, \mathbf{t})$ which may depend on the segment y_s, the label of the previous segment c_{s-1}, and the complete feature and timestamp sequences $\mathbf{x}$ and $\mathbf{t}$. The function f maps these inputs to a length F feature vector. Given a parameter vector $\theta \in \mathbb{R}^F$, the distribution over segmentations is given by

$$p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{t}) = \frac{\prod_s \exp\left(\langle \theta, \mathbf{f}(y_s, c_{s-1}, \mathbf{x}, \mathbf{t})\rangle\right)}{Z_\theta(\mathbf{x}, \mathbf{t})} \quad (1)$$

Both maximum a posteriori (MAP) and marginal inference can be performed in this model by dynamic programs with complexity $\mathcal{O}(|\mathcal{C}|^2 L^2)$ [15]. The parameters θ are typically estimated using maximum likelihood estimation, however, this requires observing the ground truth segmentations. In settings such as mHealth, acquiring the exact segmentation boundaries may be costly or even impossible. We next present our proposed method for estimating the parameters of the semi-CRF model from timestamps corresponding roughly to segment boundaries.

4 LEARNING SEMI-CRF MODELS FROM TEMPORALLY IMPRECISE LABELS

Let $\mathbf{z} = \{z_m\}_{m=1,...,M}$ be a sequence of **observations** where each observation z_m is a timestamp corresponding to a particular kind of transition. For example, each z_m may be the time a subject reported going to sleep marking the start of a sleep segment. For ease of exposition, we will assume that there is only one type of observation and will later generalize to multiple observation types. To map between our labels $\mathbf{y}$ and our observations $\mathbf{z}$, let $\mathbf{o} = \{o_i\}_{i=1,...,L}$ be a sequence of latent binary variables where $o_i = 1$ if and only if instance i is associated with an observation. Under the assumption that observations are recorded in the order they actually occurred and $\sum_i o_i = M$, $\mathbf{o}$ defines a matching between instances in the input sequence and observations in the observation sequence.

We model the observation sequence using a generative model with three components. The base segmentation model $p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{t})$ is the semi-CRF model whose parameters we are interested in estimating. The observation indicator distribution $p_\pi(o|y_s, c_{s-1}, i)$ models the probability that instance i is associated with an observation given the segment it is contained in and the label of the previous segment. Finally, the observation timestamp density $p_\phi(z|t)$ models the timestamp of an observation z given the timestamps t with which it is associated. For example, we may use a simple Bernoulli distribution for $p_\pi(o|y_s, c_{s-1}, i)$ and a normal distribution centered at t for $p_\phi(z|t)$. The specific choices for these distributions are domain specific and we demonstrate a couple different choices in section 5. With these distributions, we can now write the observation generation process as shown below:

$M \leftarrow 0$
$\mathbf{y} \sim p_\theta(\mathbf{y}|\mathbf{x})$
for $s = 1, ..., S$ **do**
  for $i = j_s, ..., k_s$ **do**
    $o_i \sim p_\pi(o|y_s, c_{s-1}, i)$
    if $o_i = 1$ **then**
      $M \leftarrow M + 1$
      $z_M \sim p_\phi(z|t_i)$

This generative process asserts that a complete segmentation is first sampled according to the semi-CRF model. Next, each instance either generates an observation or not according to $p_\pi(o|y_s, c_{s-1}, i)$. Finally, if instance i does generate an observation, an observation timestamp is sampled from $p_\phi(z|t_i)$. The variable M counts the number of generated observations. We note that additional structure could be encoded into the label observation process at the cost of a potentially more complex inference algorithm.

$$p_\omega(\mathbf{z}, \mathbf{y}, \mathbf{o}|\mathbf{x}, \mathbf{t}) = p_\theta(\mathbf{y}|\mathbf{x}) p_\pi(\mathbf{o}|\mathbf{y}) p_\phi(\mathbf{z}|\mathbf{o}, \mathbf{t}) \quad (2)$$

$$p_\pi(\mathbf{o}|\mathbf{y}) = \prod_s \prod_{i=j_s}^{k_s} p_\pi(o_i|y_s, c_{s-1}, i) \quad (3)$$

$$p_\phi(\mathbf{z}|\mathbf{o}, \mathbf{t}) = \prod_{m=1}^{M} p_\phi(z_m|t_{i(m)}) \quad (4)$$

The joint model implied by this generative process is

given in Equation 2 where the set of all parameters in the model is $\omega = \{\theta, \pi, \phi\}$. The distributions $p_\pi(\mathbf{o}|\mathbf{y})$ and $p_\phi(\mathbf{z}|\mathbf{o}, \mathbf{t})$ are defined in Equations 3 and 4. We define $i(m)$ as the function mapping observation m to the instance that generated it.

4.1 Inference and Learning

To learn the parameters of this model, we maximize the log marginal likelihood $\mathcal{L}(\omega|\mathcal{D})$:

$$\mathcal{L}(\omega|\mathcal{D}) = \sum_{n=1}^{N} \log p_\omega(\mathbf{z}_n|\mathbf{x}_n, \mathbf{t}_n) \tag{5}$$

$$p_\omega(\mathbf{z}|\mathbf{x}, \mathbf{t}) = \sum_{\mathbf{y}\in\mathcal{Y}} \sum_{\mathbf{o}\in\mathcal{O}} p_\omega(\mathbf{z}, \mathbf{y}, \mathbf{o}|\mathbf{x}, \mathbf{t}) \tag{6}$$

where $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{t}_n, \mathbf{z}_n)\}_{n=1,\ldots,N}$ consists of the observed data for all sessions. We perform this optimization using standard gradient methods. Here, we consider the gradient equation for each of the three parameter groups: θ, π, and ϕ. These equations are presented primarily to give intuition for what maximum likelihood estimation is doing in this model. The gradient equations for π and ϕ are shown below.

$$\nabla_\phi \log p_\omega(\mathbf{z}|\mathbf{x}, \mathbf{t}) = \sum_{m=1}^{M} \mathbb{E}_{p_\omega(i(m)|\mathbf{z},\mathbf{x},\mathbf{t})} \left[\nabla_\phi \log p_\phi(z_m|t_{i(m)})\right] \tag{7}$$

$$\nabla_\pi \log p_\omega(\mathbf{z}|\mathbf{x}, \mathbf{t}) = \sum_{i=1}^{L} \mathbb{E}_{p_\omega(o_i,\mathbf{y}|\mathbf{z},\mathbf{x},\mathbf{t})} \left[\nabla_\pi \log p_\pi(o_i|\mathbf{y})\right] \tag{8}$$

Both gradient equations take the form of a posterior expectation of the log gradient of the relevant distribution. The gradient with respect to the base classifier parameters also takes the form of an expected gradient of a log density and is shown below.

$$\begin{aligned}\nabla_\theta \log\, p_\omega(\mathbf{z}|\mathbf{x}, \mathbf{t}) &= \mathbb{E}_{p_\omega(\mathbf{y}|\mathbf{z},\mathbf{x},\mathbf{t})} \left[\nabla_\theta \log p_\theta(\mathbf{y}|\mathbf{x})\right] \\ &= \mathbb{E}_{p_\omega(\mathbf{y}|\mathbf{z},\mathbf{x},\mathbf{t})} \left[\nabla_\theta \langle\theta, \mathbf{f}(\mathbf{x}, \mathbf{t}, \mathbf{y})\rangle\right] - \nabla_\theta Z_\theta(\mathbf{x}) \\ &= \mathbb{E}_{p_\omega(\mathbf{y}|\mathbf{z},\mathbf{x},\mathbf{t})} \left[\mathbf{f}(\mathbf{x}, \mathbf{t}, \mathbf{y})\right] - \mathbb{E}_{p_\theta(\mathbf{y}|\mathbf{x})} \left[\mathbf{f}(\mathbf{x}, \mathbf{t}, \mathbf{y})\right]\end{aligned} \tag{9}$$

where $\mathbf{f}(\mathbf{x}, \mathbf{t}, \mathbf{y})$ denotes the sufficient statistics function for the semi-CRF model. In this case, the log-linear form of the semi-CRF model gives us the further interpretation that the learning algorithm is trying to match the expected sufficient statistics under the base semi-CRF model to the posterior expected sufficient statistics given by the observation model. This is in contrast to typical maximum likelihood estimation for a log-linear model which would match the expected sufficient statistics under the model to the observed sufficient statistics.

The primary computational challenge of this learning procedure is calculating the log marginal likelihood. This can be done exactly using a dynamic program for calculating $p_\omega(\mathbf{z}|\mathbf{x}, \mathbf{t})$. An entry in the dynamic programming table α has the following interpretation: $\alpha(k, c, m)$ is the unnormalized probability that the input subsequence $\mathbf{x}_{1:k}$ generated the observation subsequence $\mathbf{z}_{1:m}$ given that the last segment in $\mathbf{y}$ has label c. Or, written mathematically:

$$\alpha(k, c, m) \propto p_\omega(\mathbf{z}_{1:m}|\mathbf{x}_{1:k}, \mathbf{t}_{1:k}, c_{|\mathbf{y}|} = c) \tag{10}$$

Filling in this table has complexity $\mathcal{O}(|\mathcal{C}|^2 L^2 M)$ where L is the length of the input sequence, $\mathcal{C}$ is the set of possible segment labels, and M is the length of the observation sequence. A full description of this dynamic program can be found in section 2 of the supplementary materials. We use reverse-mode automatic differentiation [3] to derive a dynamic program with the same complexity to calculate the necessary gradients for learning.

4.2 MAP Inference

Our second goal is to combine temporal observations, such as self-reported activities, and wearable sensor input to infer behaviors. That is, we would like to infer the most likely segmentation of the input sequence given $\mathbf{x}$, $\mathbf{t}$, and $\mathbf{z}$. To do this, we perform full maximum a posteriori (MAP) inference over both $\mathbf{y}$ and $\mathbf{o}$

$$\mathbf{y}^*, \mathbf{o}^* = \arg\max_{\mathbf{y},\mathbf{o}} p_\omega(\mathbf{y}, \mathbf{o}|\mathbf{z}, \mathbf{x}, \mathbf{t}) \tag{11}$$

The same dynamic program used to calculate the marginal likelihood can be used to perform MAP inference by swapping summation over $\mathbf{y}$ and $\mathbf{o}$ for maximization with no change in the computational complexity.

4.3 Multiple Observation Types

In some settings, it may be desirable to allow for multiple types of observations. For example, we may want to include observations of both the beginning and end of sleep. This can be handled by including multiple observation sequences $\mathbf{z}^{(l)}$ each with length $M^{(l)}$ and observation indicator sequences $\mathbf{o}^{(l)}$ where l indicates the observation type. Observation sequences of each type are assumed to be independent conditioned on the segmentation $\mathbf{y}$ and the ordering assumption need not hold between types. The complexity of inference in this setup is $\mathcal{O}(|\mathcal{C}|^2 L^2 \prod_l M^{(l)})$.

5 EXPERIMENTS AND RESULTS

We evaluated the proposed framework's ability to accommodate the temporal label imprecision that arises in both the lab and field settings on two mHealth detection problems: sleep detection and cigarette smoking detection. In

this section we describe the datasets and models used as well as the results of these evaluations.

5.1 Sleep detection

We evaluated our framework's performance on data from the field using the Extrasensory[1] dataset [18]. This dataset contains signals from a variety of sensors including the accelerometer, gyroscope, GPS, and microphone on a mobile device as well as a wrist-worn accelerometer. Subjects carried these sensors during daily activities and self-reported a range of activities such as sleeping, eating, and exercising. We focus on the sleep detection problem, as this was one of the more abundantly reported activities. Signals from all sensors were recorded for 20 seconds every minute leading to a natural one minute discretization, which we downsampled to one instance every two minutes in order to run a large number of experiments (a 2x downsample results in a 4x inference speedup). We partitioned the data into 24 hour sessions beginning and ending at 2:00pm and dropped any session with less than four hours of recorded data or less than one hour of reported sleep. This resulted in 80 sessions from 28 subjects. While the researchers corrected obvious conflicts in the self-reported activities, there is no ground truth for this data, so we evaluated against the cleaned self-reported sleep. To simulate extra noise in the observation process, we added further synthetic noise (described below) to the observation timestamps.

Instance Features: We used the full set of instance features reported in [18] which include a number of statistical features calculated on the various accelerometer and gyroscope sensors, relative features calculated on the GPS positions, and discrete time-of-day features.

Model: Our goal in the sleep detection problem is to segment the input sequence into periods of sleep and non-sleep. We use a binary semi-CRF with a constraint that consecutive segments may not have the same label. We included as features the sum of all instance level features within segment $x_{jk} = \sum_{i=j}^{k} x_i$ and duration based features $\mathbb{I}[c_m = 1](t_k - t_j)$ and $\mathbb{I}[c_m = 1](t_k - t_j)^2$, which are similar to putting a normal distribution on the duration of sleeping activities[2]. We placed a zero-mean gaussian prior with tuned variance on the parameters of the semi-CRF model (i.e. ℓ_2 regularization).

We included two types of observations: the beginning of sleep $\mathbf{z}^{(1)}$ and the end of sleep $\mathbf{z}^{(2)}$. Because sleep was observed in all sessions, we used a deterministic observation indicator distribution. If instance i is the beginning of a sleep segment, it must generate an observation $z_m^{(1)}$ and likewise for the end of a sleep segment. No other instances may generate observations in this model.

To model the procedure of self-reporting when you go to sleep and when you wake up, we used a one-sided distribution to model the observation timestamp noise. We used the following exponential distributions to model observation timestamp noise:

$$p_\phi(z_m^{(1)}|t_{i(m)}) = \text{Exp}(t_{i(m)} - z_m^{(1)}; \lambda)$$
$$p_\phi(z_m^{(2)}|t_{i(m)}) = \text{Exp}(z_m^{(2)} - t_{i(m)}; \lambda)$$

We placed an inverse-Gamma prior with shape $\alpha = 1$ and scale $\beta = 1$ on λ. We found parameter estimation to be fairly insensitive to changes in the settings of this prior distribution and used informative values for α and β.

Train and Test Procedures: We evaluated performance using a 10-fold cross-validation procedure, where folds were calculated at the session level. The strength of the ℓ_2 regularizer was tuned to maximize instance-level F_1 over a logarithmic grid using a further 9-fold evaluation. This procedure is equivalent to assuming that some of the data has been labeled for tuning purposes. Predictions were evaluated against the self-reported labels.

Experiments: We compared semi-CRF models trained in two ways. First, we trained a semi-CRF model based on a naive alignment defined by mapping each augmented observation to the nearest instance (semi-NV). Second, we trained a semi-CRF model using the proposed weak supervision framework applied to the augmented observations (semi-WS). To test these models under a variety of noise conditions, we augmented the observation timestamps by adding different amounts of exponentially distributed noise and trained both models using these augmented observations. Finally, we tested each model when provided with different amounts of information. At test time, each model was given either all segment start observations (Start), all segment end observations (End), neither observations (None), or both observations (Start+End). Observations were incorporated into semi-WS as described in section 4.2, and were incorporated into semi-NV by mapping the provided observations to the nearest instance and performing MAP inference over the label set constrained to agree with the mapped observations.

The results from these experiments are shown in figure 1. The three plots correspond to models trained and tested on observations augmented with standard deviation $\lambda = 0, 30, 60$ minutes of temporal noise. Within each plot, the performance for each model when conditioned on different amounts of information is shown. In all cases, semi-WS outperforms semi-NV. The performance gap grows both as the standard deviation of the observation

[1] `http://extrasensory.ucsd.edu/`

[2] $\mathbb{I}[\cdot]$ is the indicator function

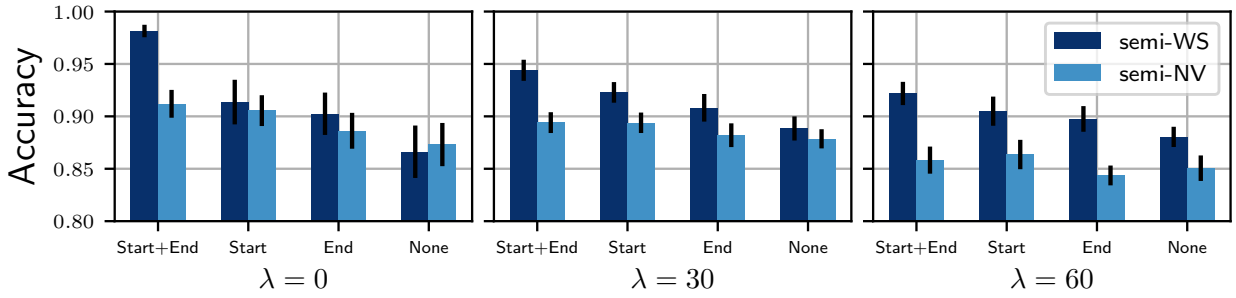


Figure 1: Performance for the semi-WS and semi-NV models on the sleep detection problem when trained on data with Exp(λ) distributed noise (measured in minutes) added to the observation timestamps. Each plot shows the performance of both models when conditioned on all segment start observations (Start), all segment end observations (End), neither (None), or both (Start+End) at test time.

noise increases and as the amount of information available at test time increases. This indicates that semi-WS is better able to learn from temporally imprecise labels, and that using an explicit observation model is useful when incorporating imprecise observations.

5.2 Smoking detection

We evaluated the proposed framework's ability to handle the types of imprecision that arise in a laboratory setting using the puffMarker smoking dataset [14]. This data was collected in a lab setting where subjects were fitted with chest-worn respiration monitors and wrist-worn actigraphy sensors and asked to smoke a cigarette while an observer marked the occurrence of smoking puffs using a mobile phone app. The respiration signal was discretized into a sequence of non-overlapping respiration cycles (a single inhalation and exhalation) and the goal is to label each respiration cycle as a smoking puff or not and segment the respiration cycles into periods of smoking and non-smoking activities. We created sessions by including random amounts of non-smoking on either side of each recorded smoking activity resulting in 23 sessions from five subjects. In addition to the raw observation timestamps, researchers visualized the respiration signal and hand-aligned the observation timestamps to respiration signal. We treat these hand aligned labels as ground truth for the purposes of evaluation, though we acknowledge that there may be errors in the alignment process. While most experiments on this data were conducted using the true observation timestamps, we also tested the robustness of our framework to extra noise which was generated synthetically and added to the raw observation timestamps.

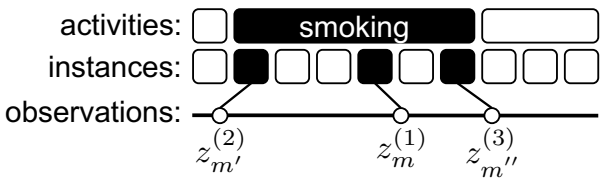
Instance Features: Features were extracted from the respiration monitor data for each respiration cycle according to [14]. Further, we extracted features from the actigraphy data using the following procedure: Let t_i be the timestamp of the maximum peak in respiration cycle i. For each actigraphy channel, extract a window beginning 8 seconds before t_i and ending 1 second after t_i and calculate as features the mean, max, min, standard deviation, median, and five bin histogram of the channel's signal within this window. The actigraphy channels included were accelerometer x, y, and z, accelerometer magnitude, gyroscope x, y, and z, gyroscope magnitude, and pitch and roll angles for a total of 100 actigraphy based features. Pitch and roll calculations using accelerometer data are only valid when the hand is stationary, so these signals were filtered using the procedure described in [14].

Respiration and actigraphy-based features have different properties as a function of time. Due to the method we used to extract actigraphy-based features, these features tend to be smooth through time, particularly as compared to the respiration features extracted from non-overlapping windows. The smooth noise model we propose tends to over-emphasize temporally smooth features at the expense of less smooth features To combat this effect, we use the actigraphy features to augment the respiration features in a manner similar to the filtering approach used in [14].

We trained a logistic regression model on a small set of instances with hand-aligned labels using only the actigraphy features, then took the predictions from this model and augmented the respiration features as $\mathbf{x}_{aug} = [\hat{c}_{act}\mathbf{x}_{resp}\ (1 - \hat{c}_{act})\mathbf{x}_{resp}]$ where $\hat{c}_{act} \in \{0, 1\}$ is a prediction from the filter model and $\mathbf{x}_{resp}$ is the vector containing only respiration features. A similar effect might be achieved by including interaction effects between the actigraphy and respiration features; however, this would result in more than 10,000 features. The filtering approach can therefore also be thought of as first doing a supervised compression of the actigraphy features and then doing a polynomial basis expansion. For more details, see section 3 in the supplemental materials.

Model: Our goal in the smoking detection problem is to label each respiration cycle as smoking or non-smoking

and to segment the input sequence into periods of smoking and non-smoking; however, smoking detection differs from typical segmentation problems in that a complete smoking activity contains a mix of smoking puffs and non-smoking respiration cycles. In terms of the model, this means that the instances contained in a positive segment may be both positive or negative (an example of this structure with example observations is shown below). To address this we used an extension of the standard semi-CRF called the hierarchical nested segmentation (HNS) model [2]. Rather than segment a sequence into positive and negative activities, the HNS model segments the sequence into periods between positive instances, termed inter-event spans. Further, the HNS model includes a cardinality potential that models the number of positive instances that make up a positive activity (or the number of consecutive positive inter-event spans). In addition to the instance level features, we included the segment duration, $t_k - t_j$ and segment duration squared $(t_k - t_j)^2$ to model the time between positive instances (i.e. the time between puffs on a cigarette). For full details of the HNS model for smoking detection, see [2], and for details on how the HNS model can be written as a semi-CRF, see section 4 of the supplementary materials. We placed a zero-mean gaussian prior with tuned variance on the parameters of the HNS model (i.e. ℓ_2 regularization).



As seen in the figure above, we included three types of observations: positive instance observations associated with any inter-event span $\mathbf{z}^{(1)}$, activity start observations associated only with the first inter-event span in a complete activity $\mathbf{z}^{(2)}$, and activity end observations associated only with the last inter-event span in a complete activity $\mathbf{z}^{(3)}$. We used the following Bernoulli distributions for our observation indicator model $p_\pi(o_i^{(l)}|\mathbf{y})$:

$$p_\pi(o_i^{(1)} = 1|\ i \text{ is the start of an inter-event span }) = \pi_1^{(1)}$$
$$p_\pi(o_i^{(2)} = 1|\ i \text{ is the start of a smoking activity}) = \pi_1^{(2)}$$
$$p_\pi(o_i^{(3)} = 1|\ i \text{ is the end of a smoking activity}) = \pi_1^{(2)}$$

where $\pi_1^{(1)}, \pi_1^{(2)} \in [0, 1]$. For the observation timestamp density, we used the following normal distribution:

$$p_\phi(z_m^{(l)}|t_{i(m)}) = \mathcal{N}(z_m^{(l)}; t_{i(m)} + \mu_l, \sigma_l^2)$$

for $l \in \{1, 2, 3\}$ where $\phi = \{\mu, \sigma\}$. This density was chosen to match the empirical noise distribution [1]. We placed a Uniform$(0, 1)$ prior on each $\pi^{(l)}$, a standard normal prior on each μ_l, and an inverse-Gamma prior with shape $\alpha = 1$ and scale $\beta = 1$ on each σ_l^2.

Train and Test Procedures: We evaluated performance using a leave-one-session-out cross-validation procedure. All tuned hyperparameters were tuned to maximize instance level F_1 over a logarithmic grid using a further nested leave-one-session-out evaluation. Predictions were evaluated against the hand-aligned labels.

Experiment 1 - Pruning Strategies: While the inference algorithm described in section 4.1 is at most quadratic in the size of each input, the overall run time can be quite high, particularly for long sequences or models with a large label set $\mathcal{C}$ such as the HNS model. In order to improve inference run times, we consider three strategies to prune the inference dynamic program.

Maximum segment length: One straightforward way to constrain the label space is to place a bound on segment lengths. For example, in the case of smoking detection, we might say that two smoking puffs separated by five minutes (or approximately 50 respiration cycles) constitute two separate smoking activities. Adding this constraint reduces the complexity of inference to $\mathcal{O}(|\mathcal{C}|^2 LBM)$ where B is the maximum segment length.

Maximum observation distance: Depending on the observation process, we might also place a constraint on the maximum time between a true event and an associated timestamp. This corresponds to using a truncated distribution for $p_\phi(z|t)$. Given a maximum observation distance of r, we can upper bound the inference complexity by $\mathcal{O}(|\mathcal{C}|^2 LB\tilde{M})$ where $\tilde{M}$ is the maximum number of observations that could be associated with a single instance or $\tilde{M} = \max_i \sum_m \mathbb{I}\left[t_i - r \leq z_m \leq t_i + r\right]$. In practice, the average improvement in runtime is better than this.

Negative instance filtering: In cascaded classification, a simple classifier is used to filter the label set for a more complex classifier [19]. This technique has been successfully applied to structured prediction problems (e.g. [20]) and we apply it here to filter the space of possible segmentations. Due to the heavy instance level class imbalance in many mHealth problems, it is often easy to learn a high recall instance-level classifier, which can then be used to clamp instance labels to the negative class. Given an instance level classifier, let $\tilde{c}_i$ be the filter model's prediction for instance i. Then, during inference, we constrain the set of possible segmentations to agree with the negative predictions of the filter model. Using this filtering procedure, the worst case complexity remains unchanged (it is possible that the filter model filters nothing), but the average case complexity becomes $\mathcal{O}(\gamma|\mathcal{C}|^2 LB\tilde{M})$ where γ is the proportion of instances that pass the filter.

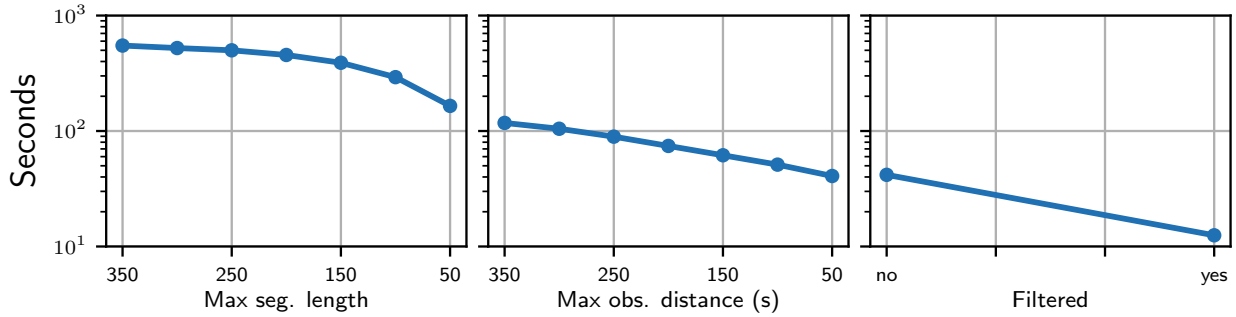


Figure 2: This figure shows the effect of changing the maximum segment length with no observation depth pruning or filtering (left), the effect of changing the maximum observation distance with no filtering (center), and the further marginal effect of filtering approximately 85% of instances (right). The maximum pruning configuration results in a 40x speedup.

To test the effect of the proposed pruning strategies, we ran an ablation experiment to assess the time required to run marginal inference in the HNS model augment with an observation model using different combinations of pruning techniques. First, we varied the maximum segment length from 350 to 50. Next, with the maximum segment length fixed at 50, we varied the maximum observation distance from 350 to 50. Finally with the maximum segment length and maximum observation distance fixed at 50, we ran inference with and without negative instance filtering. For the filtering model, we used the same actigraphy-based logistic regression model used to perform feature augmentation (Section 5.2, Instance Features). Figure 2 shows the run time in seconds for each of these settings[3]. Using all pruning strategies, the runtime of marginal inference is decreased from approximately 600 seconds to approximately 15 seconds, a 40 times speedup. We use the most aggressive pruning settings in all subsequent experiments.

Experiment 2 - Prior predictive performance: We next evaluated the ability of the proposed framework to learn the parameters of the base classifier from imprecise lab data by comparing the HNS model trained in three different ways. First, we trained the HNS model directly on the hand-aligned labels (HNS-HA). This represents the gold standard performance that we would like to achieve. Second, we trained the HNS model on labels generated by associating each observation timestamp with the closest respiration cycle (HNS-NV). This represents the naive baseline and we would expect our procedure to fall somewhere between HNS-HA and HNS-NV. Third, we trained the HNS model using the weak supervision framework proposed above (HNS-WS). Figure 3 shows the performance of all three models on the instance labeling and segmentation tasks. The HNS-WS model performs approximately as well as the HNS-HA model at both the instance labeling and segmentation tasks while the HNS-NV model performs worse than either. A paired t-test indicates that the improvement in the HNS-WS results over the HNS-NV results is statistically significant in terms of both instance labeling and segmentation ($p \leq 0.05$). These results indicate that we have achieved our primary aim of enabling learning of the HNS model from data with noisy observation timestamps.

Experiment 3 - Posterior predictive performance: We evaluated the ability of the HNS-WS model to combine sensor data with timsestamp observations at test time. As in the sleep detection experiments, we evaluated of all three models when given either all activity start observations (Start), all activity end observations (End), neither (None), or both (Start+End) at test time. The results are shown in Figure 4 (left). Unlike in our sleep detection experiments, all noise present in these observations was real and all evaluations were made against carefully hand aligned labels. While conditioning on segment observations results in improvements for all three models, these gains are much larger for the HNS-WS model. In particular, conditioning on both the segment start and end timestamps results in a 6% error reduction for the HNS-HA model and a 16% error reduction for the HNS-NV model whereas conditioning on the same information results in an 89% error reduction for the HNS-WS model.

In general, we cannot expect the noise we observe in the field to look like the noise we observe in the lab, therefore it is valuable to know how sensitive the HNS-WS model is to the correctness of the observation timestamp model. To test this, we generated synthetic observation timestamps from a normal distribution centered at the true activity start or end and varied the standard deviation of the distribution. The segmentation accuracy of the HNS-WS model when conditioning on these synthetic observations at test

[3]Runtime experiments were performed on a 2.8 GHz Intel Core i7 processor with 8GB of RAM and the inference algorithm was coded in Cython.

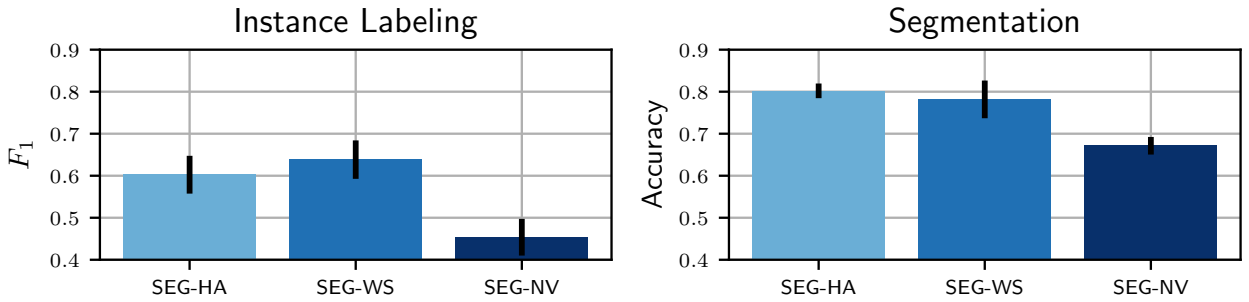


Figure 3: The left plot shows instance level F_1 score for all three models. The right plot shows the segmentation accuracy for all three models. The proposed HNS-WS model significantly outperforms the HNS-NV model. Error bars show one standard error.

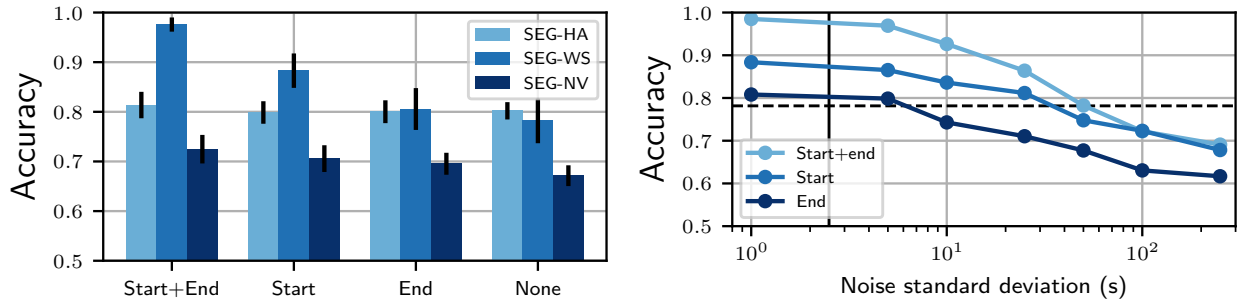


Figure 4: The left plot shows the segmentation accuracy when each HNS model is conditioned on different combinations of observations (segment start, segment end or both). The right plot shows the HNS-WS model when conditioned on segment observations with different amounts of synthetic noise added. The dashed line shows the segmentation accuracy of the HNS-WS model when conditioned on no observations (None) and the solid line shows the empirical standard deviation of the timestamp noise in the data, which reflects what HNS-WS was trained on.

time is shown in Figure 4 (right). The results show that the HNS-WS model can successfully incorporate observations with up to an order of magnitude more noise than was observed at train time. As expected, adding sufficient noise to the observations eventually causes performance to degrade; However, even with large amounts of noise, posterior segmentation accuracy plateaus between 0.6 and 0.7 compared to an accuracy of approximately 0.8 when not conditioning on any observations.

6 CONCLUSIONS

In this work, we have addressed the problem of learning time series segmentation models from noisy observation timestamps. We extended the weakly supervised learning framework of [1] to the semi-Markov CRF and HNS models and derived exact and approximate inference algorithms based on dynamic programming. We showed using real sleeping and smoking data that learning the segmentation models in this way can recover the performance of models trained on more expensive hand-aligned labels, while significantly out-performing the naive alignment strategy. Further, we showed that this framework can be used to combine noisy observations with sensor input at test time in a principled way.

This work suggests a several of interesting research directions for future research. First, in many cases it is much cheaper to gather large amounts self-report data than it is to gather lab data. The proposed framework is capable of incorporating both lab data and self-report data gathered in the field to train or fine-tune a model in a noisy semi-supervised-like learning framework. Second, personalizing detection models is an important goal in mHealth research, but is typically not practical due to the cost of obtaining labels. Our approach opens the possibility of personalizing models using less costly (but more noisy) self-report data from the field.

Acknowledgments

The authors would like to thank members of the MD2K Center (`http://www.md2k.org`) for helping to enable this research. This work was partially supported by the National Institutes of Health under award 1U54EB020404, and the National Science Foundation under award IIS-1350522.

References

[1] Roy Adams and Ben Marlin. Learning time series detection models from temporally imprecise labels. In *Artificial Intelligence and Statistics*, pages 157–165, 2017.

[2] Roy Adams, Nazir Saleheen, Edison Thomaz, Abhinav Parate, Santosh Kumar, and Benjamin Marlin. Hierarchical span-based conditional random fields for labeling and segmenting events in wearable sensor data streams. In *International Conference on Machine Learning*, pages 334–343, 2016.

[3] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *arXiv preprint arXiv:1502.05767*, 2015.

[4] Xinze Guan, Raviv Raich, and Weng-Keen Wong. Efficient multi-instance learning for activity recognition from time series data using an auto-regressive hidden markov model. In *International Conference on Machine Learning*, pages 2330–2339, 2016.

[5] Jerónimo Hernández-González, Iñaki Inza, and Jose A Lozano. Weak supervision and other non-standard classification problems: a taxonomy. *Pattern Recognition Letters*, 69:49–55, 2016.

[6] Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *Advances in neural information processing systems*, pages 897–904, 2002.

[7] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, 2001.

[8] Gideon S Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of machine learning research*, 11(Feb):955–984, 2010.

[9] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. *Advances in neural information processing systems*, pages 570–576, 1998.

[10] Annamalai Natarajan, Gustavo Angarita, Edward Gaiser, Robert Malison, Deepak Ganesan, and Benjamin M Marlin. Domain adaptation methods for improving lab-to-field generalization of cocaine detection using wearable ecg. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 875–885. ACM, 2016.

[11] Sanjay R Patel, Jia Weng, Michael Rueschman, Katherine A Dudley, Jose S Loredo, Yasmin Mossavar-Rahmani, Maricelle Ramirez, Alberto R Ramos, Kathryn Reid, Ashley N Seiger, et al. Reproducibility of a standardized actigraphy scoring algorithm for sleep in a us hispanic/latino population. *Sleep*, 38(9):1497–1503, 2015.

[12] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1796–1804, 2015.

[13] Novi Quadrianto, Alex J Smola, Tiberio S Caetano, and Quoc V Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10(Oct):2349–2374, 2009.

[14] Nazir Saleheen, Amin Ahsan Ali, Syed Monowar Hossain, Hillol Sarker, Soujanya Chatterjee, Benjamin Marlin, Emre Ertin, Mustafa Al'Absi, and Santosh Kumar. puffMarker: A Multi-sensor Approach for Pinpointing the Timing of First Lapse in Smoking Cessation. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 999–1010, 2015.

[15] Sunita Sarawagi and William W Cohen. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems*, pages 1185–1192, 2004.

[16] Hyun Oh Song, Ross B Girshick, Stefanie Jegelka, Julien Mairal, Zaid Harchaoui, Trevor Darrell, et al. On learning to localize objects with minimal supervision. In *ICML*, pages 1611–1619, 2014.

[17] Bill Triggs and Jakob J Verbeek. Scene segmentation with crfs learned from partially labeled images. In *Advances in neural information processing systems*, pages 1553–1560, 2008.

[18] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. Recognizing detailed human context in the wild from smartphones and smartwatches. *IEEE Pervasive Computing*, 16(4):62–74, 2017.

[19] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[20] David Weiss and Benjamin Taskar. Structured prediction cascades. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 916–923, 2010.