Supplemental Materials: Learning Time Series Segmentation Models from Temporally Imprecise Labels

Roy J. Adams
University of Massachusetts, Amherst

Benjamin M. Marlin
University of Massachusetts, Amherst

1 Gradient derivations

In this section, we provide detailed derivations for the gradient equations presented in section 4.1 of the main paper.

$$\begin{aligned}
\nabla_\phi \log p_\omega(\mathbf{z}|\mathbf{x},\mathbf{t}) &= \frac{1}{p_\omega(\mathbf{z}|\mathbf{x},\mathbf{t})}\sum_{o,y}\nabla_\phi p_\omega(\mathbf{o},\mathbf{y},\mathbf{z}|\mathbf{x},\mathbf{t}) && (1)\\
&= \sum_{o,y}\frac{p_\omega(\mathbf{o},\mathbf{y},\mathbf{z}|\mathbf{x},\mathbf{t})}{p_\omega(\mathbf{z}|\mathbf{x},\mathbf{t})}\nabla_\phi \log p_\omega(\mathbf{o},\mathbf{y},\mathbf{z}|\mathbf{x},\mathbf{t}) && (2)\\
&= \sum_{o,y} p_\omega(\mathbf{o},\mathbf{y}|\mathbf{z},\mathbf{x},\mathbf{t})\nabla_\phi \log p_\omega(\mathbf{o},\mathbf{y},\mathbf{z}|\mathbf{x},\mathbf{t}) && (3)\\
&= \mathbb{E}_{p_\omega(\mathbf{o}|\mathbf{z},\mathbf{x},\mathbf{t})}\left[\nabla_\phi \log p_\phi(\mathbf{z}|\mathbf{o},\mathbf{t})\right] && (4)\\
&= \mathbb{E}_{p_\omega(\mathbf{o}|\mathbf{z},\mathbf{x},\mathbf{t})}\left[\nabla_\phi \sum_{m=1}^{M}\log p_\phi(z_m|t_{i(m)})\right] && (5)\\
&= \sum_{m=1}^{M}\mathbb{E}_{p_\omega(i(m)|\mathbf{z},\mathbf{x},\mathbf{t})}\left[\nabla_\phi \log p_\phi(z_m|t_{i(m)})\right] && (6)
\end{aligned}$$

$$\begin{aligned}
\nabla_\pi \log p_\omega(\mathbf{z}|\mathbf{x},\mathbf{t}) &= \frac{1}{p_\omega(\mathbf{z}|\mathbf{x},\mathbf{t})}\sum_{o,y}\nabla_\pi p_\omega(\mathbf{o},\mathbf{y},\mathbf{z}|\mathbf{x},\mathbf{t}) && (7)\\
&= \sum_{o,y}\frac{p_\omega(\mathbf{o},\mathbf{y},\mathbf{z}|\mathbf{x},\mathbf{t})}{p_\omega(\mathbf{z}|\mathbf{x},\mathbf{t})}\nabla_\pi \log p_\omega(\mathbf{o},\mathbf{y},\mathbf{z}|\mathbf{x},\mathbf{t}) && (8)\\
&= \sum_{o,y} p_\omega(\mathbf{o},\mathbf{y}|\mathbf{z},\mathbf{x},\mathbf{t})\nabla_\pi \log p_\omega(\mathbf{o},\mathbf{y},\mathbf{z}|\mathbf{x},\mathbf{t}) && (9)\\
&= \mathbb{E}_{p_\omega(\mathbf{o},\mathbf{y}|\mathbf{z},\mathbf{x},\mathbf{t})}\left[\nabla_\pi \log p_\pi(\mathbf{o}|\mathbf{y})\right] && (10)\\
&= \mathbb{E}_{p_\omega(\mathbf{o},\mathbf{y}|\mathbf{z},\mathbf{x},\mathbf{t})}\left[\nabla_\pi \sum_{s=1}^{S}\sum_{i=j_s}^{k_s}\log p_\pi(o_i|y_s,c_{s-1},t_i)\right] && (11)\\
&= \sum_{s=1}^{S}\sum_{i=j_s}^{k_s}\mathbb{E}_{p_\omega(o_i,\mathbf{y}|\mathbf{z},\mathbf{x},\mathbf{t})}\left[\nabla_\pi \log p_\pi(o_i|y_s,c_{s-1},t_i)\right] && (12)
\end{aligned}$$

$$\begin{aligned}
\nabla_\theta \log p_\omega(\mathbf{z}|\mathbf{x}, \mathbf{t}) &= \frac{1}{p_\omega(\mathbf{z}|\mathbf{x}, \mathbf{t})} \sum_{o,y} \nabla_\theta p_\omega(\mathbf{o}, \mathbf{y}, \mathbf{z}|\mathbf{x}, \mathbf{t}) && (13)\\
&= \sum_{o,y} \frac{p_\omega(\mathbf{o}, \mathbf{y}, \mathbf{z}|\mathbf{x}, \mathbf{t})}{p_\omega(\mathbf{z}|\mathbf{x}, \mathbf{t})} \nabla_\theta \log p_\omega(\mathbf{o}, \mathbf{y}, \mathbf{z}|\mathbf{x}, \mathbf{t}) && (14)\\
&= \sum_{o,y} p_\omega(\mathbf{o}, \mathbf{y}|\mathbf{z}, \mathbf{x}, \mathbf{t}) \nabla_\theta \log p_\omega(\mathbf{o}, \mathbf{y}, \mathbf{z}|\mathbf{x}, \mathbf{t}) && (15)\\
&= \mathbb{E}_{p_\omega(\mathbf{y}|\mathbf{z},\mathbf{x},\mathbf{t})} \left[\nabla_\theta \log p_\theta(\mathbf{y}|\mathbf{x})\right] && (16)\\
&= \mathbb{E}_{p_\omega(\mathbf{y}|\mathbf{z},\mathbf{x},\mathbf{t})} \left[\nabla_\theta \langle \theta, \mathbf{f}(\mathbf{x}, \mathbf{t}, \mathbf{y})\rangle\right] - \nabla_\theta Z_\theta(\mathbf{x}) && (17)\\
&= \mathbb{E}_{p_\omega(\mathbf{y}|\mathbf{z},\mathbf{x},\mathbf{t})} \left[\mathbf{f}(\mathbf{x}, \mathbf{t}, \mathbf{y})\right] - \mathbb{E}_{p_\theta(\mathbf{y}|\mathbf{x})} \left[\mathbf{f}(\mathbf{x}, \mathbf{t}, \mathbf{y})\right] && (18)
\end{aligned}$$

2 Computing the marginal likelihood

This section gives details on the dynamic program for computing the marginal likelihood of the proposed framework described in section 4.1 of the main paper. Throughout this section we use the notation $\mathbf{x}_{j:k} = \{\mathbf{x}_i\}_{i=j,...,k}$ to refer to the subsequence of $\mathbf{x}$ from j to k (likewise for $\mathbf{t}$ and $\mathbf{z}$). The complete dynamic program is presented in 1. 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:

$$\alpha(k, c, m) \propto p_\omega(\mathbf{z}_{1:m}|\mathbf{x}_{1:k}, \mathbf{t}_{1:k}, c_{|\mathbf{y}|} = c) \tag{19}$$

Once this algorithm is complete we can calculate the unnormalized marginal likelihood for the complete model as

$$p_\omega(\mathbf{z}|\mathbf{x}, \mathbf{t}) \propto \sum_c \alpha(L, c, M). \tag{20}$$

All that remains is to normalize the unnormalized marginal likelihood. Since the observation model is locally normalized, we need only calculate the normalizer for the base semi-CRF model $Z_\theta(\mathbf{x}, \mathbf{t})$ which can be done using a dynamic program with complexity $\mathcal{O}(|\mathcal{C}|^2 L^2)$ [3].

In this algorithm, line 5 has complexity $\mathcal{O}(1)$ and is executed $\mathcal{O}(|\mathcal{C}|^2 LM)$ times, line 7 has complexity $\mathcal{O}(1)$ and is executed $\mathcal{O}(|\mathcal{C}|^2 L^2 M)$ times, and line 8 has complexity $\mathcal{O}(|\mathcal{C}|L)$ and is executed $\mathcal{O}(|\mathcal{C}|LM)$ times, so the whole algorithm has complexity $\mathcal{O}(|\mathcal{C}|^2 L^2 M)$.

```
1: for $k = 1, ..., L$ do
2:     for $c \in \mathcal{C}$ do
3:         for $m = 0, ..., M$ do
4:             for $c' \in \mathcal{C}$ do
5:                 $\beta(j, k, c, c', m) \leftarrow \sum_o \alpha(k-1, c', m-o)\, p_\pi(o|(c, k, k), c', k)\, p_\phi(z_m|t_k)^o$
6:                 for $j = 1, ..., k-1$ do
7:                     $\beta(j, k, c, c', m) \leftarrow \sum_o \beta(j, k-1, c', m-o)\, p_\pi(o|(c, j, k), c', k)\, p_\phi(z_m|t_k)^o$
8:             $\alpha(k, c, m) \leftarrow \sum_j \sum_c' \exp(\langle \theta, \mathbf{f}((c, j, k), c', \mathbf{x})\rangle) \beta(j, k, c', m)$
9: Return $\alpha$
```

Figure 1: The complete dynamic program for calculating the marginal likelihood of the observation sequence $p_\omega(\mathbf{z}|\mathbf{x}, \mathbf{t})$ in the proposed framework.

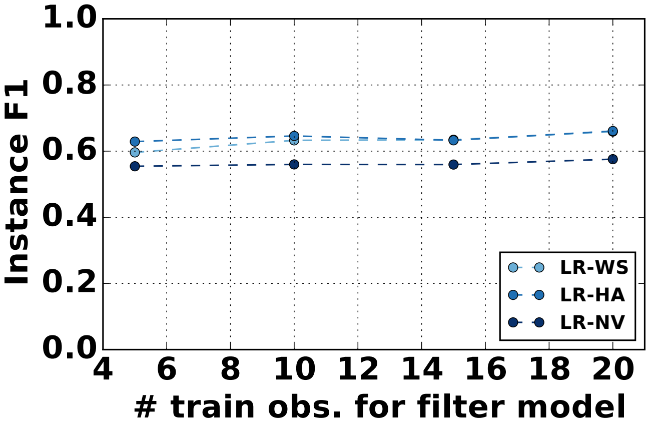

Figure 2: The instance labeling performance of logistic regression based models as a function of the number of hand-labeled sessions used to train the feature augmentation model.

3 Instance feature augmentation for smoking detection

As described in Section 5.2, we used predictions $\hat{y}_{act}$ from a logistic regression model trained using only actigraphy features on a subset of instances with hand aligned labels to augment the respiration features. The form of the resulting augmented feature vectors is $\mathbf{x}_{aug} = [\hat{y}_{act}\mathbf{x}_{resp}\ (1 - \hat{y}_{act})\mathbf{x}_{resp}]$. In this section, we consider the effect of the amount of hand-aligned data on the end-to-end prediction performance of a model learned using augmented features $\mathbf{x}_{aug}$. The experimental protocol varies the number of sessions of hand-aligned labeled instances used to train the feature augmentation model. For each number of sessions, the feature augmentation model is trained, and used to produce the augmented feature vectors $\mathbf{x}_{aug}$. For the purpose of this evaluation, a second-stage logistic regression model is then trained using the augmented features $\mathbf{x}_{aug}$.

Three settings are considered: (1) the second-stage model is logistic regression trained using hand-aligned labels (LR-HA), (2) the second-stage model is logistic regression trained using a naive alignment strategy where positive instance observations are mapped to the nearest instance (LR-NV), and (3) the second-stage model is the weakly supervised logistic regression model presented in [1] trained using the raw observation timestamps (LR-WS). In all cases, the results shown are for a leave-one-session-out experimental protocol using hand-aligned labels for testing. The results were averaged over three random seeds to account for the random sampling of the sessions used to train the feature augmentation model.

The end-to-end performance of these models is shown in Figure 2. We found that the relative performance of these models remains relatively stable as the subset size changes. In particular, there is a difference of 0.03 in the F1 score when doubling the number of sessions used to train the feature augmentation model from 10 to 20. For all experiments in section 5.2 of the main paper, we used a subset of 10 sessions to train the feature augmentation model.

4 Hierarchical Nested Segmentation as semi-CRF

This section gives details on the HNS model used in section 5.2 can be written as a constrained semi-CRF. The HNS model defines a segmentation of the input sequence into periods between positive instances, termed **inter-event spans**. Further, the segment label set $\mathcal{C} = \{0, 1, 2, ..., C\}$ includes all integers from 0 to C. A label of $c_s = 0$ indicates the the span from j_s to k_s is a non-smoking activity and a label of $c_s = c > 0$ denotes that segment s is the c'th positive inter-event span within a smoking activity (alternatively that instance j_s is the c'th positive instance in a smoking activity). To enforce these semantics, the space of allowed segmentations is constrained such that for all $s > 0$, $c_s > 0$ implies $c_{s-1} = c_s - 1$ and $c_s = 0$ implies $c_{s-1} > 0$. A consequence of these constraints is that inference complexity depends only linearly on the size of the label set (in this case $C + 1$) or inference has complexity $\mathcal{O}(|\mathcal{C}|L^2)$.

Finally, the HNS model for smoking detection is defined by the following feature function:

$$\mathbf{f}(y, c', \mathbf{x}, \mathbf{t}) = [\mathbf{x}_j,\ \sum_{i=j+1}^{k} \mathbf{x}_i,\ t_k - t_j,\ (t_k - t_j)^2,\ \mathbb{I}[c = 0]\mathbb{I}[c' = 1],\ ...,\ \mathbb{I}[c = 0]\mathbb{I}[c' = C]]$$

The first two features $\mathbf{x}_j$ and $\sum_{i=j+1}^{k} \mathbf{x}_i$ incorporate the instance level features and reflect that the first instance in a segment is positive while all others are negative. The next two features $t_k - t_j$ and $(t_k - t_j)^2$ incorporate the segment duration, in essence putting a normal distribution on the time between two positive instances. The final features $\mathbb{I}[c = 0]\mathbb{I}[c' = 1],\ ...,\ \mathbb{I}[c = 0]\mathbb{I}[c' = C]$ are used to model the number of positive instances that make up a complete activity (e.g. the number of puffs it takes to smoke a cigarette). This is refered to as a cardinality factor in [2] where the model learns a weight for every possible cardinality.

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] 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.