Positive Unlabeled Learning with a Sequential Selection Bias

Walter Gerych* Thomas Hartvigsen* Luke Buquicchio* Abdulaziz Alajaji*
Kavin Chandrasekaran* Hamid Mansoor* Elke Rundensteiner* Emmanuel Agu*

In important domains from video stream analytics to human context recognition, datasets are only *partially*-labeled. Worse yet, the labels are often applied *sequentially*, as annotators choose labels frame-by-frame or timestep-by-timestep in sequence. With labels not collected independently, this results in *sequential bias* in the labeling. Unfortunately, current state-of-the-art methods for partially labeled data are rendered ineffective under sequential bias. In this work, we propose a novel solution to tackling this open sequential bias problem, called *DeepSPU*. *DeepSPU* recovers missing labels by constructing a model of the sequentially biased labeling process itself. This labeling model is then learned jointly with the prediction model that infers the missing labels in an iterative training process. Further, we regulate this training using a theoretically-justified cost functions that prevent our model from converging to incorrect but low-cost solution. Our experimental studies demonstrate that *DeepSPU* consistently outperforms the state-of-the-art methods by over 10% on a rich variety of real-world datasets.

1 Introduction

Motivation. Collecting fully-labeled datasets is an expensive and arduous task [1], which explains why many datasets are only *partially* labeled [5]. Additionally, it is common for only some positive instances to be labeled while explicit negative labels are not given [5], due to factors such as the difficulty of constructing a representative set of reliable negatives. For instance, we can identify products that *are* of interest to a customer based on their purchase history, but it is not straightforward to reliably collect data on what they are *not* interested in. For these reasons, it is important to construct classifiers for incompletely labeled data that do not require negative annotations. State-of-the-art approaches for learning from data with only incomplete positive labels require an accurate estimation of the likelihood that any given positive instance receives a label, known as the *propensity score* [5]. However, existing approaches [3] overlook the fact that the labels annotating sequential data are often clustered together in time. Thus, the likelihood that a given instance is labeled is dependent on the labels of its surrounding instances. We refer to this as *sequential bias*. Overlooking this sequential bias results in an incorrect propensity score and thus significantly reduced classification performance - as demonstrated in our experimental study (Section 5).

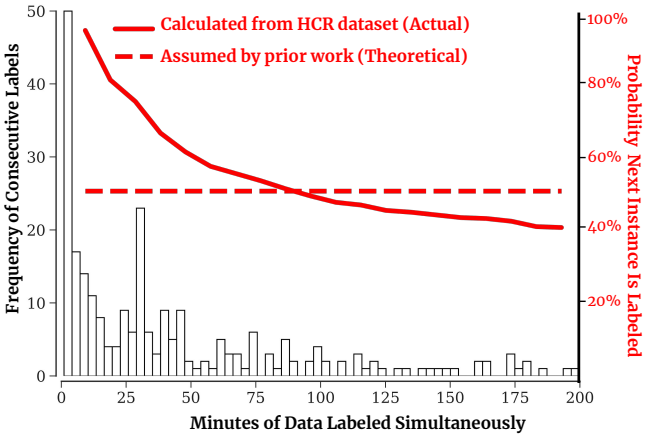

Figure 1: Histogram of lengths of consecutive labels (in minutes) in the *ExtraSensory* [25] benchmark HCR dataset.

A prime example of sequential bias in data is found in the field of *Human Context Recognition* (HCR), which focuses on building models to infer what activities or states individual humans are in given mobile sensor data [25] with the target applications ranging from security to mobile health care [1]. To construct robust HCR classifiers, it is crucial that practitioners collect realistic datasets of human activities on which to train their models. To attempt to collect such realistic labeled HCR data, data collection participants report their contexts while wearing mobile sensors. This can take many days, so participants leave many of their activities unlabeled to avoid the burden of constantly reporting their state. Participants instead label blocks of contiguous data for time periods where they are free to actively put time into data labeling. A moment is more likely to be labeled if a label has just been

*Worcester Polytechnic Institute,
{wgerych, twhartvigsen, ljbuiquicchio, asalajaji, kchandrasekaran, hmansoor, rundenst, emmanuel}@wpi.edu

collected, and vice versa. This results in a sequential bias, as instances adjacent time are likely to either all be labeled or unlabeled. The above labeling-related observations are indeed commonplace as demonstrated by our analysis of the real-world benchmark HCR dataset [25] featured in Figure 1 where the histogram showcases the frequency of blocks of time of consecutive labeling by real users.

Many other applications are susceptible to sequential bias, including intrusion detection from video [22] and illness prediction from medical records [23]. This causes crucial problems as existing partially-labeled classification methods [3, 17] show drastically reduced accuracy when sequential bias is not accounted for (as demonstrated in our Exp. Results in Section 5).

State-of-the-Art. One popular family of semi-supervised methods that learn from incompletely-labeled data are positive unlabeled (PU) classifiers [5, 9, 18, 8, 17, 3, 16, 11]. Notably, these PU methods do not require *any* negative examples to be explicitly labeled. This is a key strength of PU methods because representative negative examples, typically required by semi-supervised methods, are often not feasible to acquire. For instance, in the HCR example, there are a nearly infinite number of possible activities that an individual is *not* performing at any given time. Consequentially, participants are only expected to provide *positive* labels for the activities that they actually are performing [25].

Unfortunately, existing PU methods make unrealistically simplifying assumptions on how labels are applied. Specifically, they either assume that the labeling process carries no bias (the probability of a sample being an unlabeled positive instance is uniform) [9, 8, 17, 15], only depends on local attributes of each instance [3, 16], or else on simply counting the number of labeled instances within a window [10]. This means that existing methods do not adequately model *sequential bias*, which often manifests as a complex time-evolving function.

Problem Description. Given a dataset of sequences of instances (i.e., frames in a video, or mobile sensor readings at given timesteps), our goal is to predict the true class likelihoods of each instance within a sequence given only partial positive labels during training, such that only *some* but not all of the instances belonging to the positive class are annotated. In particular, we focus on the difficult case when labels have been assigned with a *sequential bias*, where the likelihood a positive instance is labeled varies over time.

Technical Challenges. This problem is challenging because it corresponds to two difficult and worse yet interdependent subproblems whose respective solutions depend on solving first the alternate problem. First, we have the *dependency* problem: if we had a model of the latent labeling process (which we call the *propensity model*) that allowed us to identify the true unlabeled positive instances, then we could use this propensity model to train a classifier to produce the true class likelihoods. However, we need these same true class likelihoods in order to first train the propensity model - causing a cyclic dependency. Second, a *naive* solution that assumes that all unlabeled instances are negative (or, equivalently, that all positive instances are labeled) would perfectly explain the observed models. Clearly, our approach must avoid falling into this naive and inaccurate solution.

Our Approach: DeepSPU. We propose *Deep Sequential PU* (*DeepSPU*), which is the first Positive Unlabeled method to use a propensity score model that predicts the likelihood that a given positive instance is labeled while allowing for a complex time-evolving labeling function. The propensity score allows us to train a classifier network given only partially labeled data. We achieve this by developing a novel learning method that overcomes the cyclic dependency problem by iteratively learning both the propensity score model and the classifier using weakly-labeled data. Further, we introduce two novel PU cost terms: the Prior-Matching Costs (PMC) and the Observation-Matching Costs (OMC), which prevent the propensity model and classifier from collapsing into the aforementioned naive and incorrect solution.

Contributions. Our main contributions are:

- We introduce two novel PU cost terms, Prior-Matching and Observation-Matching, which we prove prohibits collapse into certain incorrect and adversarial solutions.
- We develop the DeepSPU model that mitigates sequential bias. DeepSPU uses a joint learning strategy to jointly estimate the two interdependent latent variables: the propensity score and the true class probabilities, without any direct supervision for either learning task.
- With a series of rigorous experiments, we show that our *DeepSPU* method outperforms state-of-the-art PU methods on several real-world datasets.

2 Related Work

Many approaches to PU learning exist, such as re-weighting predictions [26, 9], iteratively identifying reliable examples [12, 19], and most notably risk minimization [21, 8, 17]. However, all these state-of-the-art approaches share the often unrealistic assumption that *no bias* exists in the labeling process, sequential or otherwise. When a bias is present, these methods are sus-

ceptible to learning skewed decision boundaries and thus prone to making biased and incorrect classifications [5].

A few recent PU methods have begun to model bias in the labeling process. One approach assumes the likelihood that a given true positive instance is labeled depends solely on its distance from the negative distribution [16], while others explore the case when this labeling likelihood depends on the general position of the positive instance in the feature space [3]. A slight modification of [3] designed specifically for sequential data was proposed in [10], which bases the propensity score on the number of labeled instances within a window, but does not allow for a complex, sequentially-evolving propensity score.

Our DeepSPU method extends beyond these methods by tackling *both* this feature-level bias *and* the previously-overlooked sequential bias.

3 Problem Definition

3.1 Positive Unlabeled Learning. Formally, let $\mathcal{D} = \{(X^{(j)}, \mathcal{L}^{(j)})\}_{j=1}^{N}$ be a dataset $\mathcal{D}$ of N sequence pairs, where $X^{(j)}$ is a sequence of real values and $\mathcal{L}^{(j)}$ is a sequence of binary label indicators, $|X^{(j)}| = |\mathcal{L}^{(j)}|$. For readability, we drop the superscript j and describe our approach in terms of one sequence. Let $X = (x_1, x_2, ..., x_T)$ be a sequence of T real values (which we refer to as a sequence of *instances*), and $\mathcal{L} = (\ell_1, \ell_2, ..., \ell_T)$ be an associated sequence of label indicators such that $\ell_i = 1$ if x_i is labeled positive, and is 0 otherwise. Additionally, for each feature-label sequence pair there is an *unobserved* binary true class sequence, $Y = (y_1, y_2, ..., y_T)$, $y_i \in \{0, 1\}$, representing the underlying classes of the instance. In addition to y_i being unavailable during training, $Pr(y_i = 1|\ell_i = 1) = 1$ as we assume there are no labels for negative instances, while $Pr(\ell_i = 1|y_i = 1) \neq 0$ as we assume not all positive instances are labeled.

We consider both *feature-level* and *sequential* biases in the labeling process. Feature-level bias assumes the propensity score for a positive instance depends on local features of the instance and thus $Pr(\ell_i = 1|y_i = 1) \neq Pr(\ell_i = 1|y_i = 1, x_i)$. Sequential bias assumes the propensity score of a positive instance can also depend on the label status of preceding instances, $\ell_{1:i-1}$, and thus it may be that $Pr(\ell_i = 1|y_i = 1) \neq Pr(\ell_i = 1|y_i = 1, x_{1:i}, \ell_{1:i-1})$. We call a propensity score that captures sequential and feature-level bias a *sequential propensity score* $q_i = Pr(\ell_i = 1|y_i = 1, x_{1:i}, \ell_{1:i-1})$.

Our goal is to train a classifier $g_\theta(\cdot)$ with parameters θ to solve the binary classification problem, such that $g_\theta(X) = Pr_Y(Y|X)$. During training, only the features X and label status indicator $\mathcal{L}$ are observed while the true class Y is unobserved.

3.2 Background on Empirical PU Risk Minimization. In standard positive-negative binary classification, the *risk* of a classifier g_θ is given as:

$$R(g_\theta) = \pi \mathbb{E}_p[C^+(g(x))] + (1 - \pi)\mathbb{E}_n[C^-(g(x))],$$

where $\mathbb{E}_p$ and $\mathbb{E}_n$ are the distribution over the positive and negative instances respectively, and C^+ is the loss incurred from predicting $g_\theta(x)$ given that the true instance is positive while C^- is the loss incurred from predicting $g_\theta(x)$ when the true instance is negative. Several recent works have focused on reformulating the above risk into a "positive-unlabeled" risk that takes expectations over the labelled and unlabeled distributions rather than the positive and negative distributions, as the latter two distributions can not be estimated directly from PU data [17, 8]. Directly minimizing the empirical PU risk has been successful in the unbiased SCAR setting [8, 17]

Additionally, Bekker *et al.* proposed a PU risk for the SAR setting, where there is bias in the labeling that is a function of the feature values of each instance [3]. For classifier g_θ this risk is given as:

$$\begin{aligned} R(g_\theta) =& \pi c \mathbb{E}_\ell \Big[\frac{1}{e(x)} C^+(g(x)) + \Big(1 - \frac{1}{e(x)}\Big) C^-(g(x)) \Big] \\ &+ (1 - \pi c)\mathbb{E}_u[C^-(g_\theta(x))], \end{aligned} \tag{3.1}$$

where $e(x)$ is the *propensity score* and represents the probability that a positive instance is labeled. In the formulation proposed by Bekker *et al.* the propensity score is *only* a function of the local feature values x.

A classifier that directly minimizes Bekker's PU risk has not been proposed for the case where both the propensity score $e(x)$ and the posterior $P_Y(Y|X)$ are unknown. This is due to the difficulty of estimating both the propensity score and the posterior jointly, which arises from the fact that in the above risk the perceived performance of the estimated posterior g_θ (and thus the gradients incurred) is based on the estimate of the propensity score and vice versa. Ergo, a poor estimate of the propensity score can lead the classifier g_θ to a poor solution while a poor classifier g_θ leads to an inaccurate estimator of the propensity score.

As described in the following section our proposed *DeepSPU* method does succeed in learning the propensity score and g_θ jointly by minimizing the PU risk directly. *DeepSPU* overcomes the aforementioned difficulty of training the two latent variables through risk minimization by using a novel iterative training algorithm coupled with additional regularization terms.

4 Methodology

4.1 Overview. We describe our proposed general estimation procedure for sequentially biased PU data, along with a specific model for learning in this setting.

4.2 Learning with Sequential Bias. Our proposed learning strategy features two major components. First, we employ an innovative Iterative Learning Strategy, iteratively training a classifier model and a propensity score model by minimizing the *positive unlabeled risk* [5]. This is achieved without explicit feedback of the two estimated latent variables. Second, we design two novel PU cost (regularization) terms that are employed during the iterative training. These cost terms prevent the networks from converging on naive solutions which minimize the PU risk but do not result in correct probability distributions for the latent target variables.

4.2.1 Iterative Learning Strategy. As stated in section 3.2 typical classifiers for fully-labeled data are trained to minimize the expected value of a loss function C, known as the *risk*. If the propensity score is known then the standard risk R can be expressed in terms of expectations over only positive and unlabeled distributions, instead of the positive and negative distributions [8, 3]. Thus, to train our model we express the *empirical positive unlabeled risk*, R_{PU}, in terms of our novel sequential propensity score q_i as:

$$
\begin{aligned}
R_{\mathrm{PU}}(g, q|X, \mathcal{L}) = \frac{1}{T}\Big(\sum_{x_i|\ell_i=1} \Big(\frac{1}{q_i} C^+(g(x_i)) \\
+ \Big(1 - \frac{1}{q_i}\Big) C^-(g(x_i))\Big) \\
+ \sum_{x_i|\ell_i=0} (C^-(g(x_i)))\Big),
\end{aligned} \tag{4.2}
$$

where C^+ is the loss incurred from predicting $g(x)$ assuming that the true class is positive, C^- is the loss incurred from predicting $g(x)$ assuming the true class is negative, and $q_i = Pr(\ell_i = 1|x_{1:i}, \ell_{1:i-1})$ is the propensity score of the i-th instance. If our propensity scores are accurate, then minimizing the above equation will correspond to minimizing the true risk. This means that in effect we can learn the same classifier that we would have found if we had been given fully labeled data.

THEOREM 4.1. *If $q_i = Pr(\ell = 1|y_i = 1, x_{1:i}, \ell_{1:i-1})$ for $i = 1 : T$, then $R_{pu}(g, q|X, \mathcal{L})$ is an unbiased estimation of the true risk $R(g|Y)$.*

Proof.

$$
\begin{aligned}
\mathbb{E}[R_{PU}(g, q|X, \mathcal{L})] &= \frac{1}{m}\sum_{i=1}^{m} y_i q_i \Big(\frac{1}{q_i} L^+(g(x_i)) \\
&\quad + (1 - \frac{1}{q_i}) L^-(g(x_i))\Big) \\
&\quad + (1 - y_i q_i) L^-(g(x_i)) \\
&= \frac{1}{m}\sum_{i=1}^{m} y_i L^+(g(x_i)) \\
&\quad + (1 - y_i) L^-(g(x_i)) \\
&= R(g|Y)
\end{aligned}
$$

□

As stated, Theorem 4.1 indicates that we can train a classifier on positive and unlabeled data if we have accurate propensity scores. However, the propensity score is in general an unobserved variable to use as a target during training, and thus there is no straightforward way to minimize the PU risk. The naive approach would be to simply learn the parameters of the classifier g and propensity model q simultaneously by minimizing Equation 4.2. However, there is no guarantee that Equation 4.2 is an unbiased estimate of the true risk if the estimated propensity scores are incorrect (and thus no indication that this would yield a good classification model). We propose to overcome this challenge with a novel *Iterative Learning Strategy*, outlined below.

We begin training the classification model and propensity score model by initializing the propensity scores with "good" estimates, acquired by assuming the labeling likelihood $c = Pr(\ell_i = 1|y_i = 1)$ is constant for all positive instances. c is easily computed given a prior on the class $\pi = Pr(y = 1)$ [5], and has been shown to be successfully estimated from the initial data set [13, 14]. We show how to derive c from π in the appendix.

Next, we train a propensity model and a classification model iteratively in separate interleaved stages. That is, at each stage of training, either the parameters of the propensity model or classification model are updated independently from each other.

4.2.2 Regularizing Cost Terms. It is possible to minimize Equation 4.2 by naively setting $q_i = 1 \; \forall \; i$ and $g(x_i) = \ell_i$. In this case, the classifier does not predict the instance's *true class*, and instead erroneously predicts whether or not an instance is labeled [3]. To avoid this adversarial solution, we propose two cost terms. First, we add the *Prior-Matching Costs (PMC)*, which drives the percentage of predicted positives to match

the percentage of true positive we expect according to our prior. The PMC is given by the KL divergence between the unconditioned distribution of predicted positives and $Bern(\pi)$, the Bernoulli distribution parameterized by the true class prior π:

$$\text{PMC}(X) = \text{KL}(Pr(\hat{y} = 1|\Theta)||Bern(\pi)). \tag{4.3}$$

Thus, the number of predicted positives are driven to match the expected number of true positives. By definition, the PMC will be larger than the number of labeled instances, and thus Equation 4.3 will incur a high penalty cost for the *adversarial solution*.

Additionally, if the estimated propensity scores match the true propensity scores, then our propensity score multiplied by the class probability of the i-th instance will equal the probability that the i-th instance is labeled, as stated in Theorem 4.2. This informs our next cost term.

THEOREM 4.2. *If $q_i = Pr(\ell_i = 1|x_{1:i}, \ell_{1:i-1}, y_i = 1)$ is the sequential propensity score of the i-th instance, then $Pr(\ell_i = 1|x_i, \ell_{1:i-1}) = q_i \cdot Pr(y_i|x_i)$.*

Proof.

$$\begin{aligned}
&Pr(\ell_i = 1|x_{1:i}, \ell_{1:i-1}) \\
&\quad = Pr(\ell_i = 1|x_i, \ell_{1:i-1}, y_i = 1)Pr(y_i = 1|x_i, \ell_{1:i-1}) \\
&\quad + Pr(\ell_i = 1|x_i, \ell_{1:i-1}, y_i = 0)Pr(y_i = 0|x_i, \ell_{1:i-1}) \\
&\quad = Pr(\ell_i = 1|x_i, \ell_{1:i-1}, y_i = 1)Pr(y_i = 1|x_i, \ell_{1:i-1}) \\
&\quad = Pr(\ell_i = 1|x_i, \ell_{1:i-1}, y_i = 1)Pr(y_i = 1|x_i)
\end{aligned}$$

□

In short, we propose to explicitly encourage the product of the estimated propensity scores and predicted class probabilities to match the observed labels. We accomplish this by adding the Binary Cross Entropy (BCE) between them as next cost term. We refer to this as the *Observation-Matching Costs (OMC)*, as it requires our propensity score and class predictions to match the observed data:

$$\text{OMC}(X) = -\sum_{j=1}^{N}\sum_{i=1}^{T} \ell_i^{(j)} log\Big(\hat{y}_i^{(j)} q_i^{(j)}\Big) \tag{4.4}$$

We thus propose a final combined cost function sequentially biased PU data as follows:

$$L(\Phi, \Theta) = I \cdot J_{Q_\phi}(g_\Theta) + (1 - I) \cdot J_{g_\Theta}(Q_\phi),$$

where I is an indicator variable that equals 1 if we're updating the parameters of the classification model g_Θ and is 0 otherwise, and J is the cost term defined as:

$$J_a(b) = R_{PU} + \lambda_1 \text{PMC} + \lambda_2 \text{OMC}, \tag{4.5}$$

with λ_1 and λ_2 being weights on the corresponding cost terms and $J_a(b)$ corresponding to the above equation as a function of b while a is held constant.

4.3 The DeepSPU Model for Sequentially Biased Data. We now propose a specific model to minimize Equation 4.5. Our model, *DeepSPU*, consists of three sub-networks: 1) the *Representation Network*, which learns a robust representation of the input data, 2) the *Sequential Propensity Network*, which models the likelihood that an instance is labeled given that it is positive, and 3) the *Classification Network*, which models the likelihood that a given instance belongs to the positive class. The Sequential Propensity Network, used only during the training stage, is crucial in the training of the Classification Network as it allows us to train the Classification Network given only weakly-labeled training data. After training, the Classification Network can be deployed without the Sequential Propensity Network to predict the true class of new instances.

4.4 Representation Network. The Sequential Propensity Network and the Classification Network both share a common base representation of the input data. This shared representation is modeled by a recurrent neural network (RNN) B_Ω with parameters Ω, such that B_Ω takes in the sequence of input data $X = (x_1, x_2, \cdots, x_T)$ and maps each element of the sequence to a corresponding latent representation $H = (h_1, h_2, \cdots, h_T)$. Each latent representation h_i is given by $h_i = F_h(x_i, h_{i-1})$, with the specific form of F_h determined by the choice of RNN network. Our implementation of *DeepSPU* uses a Gated Recurrent Unit (GRU) for the RNN [7], though the same principles apply to other RNN architectures.

4.5 Sequential Propensity Network. The aptly-named Sequential Propensity Network models the propensity score conditioned on the local feature values of each instance and the feature values of all preceding instances, along with the label indicators of all preceding instances. The first component of this subnetwork is another GRU that for each label ℓ_i produces a latent representation s_i conditioned on all previous labels. In effect, s_i summarizes the label subsequence ℓ_1 to ℓ_i into one latent variable. This GRU is coupled with a feed-forward network (FFN) that takes in the latent label representation s_{i-1}, the previous label indicator ℓ_{i-1}, and the representation of the input features learned by the Representation Network h_i in order to produce the propensity score q_i of the ith instance. With Φ representing the learnable parameters of the Sequential Propensity Network Q_Φ, Q_Φ models the se-

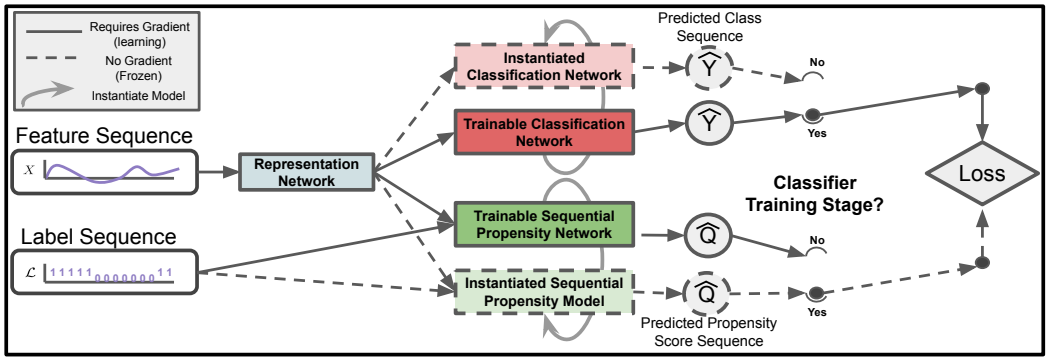

Figure 2: An overview of the DeepSPU training process.

quential propensity score as $Q_\Phi(h_i, s_{i-1}, \ell_{i-1}) = q_i = Pr(\ell_i = 1 | x_{1:i}, \ell_{1:i-1})$.

4.6 Classification Network. The shared hidden representation h_i is also passed as input into the *DeepSPU classifier subnetwork*, which models the positive class probability given the features of the input data. Our classifier model $g_\Theta(x_i) = P(y_i | x_{1:i}, \Theta)$ with parameters Θ is given by: $g_\Theta(x) = F_g(B_\Omega(x_i); \Theta)$, where F_g is a fully connected network with parameters Θ. The predicted class value for an instance x_i is given by $round(g(x_i))$, where $round(z) = 1$ if $z > 0.5$ and is 0 otherwise. We chose 0.5 as the cutoff as $g_\Theta(x_i)$ represents the *probability* that $y_i = 1$.

5 Experiments

5.1 Data and Experimental Details Datasets. We evaluate our models and a relevant set of baselines on several publically-available real-world sequential human context recognition datasets.

`UCI HAR` [2], `Older Healthy (OH) HAR` [24], and `ExtraSensory (ES)` [25]. `ExtraSensory` was subsampled at one reading for every 10 minutes of collected data due to the size of the dataset. Due to extreme class imbalance, we combine the multiple classes into one for some of the datasets. We combine `UCI HAR` into 3 classes: one representing Walking-related activities, one that combines Going Up and Down Stairs, and one that combines stationary activities. We likewise combine stationary activities in `OH HAR`. We also evaluate on `RealityCommons (RC Flu)`[1] [20], an in-the-wild health dataset, where the task is to classify whether the individual is experiencing flu symptoms on a given day using mobile sensor data.

We choose these datasets as they are either real-world examples of datasets (ES, RC Flu), or datasets with feature-values typical of real-world HCR data (UCI HAR, OH HAR). While UCI HAR and OH HAR have reliable labels, we simulate sequential bias on these datasets to evaluate our method. ES and RC Flu naturally have a sequential bias.

Each dataset has features computed from mobile sensor data, such as accelerometer, gyroscope, and magnetometer readings. From these readings statistical features, such as means and auto-correlations, are computed. The labels are either human activities (i.e., walking, sitting) or in the case of RC Flu, ailment state (sick or healhty). Additional details are given in the given links to the datasets and in the Appendix.

Compared methods. We compare DeepSPU to the following state-of-the-art methods:

- *Positive-Negative (PN) Classifier* [7]: As a baseline, we adopt a standard binary positive-negative classifier that treats all unlabeled instances as negatives. The model otherwise has the same structure as the DeepSPU classification network.

- *uPU* [8]: uPU is a recent highly-influential approach to training deep networks on PU data. uPU's convex approach minimizes the empirical PU risk. However, this approach assumes *the propensity score is constant.*

- *nnPU* [17]: Similarly to *uPU*, *nnPU* also assumes that the propensity score is constant to minimize the empirical PU risk. Additionally, nnPU clips the risk of the unlabeled instances at 0 to avoid infinitely negative risks and is thus liable to overfit during training.

- *SAR-EM* [3]: SAR-EM is the leading PU method for learning under *feature-level biased labeling.* SAR-EM jointly trains a classifier and propensity network using an Expectation-Maximization algo-

[1] http://realitycommons.media.mit.edu/socialevolution4.html

Datasets:	ES Lying	ES Sitting	ES Walking	ES Sleeping	RealityCommons
PN:	0.68	0.60	0.62	0.66	0.50
uPU:	0.70	0.62	0.65	0.70	0.60
nnPU:	0.70	0.62	0.66	0.72	0.62
SAR EM:	0.66	0.63	0.58	0.69	0.54
DeepSPU:	**0.77**	**0.66**	**0.68**	**0.78**	**0.66**

Table 1: Performance of DeepSPU vs compared methods on naturally weakly labeled real-world datasets experiencing sequential bias. DeepSPU outperforms the others. Results reported as Balanced Accuracy. Higher is better.

rithm, such that the propensity network can handle feature-level but *not* sequential bias.

- *Burst SAR-EM* [10]: A slight modification of SAR-EM for biased sequential data. This method computes the number of labeled instances within a window around each instance, and uses this value as the input feature for the propensity score model.

Evaluation metric. To measure performance, we use *balanced accuracy* (BA) defined as $BA = \frac{1}{2} * \left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$. BA is similar to standard accuracy metric, but accounts for class imbalance: BA of 0.5 is achieved by a naive classifier, regardless of class imbalance. An optimal classifier achieves a BA of 1.0.

Implementation. A 70%/10%/20% train/ validation/test split is used for each dataset. The base classifier for all methods is a GRU with a 10-dimensional hidden layer. A 1-layer feed-forward neural network uses the GRU's latent representation as input for each instance and serves as the classifier. DeepSPU has an additional 1-layer GRU for the propensity network. Each method is trained until convergence (200 epochs).

5.2 Experimental Study on Classifying Naturally-Unlabeled Data First, we demonstrate *DeepSPU*'s ability to classify data that naturally exhibits sequential bias. This can be clearly shown for the `ExtraSensory` and `RealityCommons` datasets, because both were collected "in the wild" with study participants labeling their own data sequentially. Thus they are naturally weakly labeled. Due to the study design it is unclear if an instance that is not labeled positive is negative or an unlabeled positive [6, 20], thus fitting the positive unlabeled problem description. For this experiment, we estimate the class prior for *nnPU* and *DeepSPU* using *TIcE* [4]. As shown in Table 1, *DeepSPU* significantly outperforms state-of-the-art methods for both datasets. This demonstrates that tackling sequential bias inherent in these real-world datasets is impactful. The baseline underperforms the PU methods in almost all cases, highlighting the importance of leveraging PU classifiers for these weakly-labeled real-world problems. Overall, these results confirm that the pervasive problem of sequential bias in labeling can be mitigated by *DeepSPU*.

5.3 Experimental Study on Learning from Various Levels of Limited Labels. In practice, the percentage of labeled positives available during training will vary from dataset to dataset. Thus, we perform this next experiment to study the impact of the proportion of labeling on each method's performance. In line with much of the recent PU work [5, 3, 17], we achieve this by removing subsets of labels from each dataset prior to training. We range label availability from 5% to 15% of all positive instances, and remove the labels from all negative instances. To encourage *sequential* bias, unlabeling is done sequentially: A binary Markov Chain decides whether or not to remove the label for each instance in turn. The likelihood the Markov Chain switches from "labeling" to "unlabeling" states is varied according to the desired level of unlabeling.

For each sub-sequence of consecutive positive instances within each sequence, a binary Markov chain was run to decide which instances to unlabeled. We initiated the Markov chain at state '1', where state '1' indicates that the corresponding positive instance remains labeled. The Markov chain could transition to state '0' with probability γ. We varied the value of γ to evaluate DeepSPU's performance for various levels of mislabeling. For instances where Markov chain is in state '0' the corresponding positive instance was set to be unlabeled. The chain has a 1% probability of transitioning back to '1' from state 0. This process introduces *sequential* bias, as the likelihood that a given instance is mislabeled depends on whether the previous instance is mislabeled (a given instance has a 100 X (1 - γ) % chance of being labeled if the preceding instance is labeled).

As seen in Table 2, *DeepSPU* significantly outperforms all other methods for every level of unlabeling. This demonstrates that even with very few labels, modeling sequential bias leads to significant improvements in performance. As expected, the *Positive Negative* performs the worst, as it does not account for unlabeled positives. Surprisingly, *SAR-EM* is often outperformed by *nnPU*, despite *nnPU* assuming that no bias arises in the labeling. This may be explained by *nnPU*'s natural aversion to overfitting [17], while *SAR-EM* may learn spurious relationships in the labels when mistakenly modeling sequential bias as feature-level bias. This is corroborated by *uPU* likewise beign outperformed by *nnPU*, as *uPU* is also prone to overfitting. When SAR-EM is modified to account for sequential labeling (Seq

Method:	PN			uPU			nnPU			SAR-EM			Seq SAR-EM			DeepSPU (ours)		
% Labeled:	5	10	15	5	10	15	5	10	15	5	10	15	5	10	15	5	10	15
OH Stationary	0.50	0.50	0.51	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.51	0.51	**053**	**0.53**	**0.55**
OH Ambulating	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.51	0.52	0.50	0.50	0.50	0.50	0.51	0.52	0.50	**0.52**	**0.54**
UCI Walking	0.50	0.50	0.50	0.50	0.51	0.53	0.50	0.51	0.52	0.51	0.52	0.54	0.53	0.58	0.60	**0.58**	**0.62**	**0.72**
UCI Stairs	0.50	0.50	0.50	0.59	0.62	0.72	0.64	0.65	0.67	0.52	0.53	0.58	0.52	0.57	0.62	**0.79**	**0.80**	**0.86**
UCI Stationary	0.50	0.51	0.53	0.91	0.91	0.93	0.91	0.91	0.94	0.79	0.79	0.85	0.80	0.85	0.92	**0.96**	**0.96**	**0.97**
ES Walking	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.53	**0.51**	**0.52**	**0.54**
ES Sitting	0.50	0.50	0.50	0.50	0.50	0.52	0.50	0.50	0.52	0.53	0.53	0.54	0.53	0.54	0.54	**0.55**	**0.58**	**0.60**
ES Sleeping	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.51	0.50	0.50	0.50	0.50	0.50	0.53	**0.58**	**0.65**	**0.66**
ES Lying	0.50	0.50	0.51	0.50	0.50	0.51	0.50	0.51	0.53	0.50	0.50	0.50	0.50	0.52	0.55	**0.62**	**0.69**	**0.72**

Table 2: Classification performance for various levels of labeling on the Older Healthy, UCI HAR, and ExtraSensory HAR datasets. Results reported as Balanced Accuracy (higher is better).

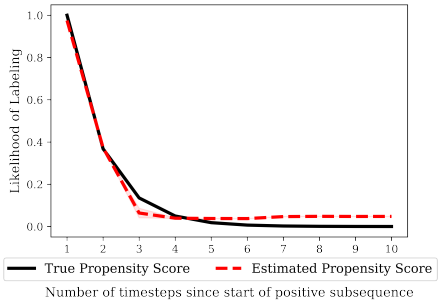

Figure 3: DeepSPU's estimated propensity score vs the true propensity scores. The estimated scores match the true scores almost perfectly.

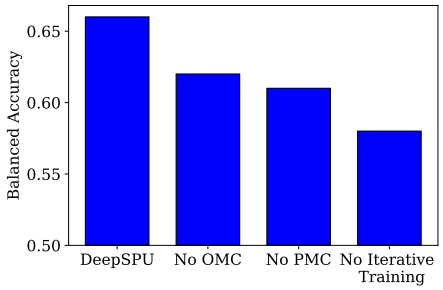

Figure 4: The ablation study of *DeepSPU* indicates that each of its major components significantly improves its performance.

SAR-EM), performance generally improves. However, as this approach simply counts the number of instances within a window around each instance, it cannot adequately model complex sequential labeling behaviors.

5.4 Ablation Study. *DeepSPU* minimizes PU risk using three components: The iterative training algorithm, the *observation-matching cost* (OMC), and the *prior-matching cost* (PMC). In this experiment, we demonstrate the necessity of each component in an ablation study on the `ExtraSensory Sitting` dataset. Removing any of the three components results in significantly lower classification accuracy. Specifically, removing the iterative training strategy impacts the performance most significantly. This is expected, as without the iterative training strategy the cost function is likely to be biased. We also see that the PMC divergence, which encourages the percentage of predicted positives to match the class prior, is the more important of the two novel cost terms.

5.5 Evaluating Performance of Propensity Model Finally, we perform an additional experiment to demonstrate that DeepSPU's propensity network accurately learns the true propensity score. To achieve this, we create a dataset containing subsequences of positive instances and subsequences of negative instances. The feature values for the positive instances are drawn from a normal distribution with mean 0 and unit variance, while the features for the negative instances are drawn from a normal distribution with mean 10 and unit variance. The positive subsequences are labeled using a gamma distribution to decide which positive instances receive labels with shape, location, and scale parameters all set to 1. We then train DeepSPU on this data and extract its learned propensity scores. As shown in Figure 3, the estimated propensity scores match the true propensity scores almost perfectly. This implies that our DeepSPU method's Sequential Propensity Network is able to accurately recover the true propensity score. Evidence that the obtained propensity scores match the true propensity scores implies that the learned class probabilities will be accurate.

6 Conclusion

We propose *DeepSPU*, the first deep PU solution for complex sequential bias in the labeling. We formulate a novel iterative learning strategy to jointly train a classification model and labeling likelihood (propensity) model, along with designing two theoretically-justified

PU cost terms to account for this bias. Through a series of extensive experimental results we demonstrate that the previously-overlooked sequential labeling bias naturally arises in real-world datasets. Also, the state-of-the-art PU methods have poor performance when this type of bias is present, while *DeepSPU* achieves robust classification performance under sequential labeling bias for a variety of real-world data sets.

Acknowledgements. This work was funded by the DARPA WASH program HR001117S0032. Results in this paper were obtained in part using a high-performance computing system acquired through NSF MRI grant DMS-1337943 to WPI.

References

[1] A. Alajaji, W. Gerych, K. Chandrasekaran, L. Buquicchio, E. Agu, and E. Rundensteiner. Deepcontext: Parameterized compatibility-based attention cnn for human context recognition. In *ICSC*. IEEE, 2020.

[2] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *Esann*, 2013.

[3] J. Bekker and J. Davis. Learning from positive and unlabeled data under the selected at random assumption. In *Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*, 2018.

[4] J. Bekker and J. Davis. Estimating the class prior in positive and unlabeled data through decision tree induction. In *AAAI*, 2018.

[5] J. Bekker and J. Davis. Learning from positive and unlabeled data: a survey. 2020.

[6] Y.-J. Chang, G. Paruthi, H.-Y. Wu, H.-Y. Lin, and M. W. Newman. An investigation of using mobile and situated crowdsourcing to collect annotated travel activity data in real-word settings. *International Journal of Human-Computer Studies*, 2017.

[7] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

[8] M. Du Plessis, G. Niu, and M. Sugiyama. Convex formulation for learning from positive and unlabeled data. In *ICML*, 2015.

[9] C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *ACM SIGKDD*, 2008.

[10] W. Gerych, L. Buquicchio, K. Chandrasekaran, A. Alajaji, H. Mansoor, A. Murphy, E. Rundensteiner, and E. Agu. Burstpu: Classification of weakly labeled datasets with sequential bias. In *IEEE Big Data*, 2020.

[11] T. Guo, C. Xu, J. Huang, Y. Wang, B. Shi, C. Xu, and D. Tao. On positive-unlabeled classification in gan. In *CVPR*, 2020.

[12] D. Ienco and R. G. Pensa. Positive and unlabeled learning in categorical data. *Neurocomputing*, 2016.

[13] S. Jain, M. White, and P. Radivojac. Estimating the class prior and posterior from noisy positives and unlabeled data. In *NeurIPS*, 2016.

[14] S. Jain, J. Delano, H. Sharma, and P. Radivojac. Class prior estimation with biased positives and unlabeled examples. In *AAAI*, 2020.

[15] L. Jiang, D. Li, Q. Wang, S. Wang, and S. Wang. Improving positive unlabeled learning: Practical aul estimation and new training method for extremely imbalanced data sets. *arXiv*, 2020.

[16] M. Kato, T. Teshima, and J. Honda. Learning from positive and unlabeled data with a selection bias. In *ICLR*, 2019.

[17] R. Kiryo, G. Niu, M. C. du Plessis, and M. Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *NeurIPS*, 2017.

[18] X.-L. Li and B. Liu. Learning from positive and unlabeled examples with different data distributions. In *ECML*, 2005.

[19] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu. Building text classifiers using positive and unlabeled examples. In *ICDM*, 2003.

[20] A. Madan, M. Cebrian, S. Moturu, K. Farrahi, et al. Sensing the" health state" of a community. *IEEE Pervasive Computing*, 2011.

[21] C. Northcutt, T. Wu, and I. Chuang. Learning with confident examples: Rank pruning for robust classification with noisy labels. In *UAI*, 2017.

[22] I. Rodríguez-Moreno, J. M. Martínez-Otzeta, B. Sierra, I. Rodriguez, and E. Jauregi. Video activity recognition: State-of-the-art. *Sensors*, 2019.

[23] M. Schaekermann, E. Law, K. Larson, and A. Lim. Expert disagreement in sequential labeling: A case study on adjudication in medical time series analysis. In *SAD/CrowdBias@ HCOMP*, 2018.

[24] R. L. S. Torres, D. C. Ranasinghe, Q. Shi, and A. P. Sample. Sensor enabled wearable rfid technology for mitigating the risk of falls near beds. In *RFID*. IEEE, 2013.

[25] Y. Vaizman, K. Ellis, and G. Lanckriet. Recognizing detailed human context in the wild from smartphones and smartwatches. *IEEE Pervasive Computing*, 2017.

[26] D. Zhang and W. S. Lee. A simple probabilistic approach to learning from positive and unlabeled examples. In *UKCI*, 2005.