BurstPU: Classification of Weakly Labeled Datasets with Sequential Bias

Walter Gerych[1], Luke Buquicchio[1], Kavin Chandrasekaran[1], Abdulaziz Alajaji[1], Hamid Mansoor[2], Aidan Murphy[3], Elke Rundensteiner[1,2], and Emmanuel Agu[1,2]

[1]Data Science Program, Worcester Polytechnic Institute, Worcester, MA
[2]Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA
[3]Department of Mathematics, Virginia Polytechnic Institute, Blacksburg, VA

2020 IEEE International Conference on Big Data (Big Data) | 978-1-7281-6251-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigData50022.2020.9377957

***Abstract*—In big data applications from digital health to assisted living smart systems, only a fraction of data instances used for training classifiers tend to be labeled. One important subfield of weakly labeled learning, called Positive Unlabeled (PU) learning, does not require a completely labeled dataset in order to train a strong classifier. This is crucial as in many domains it is expensive or impossible to obtain a completely labeled dataset. While prior PU work assumed that unlabeled instances occurred with a random uniform distribution, we observe that labeled (and unlabeled) data tends to occur in long contiguous sequences (or *bursts*) due the prevalent burst labeling behavior by human annotators. Burst labeling leads to a sequential bias in PU data not addressed by state-of-the-art methods. To tackle this open problem of learning under sequential bias, we propose *BurstPU*, the first framework for training a classifier on sequentially labeled PU data. *BurstPU* addresses the challenge that two interdependent models must be learned, namely, the classification model and the *labeling likelihood* model, with the later predicting the likelihood that a given instance is labeled. The labeling likelihood model is then needed during the training of the classification model to account for the bias in the labeling process. Our experimental study demonstrates that *BurstPU* consistently outperforms all state-of-the-art PU methods on a rich variety of diverse real-world datasets, and can learn from fewer labeled instances compared to state-of-art PU methods.**

***Index Terms*—Labeling likelihood, classification, incomplete labeling, positive unlabeled.**

I. Introduction

Background. Traditionally, supervised classification assumes ground-truth class labels are available for all instances during training. However, in many real-world scenarios involving big data only a small fraction of the total data is labeled while the majority of the data are unlabeled [1, 2]. One example is the domain of Human Activity Recognition (HAR) systems [1, 3] that aim to recognize user activities from mobile sensors on smartphones and other digital devices [1]. To develop robust classifiers for HAR systems, large quantities of sensor data are collected using smartphones and wearables over extended periods as the owers of these smart devices go about their normal routines [1]. This results in data that is more true to life than data collected in a laboratory setting, but requires individuals to label their own data. Unfortunately, since the data is collected continuously [1], it is infeasible to expect individuals to label every activity they perform throughout the day. Additionally, as there is an infinite number of activities that an individual does *not* perform at a given time, HAR data collection thus typically asks participants to provide only positive labels for the activities that they indeed did perform [1]. This results in data that contains reliable labels for only the positive class while the rest of the data remains unlabeled. Thus there is no straightforward way to disambiguate a true negative instance from the unlabeled positive instances. This type of data is known as *Positive Unlabeled* (PU) data [2].

PU data is prevalent across many applications [2, 4]. For instance, PU data may arise when labeling objects within video streams for the purpose of scene description for the visually impaired [5, 6], and when annotating medical data such as ECG sensor readings to detect abnormalities [7]. While state-of-the-art PU methods assume that labels are assigned independently of one another, we observe that this does not match the reality of data collection in many domains.

Motivation for burst labeling patterns. Through conducting our own HAR data collection study [8] and analyzing the data from similar studies [1], we observed that annotators often label data in bursts producing contiguous sequences of labeled data [3]. We henceforth refer to this as the *burst labeling pattern.* In the motivating HAR scenario, users may label diligently during time periods when they are free and fail to label for long periods when they are busy. To reduce the tedium associated with labeling, some annotation interfaces permit users to label batches of past or future sensor data at a time (e.g., retrospectively during their free time at the end of the day) [3]. While convenient, such interfaces generate coarse-grained burst labels. From a PU perspective, these *burst labeling* patterns result in a *sequential bias in the labeling*. Instances are more likely to be labeled if their surrounding instances are also labeled. An illustration of burst labeling is shown in Figure 1. In the figure's example, two subsequences of 'walking' are shown. One subsequence is partially labeled, while the other is completely unlabeled. This implies that instances within the partially labeled subsequence are more likely to be labeled than instances in the unlabeled subsequence.

State-of-the-art and its Shortcomings. Typical semi-supervised learning approaches require reliable samples from **both** the positive and negative class, and are therefore inapplicable to PU data [9]. Thus, customized learning methods

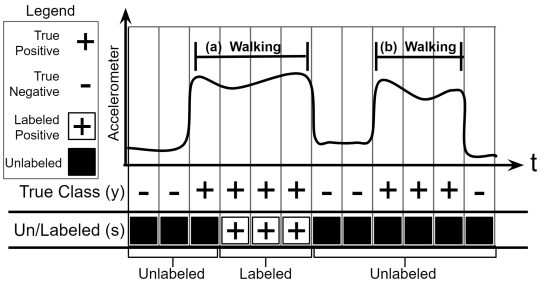


Fig. 1: In this sequentially biased Positive Unlabeled data example, two sub-sequences correspond to the positive class 'Walking'. Sub-sequence (a) has been partially annotated, while (b) is completely unlabeled. In this burst labeling setting, we observe that a positive instance is more likely to be labeled if the preceding instance was labeled, and vice versa.

for training classifiers specifically on PU data have recently received much attention [9–12]. These state-of-the-art *Positive Unlabeled (PU) learning* methods make the simplifying yet unrealistic assumption that the occurrence of labeled data instances are independent. They either assume the probability of an instance being labeled is *Selected Completely At Random* (SCAR) or *Selected At Random* (SAR), meaning the likelihood of a true-positive instance going unlabeled is either constant (*i.i.d.*) [10–12] or solely based on the local attributes of the instance itself [9, 13], respectively. Existing PU methods, being based on either the SAR or SCAR assumptions, fail to address the burst labeling problem as they are not equipped to utilize knowledge of burst labeling during their learning process [2].

Problem Statement.

In this work we tackle the problem of learning under *burst labeling*. Burst labeling corresponds to the case where the likelihood that a given instance is labeled depends on whether its surrounding instances were also labeled, as illustrated in Figure 2. Figure 2 a) shows that under existing PU models the labeling likelihood for two instances with the same feature values is identical, regardless of whether or not the surrounding instances are labeled. Figure 2 b) shows the scenario in which our problem is set; where two instances with identical feature values may have different labeling likelihoods depending on whether the surrounding instances are labeled. Thus, our *burst labeling learning* problem is to train a classifier to predict the true class using PU training labeled by burst labeling. Our aim is to optimize the performance of the PU classifier trained on this PU data so that it is comparable to the performance of a similar classifier trained on the fully labeled data, where performance is measured using MSE.

Challenges. Burst labeling learning poses two challenges:

1) *Learning with biased labels.* During training, positive labels for only some of the true positive instances are available, while the remaining data (true positives and all true negative instances) are unlabeled. Worse yet, as the probability of label assignment is not constant under burst labeling, the distribution of labeled instances is biased and is thus not the same as the distribution of true positive instances [2].
2) *Inference of unknown labeling and class likelihoods.* The likelihood that an instance is labeled needs to be inferred in an unsupervised manner. If the true class likelihood was known, then the labeling likelihood could easily be inferred. Unfortunately, we face a chicken-and-egg problem in that to train a classifier for the true class likelihood, the burst likelihood would need to be known.

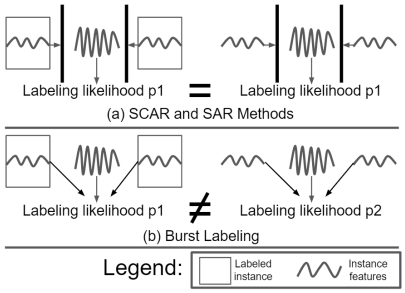


Fig. 2: a) Under SAR and SCAR (top), two instances with the same feature values will, by definition, have the same labeling likelihood. b) Under our *burst labeling* (bottom), they could have different labeling likelihoods depending on the labels of their surrounding instances.

Our Proposed Approach: *BurstPU*. We propose *BurstPU*, the first PU learning method that solves burst labeling learning while leveraging the knowledge that labels are assigned in sequentially. One innovation of the *BurstPU* learning method is the design of a *burst likelihood model* that represents the likelihood of a given instance being labeled given the labels of surrounding instances. *BurstPU* then utilizes this burst likelihood model to re-weight each PU instance during training to address the challenge of **learning with biased labels**. Secondly, we design a training algorithm for BurstPU that utilizes an Expectation-Maximization algorithm and the novel *burst empirical risk* to jointly learn both this burst likelihood score along with the BurstPU classification model. This way BurstPU succeeds to learn both the classification and burst likelihood models using only observed data, thus solving the above challenge of the **inference of unknown labeling and class likelihoods**.

Contributions. We make the following contributions:

- *Characterize burst labeling*, which captures labeling characteristics of many applications;
- *Develop the first burst labeling PU learning framework* that learns from data that follows this new burst labeling;
- *Introduce a novel burst likelihood model* that effectively models the labeling process of burst labeling;
- *Establish the error bound* between a classifier learned through minimizing the aforementioned burst risk to that of one found by minimizing the true risk;
- *Demonstrate experimentally that BurstPU consistently outperforms state-of-the-art PU methods* on real-world datasets by 7% in mean squared error.

II. Related Work

Existing PU methods do not address burst labeling. Rather, they fall into one of two broad categories: 1) those that make the *selected completely at random* (*SCAR*) assumption, and 2) those that make the *selected at random* (*SAR*) assumption.

SCAR is the most common assumption made by PU learning methods [4, 10–12, 14]. Under *SCAR*, it is assumed that each true positive instance has the same likelihood of being labeled [10]. Thus, the labeling likelihood is *constant*. State-of-the-art *SCAR* methods make use of empirical risk minimization of the Positive Unlabeled risk [11]. An improvement to this approach [12] introduced a non-negative PU risk estimator, which is less prone to overfitting. *SCAR* is a restrictive assumption, as it differs from BurstPU by not allowing for the existence of any bias in the labeling process. For this reason it is not applicable for the *burst labeling learning problem.*

Recently, *SAR* [9, 13], a less restrictive assumption than *SCAR*, is introduced that allows for simple bias in labeling. That is, *SAR* assumes that the probability of an instance being labeled is not constant and depends only on the instance's feature values. However, *SAR only* allows for the bias to depend on the instance's feature values. Unlike BurstPU, it is unsuitable for learning under burst labeling as two instances with identical feature values will be given the same labeling likelihood regardless of surrounding instances.

III. Problem Formulation: Burst PU Learning

Intuitively, in this work we solve PU learning under the realistic *burst labeling* problem setting. This means that we allow for the probability that a true positive instance is labeled to vary depending on if its surrounding instances are labeled. This is denoted by $Pr(s_i = 1|y_i = 1) \neq Pr(s_i = 1|y_i = 1, s_{k\neq i})$, where s_i and y_i are the observed label and true class of the ith data instance respectively, and $s_{k\neq i}$ is the label of all other instances besides the ith instance. We call the likelihood that a given instance x_i is labeled the *burst labeling likelihood* $q_i = Pr(s_i = 1|y_i = 1, s_{k\neq i})$.

More formally, we define burst labeling learning as follows: Let $\mathcal{D} = \{D^1, D^2, ..., D^n\}$ be a dataset of n PU sequences, where $D^j = (X^j, \mathcal{S}^j, Y^j)$. $X^j = (x_1^j, x_2^j, ..., x_m^j)$ denotes a sequence of observed data instances, $\mathcal{S}^j = (s_1^j, s_2^j, ..., s_m^j)$ an associated set of observed variables indicating whether or not the corresponding instance is labeled, and $Y^j = (y_1^j, y_2^j, ..., y_m^j)$ the set of unobserved true class labels for all instances. We say that y_i^j is *unobserved*, because with PU data we do not have access to the true class of each instance. Instead, instances that are labeled are a subset of positive instances, while unlabeled instances are a mix of some positive and all negative instances. For the sake of readability we drop the superscript j and refer to only a single sequence However, the actual dataset may consist of multiple independent sequences.

As PU learning corresponds to learning a (binary) classifier[1], the unobserved class y_i can take on the value of 1 or 0, such that y_i is the binary class label where positive instances of the class correspond to $y_i = 1$, with $y_i = 0$ otherwise. s_i is an indicator variable that indicates whether or not a given instance is labeled such that $s_i = 1$ if the instance is labeled and $s_i = 0$ otherwise.

[1]We henceforth focus on binary classifiers, while in multi-label HAR classification, a binary classifier is trained for each of the activities.

Symbol	Meaning
$\mathcal{D}$	Dataset of sequences of features (X), labels ($\mathcal{S}$), and unobserved true class (Y)
x_i	Observed ith data instance. $x \in \mathbb{R}^M$, $M \geq 1$
s_i	Observed label indicator for x_i. s_i = 1 if labeled, 0 otherwise
y_i	Unobserved true class for x_i. y_i = 1 if the positive class, 0 otherwise.
q_i	Unobserved labeling likelihood q_i for x_i $q_i = \Pr(s_i = 1\|y_i = 1, s_{k\neq i}, T_i)$
$s_{k\neq i}$	$S - \{s_i\}$
T_i	Unobserved number of directly preceding instances of x_i with positive true class
$\hat{v}$	Estimated value for any variable v

TABLE I: Reference for symbols used in this work.

Further, we make the standard *positive unlabeled assumption* [10] that negative instances are not labeled; i.e., $Pr(s_i = 1|y_i = 0) = 0$. Note that the inverse is not true; we do **not** assume that all true positives are labeled; i.e., $Pr(y_i = 1|s_i = 0) \neq 0$. Additionally, we assume that if the previous instance was a true positive then the current instance is also likely to be positive, expressed by $Pr(y_i = 1|y_{i-1} = 1) \neq Pr(y_i = 1|y_{i-1} = 0)$. This corresponds to the belief that positive instances occur in bursts or subsequences, which as discussed in Section 1 is realistic for a broad range of real-world applications.

Definition 1. A PU dataset is labeled under **burst labeling** if the likelihood of an instance being labeled is dependent on the labels of the preceding and surrounding instances. Thus, $Pr(s_i = 1|y_i = 1) \neq Pr(s_i = 1|y_i = 1, s_{k\neq i})$.

We do not make the rigid SCAR assumption that the likelihood of a positive instance being labeled is the same for all positive instances. Instead, we perform our *PU learning under the more realistic but more difficult burst labeling problem setting* (Def. 1). Burst labeling differs from the SAR assumption [9], in that unlike SAR, we now leverage the sequential dependency between the labels of neighboring data instances. Lastly, we tackle the challenge that the labeling likelihood is **not** known during training, as in the real world this value would rarely be known.

Problem Definition: Our burst labeling learning problem is to train a classifier $f(x_i) = Pr(y_i|x_i)$ to correctly predict the true class y_i given only PU data applied under burst labeling. That is, where the true class y_i is unknown, the indicator s_i has been applied under burst labeling (Def. 1), and the likelihood that a given instance x_i is labeled, q_i, is unknown. We measure success by whether or not the PU classifier achieves comparable performance to a classifier that is trained on perfectly labeled data, with the performance measured by the MSE between the predicted probabilities and the true class

labels of a held-out test set.

IV. Proposed Burst PU Learning Methodology

A. Burst Likelihood Function

To perform PU learning under burst labeling, we introduce the *burst likelihood function*, which estimates the burst labeling likelihood.

Definition 2. [Burst Likelihood Function] The burst likelihood function for the ith instance, x_i, is defined as:

$$q_n(x_i, s_{k\neq i}, T_i) = Pr(s_i|y_i = 1, s_{k\neq i}, T_i)$$

where $s_{k\neq i} = S - \{s_i\}$ and T_i is the number of directly preceding true positive instances, such that T_i is defined as:

$$T_i = \begin{cases} max_k \text{ s.t. } y_{i-1} = y_{i-2} = ... = y_{i-k} & \text{if } y_{i-1} = 1 \\ 0 & \text{otherwise} \end{cases}$$

Intuitively, T_i measures whether the ith instance, x_i, is in a burst of positive instances and, if so, for how long the sequence had occurred up until that point. This measure serves two purposes: 1) In the case where a given instance x_i has no surrounding labeled instances, T_i informs the burst likelihood function of whether the surrounding instances were not labeled due to being true negatives or if they were unlabeled true positives. 2) T_i captures the observation that when individuals perform burst labeling, if they decide to label a sequence they may incorrectly record the start and end times.

Clearly we can not calculate T_i directly from the data as the true class y_i is an unobserved variable during training. Instead, we estimate T_i as:

$$\hat{T}i = \begin{cases} max_k \; s.t. \; \hat{y}_{i-1} = \hat{y}_{i-2} = ... = \hat{y}_{i-k} & \text{if } \hat{y}_{i-1} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\hat{y}_i$ is an estimated class label. $\hat{y}_i$ can be estimated from another PU learning method, such as SAR-EM [9]. These initial estimated class labels are not as accurate as the final class predictions obtained using BurstPU as they do not account for burst labeling. However, we have experimentally validated that using $\hat{T}_i$ for the initial estimation in the learning process of BurstPU improves classification performance.

If the burst labeling likelihood was known, then we could perform PU learning under the burst labeling scenario through PU empirical risk minimization as is done by [12]. That is, we could reformulate the empirical risk equation to take into account the burst likelihood score similar to how [9] utilize the propensity score for empirical risk minimization. We define the *estimated burst PU risk* as:

$$\hat{R}_{burst}(\hat{y}_i|q_i, s_i) = \frac{1}{m}\sum_{i=1}^{m} s_i\Big(\frac{1}{q_i}C_1(\hat{y}_i) + (1 - \frac{1}{q_i})C_0(\hat{y}_i)\Big) + (1 - s_i)C_0(\hat{y}_i) \tag{2}$$

where C is a loss function and $C_1(\hat{y})$ and $C_0(\hat{y})$ are the costs incurred from predicting $\hat{y} = 1$ and $\hat{y} = 0$ according to that loss respectively. C can be a standard loss function such as the hinge loss or logistic loss (which is the loss used for BurstPU). The properties required for choices of loss function C are described in [12].

Minimizing the empirical burst risk is appropriate for training a classifier as the burst risk is equal in expectation to the true risk [9]:

Theorem 1. *$\hat{R}_{burst}(\hat{y}_i|q_i, s_i)$ is an unbiased estimation of the true risk $R(\hat{y}_i|y_i)$.*

Proof.

$$\begin{aligned}\mathbb{E}[\hat{R}_{burst}(\hat{y}_i|q_i, s_i)] &= \frac{1}{m}\sum_{i=1}^{m} y_i q_i\Big(\frac{1}{q_i}C_1(\hat{y}_i) + (1 - \frac{1}{q_i})C_0(\hat{y}_i)\Big) \\ &\quad + (1 - y_i q_i)C_0(\hat{y}_i) \\ &= \frac{1}{m}\sum_{i=1}^{m} y_i C_1(\hat{y}_i) + (1 - y_i)C_0(\hat{y}_i) \\ &= R(\hat{y}_i|y_i)\end{aligned}$$

■

Theorem 1 shows that the empirical burst likelihood risk is an unbiased estimator of the true risk. Thus, minimizing the empirical burst likelihood risk is analogous to minimizing the standard risk that would be minimized if we had perfectly labeled data.

If the burst likelihood score is known, we can train a classifier f to predict the true class y_i by minimizing the burst risk in Equation 2. The re-scoring process corresponding to minimizing Equation 2 addresses the challenge of **biased labeling**, as re-weighting each instance by the burst likelihood score would then account for the bias in the labeling process.

Below, we establish a bound on the error of a classifier trained through empirical risk minimization of the burst likelihood estimator. This bound represents how much the empirical burst risk may differ from the true standard risk within a probability of $1 - \eta$. In effect, it quantifies the maximum difference between our classifier trained on PU data and a classifier trained on perfectly labeled data. Note that this bound differs from the bound under the *SAR* assumption [9], because the labeling likelihood is not independent and identically distributed under burst labeling.

Theorem 2. *Let $f^* = \arg\min_{f\in\mathcal{H}} \hat{R}_{burst}(f)$ be the classifier found through empirical risk minimization of the burst likelihood risk in hypothesis space $\mathcal{H}$, where $\hat{R}$ is chosen with appropriate C such that $\hat{R}$ is c-Lipschitz with respect to the Hamming metric on $\mathbb{R}$. Then, with probability $1 - \eta$, $\hat{R}_{burst}(f^*|q_i, s_i) \leq R(f^*|y_i) + \sqrt{2c^2\|\Delta_n\|_\infty^2 ln\frac{2|\mathcal{H}|}{\eta}}$, where Δ_n is the mixture coefficient matrix.*

Proof.

$$\begin{aligned}&Pr(\hat{R}_{burst}(f^*|q_i, s_i) - R(f^*|y_i) \geq \epsilon) \\ &\leq Pr(\vee(\hat{R}_{burst}(f^*|q_i, s_i) - R(f^*|y_i)) \geq \epsilon) \\ &\leq \sum_{i=1}^{|\mathcal{H}|} Pr(\hat{R}_{burst}(f^*|q_i, s_i) - R(f^*|y_i) \geq \epsilon)\end{aligned}$$

Then, by the Kontorovich inequality [15]

$$\leq |\mathcal{H}| \cdot 2 \cdot exp(-\frac{\epsilon^2}{2c^2\|\Delta_n\|_\infty^2}) = \eta$$

Solving for ϵ, we find

$$\epsilon = \sqrt{2c^2\|\Delta_n\|_\infty^2 ln\frac{2|\mathcal{H}|}{\eta}}$$

■

Note that the mixture coefficient matrix Δ_n is a matrix containing values that quantify the dependence of our random variables as defined in [15].

It is important to note here that in practice, the burst likelihood score is likely to be unknown. Unfortunately, we cannot use the burst likelihood risk (Equation 2) to estimate the burst likelihood score as we would need an estimation of the true class labels in order to do so. We cannot obtain estimates for the true class labels without an estimation of the burst likelihood score. This is the aforementioned **first challenge of inference of the interdependent unknown labeling likelihoods and the class likelihoods** (Section 1).

In the case of unknown labeling likelihoods, we would need to estimate this burst likelihood score from the data itself. After estimating the score for each instance, we then could train a final classifier by minimizing the above risk function using the estimated burst likelihood scores.

B. The Burst Likelihood Score Model

In order to estimate the burst labeling likelihood, q_i, defined in Definition 4.1, we assume that q_i can be modeled through some classifier with parameters Φ. Additionally, due to the fact that in practice when there is a long enough duration between subsequences of true positives, the labeling of an instance in one burst is effectively independent of whether or not the earlier burst was labeled, we assume that for $0 < \delta << 1$ there exists a $k > 0$ such that $corr(x_i, x_k) < \delta$ ($\forall\ i >> k$).

Thus, this implies that:

$$\begin{aligned}\hat{q}(x_i, s_{k\neq i}, T_i; \Phi) &= \Pr(s_i|y_i = 1, s_{i-k:i-1}, si+1:i+k, T_i) \\ &\approx \Pr(s_i|y_i = 1, s_{i-k:i-1}, si+1:i+k, T_i) \\ &= \hat{q}(x_i, s_{i-k:i-1}, si+1:i+k, T_i; \Phi).\end{aligned}$$

We can therefore train $\hat{q}_i$ using the indicator, s_i, of only the preceding and succeeding k instances, and not *all* instances. Further, we assume that $Pr(s_i|y_i, s_{i-k:i-1,i+1:i+k}, T_i)$ can be modeled by $Pr(s_i|y_i, Count_k(x_i), T_i)$, where $Count_k(x_i)$ returns the count of labeled instances within a window size of k around the given instance x_i. In other words, we assume that the likelihood that a given instance is labeled depends on whether the surrounding instances are labeled such that this dependency can be captured by the number of labeled instances among the surrounding instances.

Characteristics	Datasets			
	EEG	**Occupancy**	**Ozone**	**ITW HAR**
# Features	4	7	73	52
# Instances	14640	20560	2536	26798
Binary?	Yes	Yes	Yes	No

C. Learning the Burst Likelihood Model and Classifier

Note that the burst labeling likelihood is conditioned on the true class y_i, as stated in Definition 2. This means we cannot train a burst labeling likelihood model from PU data in a straightforward manner, as y_i is an unobserved latent variable. In order to overcome this "chicken and the egg" problem, we learn $\hat{q}_i$ jointly with $\hat{f}$, where $\hat{f}(x_i; \Theta)$ is the model for $\Pr(y_i|x_i)$. We do this through an expectation-maximization (EM) process as this has been shown to be effective for PU learning in other PU scenarios [9]. We utilize our EM algorithm to train $\hat{q}_i$ and $\hat{f}$ by maximizing the likelihood of $Pr(x_i, s_i, y_i)$, where:

$$Pr(x_i, s_i, y_i|s_{k\neq i}, T_i) \sim Pr(x_i)Pr(y_i|x_i)Pr(s_i|y_i, s_{k\neq i}, T_i)$$

Learning the burst likelihood score in this way addresses the aforementioned second challenge of **learning with biased labels** (Section 1).

D. Local Sequence Certainty Parameter

[9] showed that a classifier $\hat{f}$, which predicts the probability of being *labeled* rather than the probability of being the positive class, and a propensity function (in our case, the burst labeling likelihood function) that always predicts that the instance is labeled, will perfectly model the observed data. Obviously, this is not the desired solution, as instead we want to encourage the burst likelihood function to be small for unlabeled positives and large otherwise. We thus need to weigh the output of the burst likelihood estimator by a positive value < 1. We expect the labeling likelihood of a given true positive instance to decrease as the number of directly preceding instances with a high probability of being labeled increases.

Thus, we now propose to weigh $\hat{q}_i$ by $(1-\lambda)^{(1+\lambda)m^*}$, where $0 < \lambda << 1$ and m is given by:

$$\begin{aligned} m^* &= \arg\max_m \sum_{j=1}^{m} f(x_{i-j}), \\ \text{s.t. } rnd(f(x_{i-1})) &= ... = rnd(f(x_{i-m})) = 1 \\ \text{and} s_{i-1} &= ... = s_{i-m} = 0, \end{aligned}$$

where rnd is the rounding function. Here, m^* therefore denotes the number of preceding instances that $\hat{f}$ predicts to be positive instances. This means $(1-\lambda)^{(1+\lambda)m^*}$ begins as a term close to 1 when the previous instance is predicted as a negative instance, but decreases towards 0 as the number of consecutive previous unlabeled instances that are likely to be true positives increases.

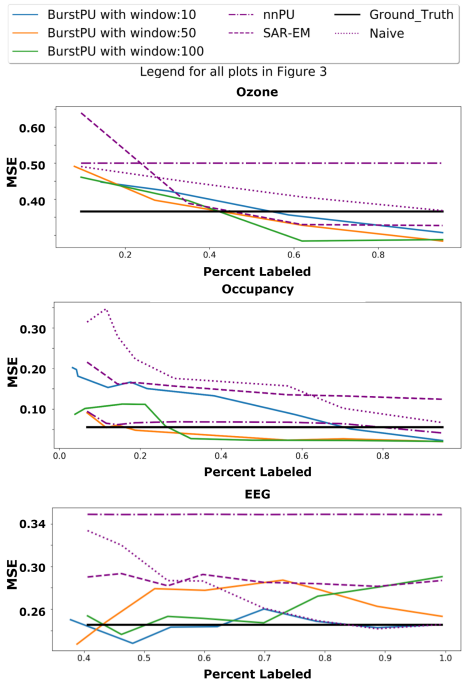


Fig. 3: MSE of BurstPU vs the state-of-the-art PU methods and baselines for the Ozone [16], Occupancy [17], and EEG [18] datasets. As we introduce various levels of unlabeling into these datasets in this experiment, the percent unlabeled corresponds to the ratio of labeled instances to true positive instances. Results are averaged over 5 runs.

V. Experimental Study

A. Datasets

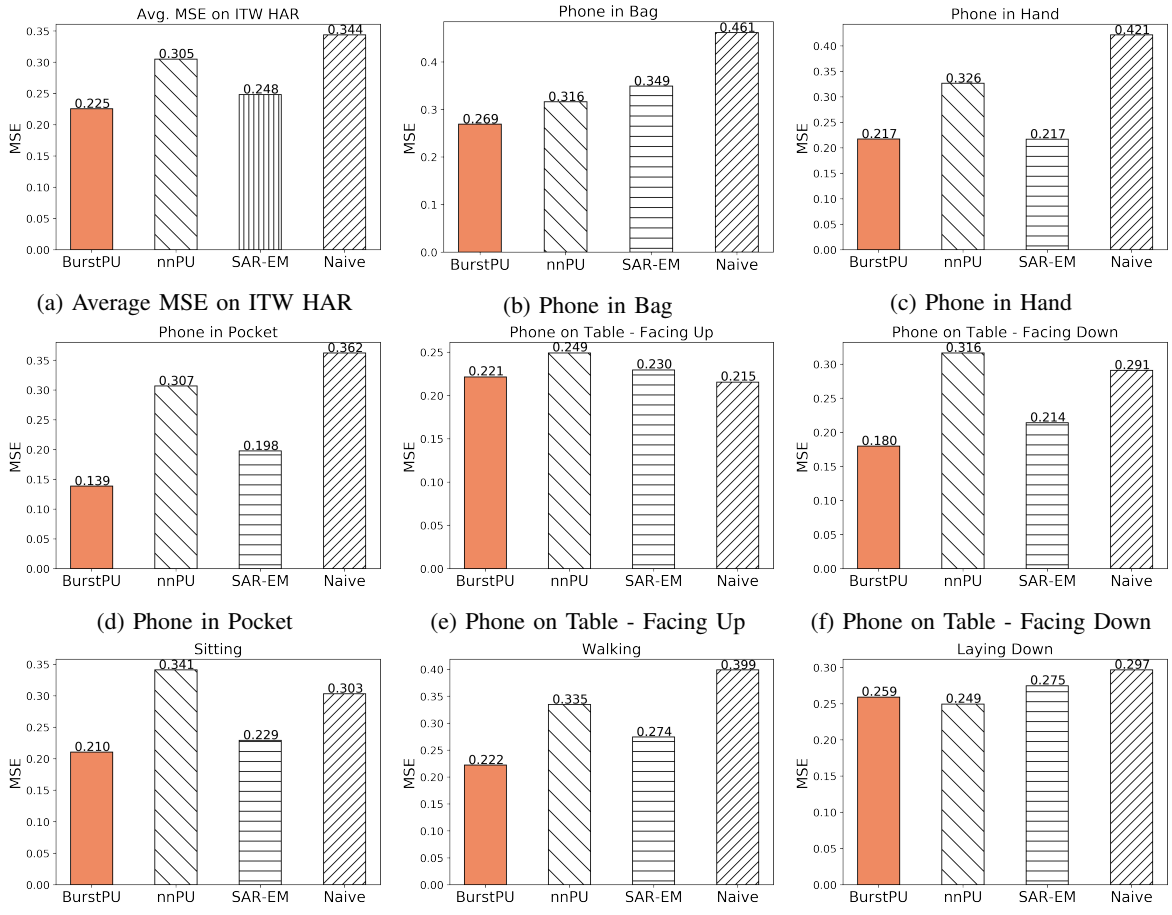
To study the ability of our BurstPU method for PU learning, we evaluate it on four real-world sequential datasets from the literature. For three of the datasets, all ground truth labels are known. For these datasets, we introduce various levels of unlabeling and compare our method against state-of-the-art methods for these unlabeling scenarios. The fourth dataset is a HAR dataset, and thus is already naturally a PU dataset. For this reason, we do not introduce additional mislabeling when performing analysis on this ITW HAR dataset. For all datasets, a 70-30 train-test split was used. For the 3 non-HAR datasets, the last 30% of the sequence was used. For our HAR dataset, the split was random and on the user level. We will release this split when release the HAR data pending IRB approval.

- `EEG Eye State [18]`: This dataset was collected in a controlled study. A participant wore an Emotiv EEG Neuroheadset which collected EEG data from the participant for 117 seconds while a video of the participants' eyes was recorded. The EEG data was later annotated with the eye state (i.e., eye open or closed) observed using the recorded video.=
- `Occupancy [17]`: This dataset corresponds to the binary classification task of determining whether or not a room in an office was occupied given light, temperature, humidity and CO_2 data. Pictures of the office were used by the researchers to assign the ground truth labels. .
- `Ozone [16]`: Atmospheric data was collected in Texas in the Houston, Galveston and Brazoria area daily from 1998 to 2004. Corresponding information indicating whether or not each day was classified as an "ozone action day" is provided. The associated features are statistical properties of atmospheric conditions, such as the peak wind speeds and temperatures at various times throughout the day. More details on the variety of features in this dataset were made available in [16].
- `ITW HAR [8]`: Our research group collected an *in-the-wild* (ITW) HAR dataset, where "in the wild" refers to the fact that the dataset was collected from participants as they went about their normal routines. Participants were asked to annotate a finite set of activities as they went about their day. Data was collected every minute, and thus participants could label their activities down to a 1-minute granularity. However, due to the burden of fine-grained labeling, participants were allowed to annotate their data at a granularity of their choosing; i.e., they could select an hour of data and apply the same label to the entire time period. This resulted in burst labeling behavior by the participants. For this analysis, 52 statistical features were computed from the accelerometer and gyroscope. These features are standard features computed in HAR studies, and are the ones used by [1] in their study.

B. Compared Methods

We compare BurstPU's performance to the following algorithms, which include the state-of-the-art PU learning methods as well as standard baselines commonly found in PU studies. As each method can use an arbitrary classifier for the classification step, in our experiments all methods use the same base classifier. We use a logistic regression classifier with the same hyperparameter settings used in Sci-Kit Learn's implementation of logistic regression in version 0.21.3 [19].

- *nnPU* [12]: nnPU is the leading method for training deep networks on PU data. nnPU corresponds to the process of minimizing the empirical PU risk given in Equation 1 under the SCAR assumption. Additionally, due to the property that the PU risk can be infinitely negative and is thus liable to overfit during training, nnPU clips the risk over the unlabeled instances to be no less than 0.
- *SAR-EM* [9]: SAR-EM corresponds to the leading PU method for learning under biased labeling. SAR-EM jointly trains a classifier and label likelihood model using an EM algorithm. Any machine learning or deep learning model that returns a prediction probability can be used within SAR-EM.
- *Naive Classifier*: We use a logistic regression classifier that treats all unlabeled instances as negative instances. In other words, this method makes the standard assumption that the labels available during training correspond to the true class. We expect this method to perform worse as the total percentage of unlabeled instances increases. As



(a) Average MSE on ITW HAR (b) Phone in Bag (c) Phone in Hand

(d) Phone in Pocket (e) Phone on Table - Facing Up (f) Phone on Table - Facing Down

(g) Sitting (h) Walking (i) Sleeping

Fig. 4: Comparative Study of BurstPU vs the other comparative methods on the ITW HAR dataset measured by MSE. (a) MSE averaged out across all 8 activities in ITW HAR dataset, while (b) to (i) instead report the average MSE for each of the particular classes; i.e., (b) for Phone-in-Bag activity, etc. For MSE, lower is better.

this is the most straightforward and naive way to address the PU problem, we expect the other PU methods to outperform this method.

- *Ground-Truth Classifier*: We apply a logistic regression classifier to three datasets where the ground truth labels are known. This method is thus able to perform standard binary classification, as it is trained on the true class. Generally, we expect PU methods to perform worse than this method as unlike this ground-truth classifier they are trained incomplete data. This gives a baseline of ideal performance.

C. Experimental Results

1) Performance of BurstPU for Various Levels of Unlabeling: In this experiment, we evaluate BurstPU's ability to perform PU learning under the burst labeling assumption. The goal is to train BurstPU on PU data, and then evaluate the performance of its classifier to perform positive-negative binary classification on testing data. We transformed the fully labeled datasets: *Occupancy*, *Ozone*, and *EEG*, described above into PU datasets by removing the labels from all negative and a subset of the positive instances.

To capture the burst labeling scenario, the unlabeling of the positive instances was done in bursts. To accomplish this, we unlabeled consecutive positive instances such that the average length of each applied unlabeled burst was 90 instances with a variance of 3 instances.

We varied the total number of unlabeled bursts to control the measure of unlabeling. We define the measure of unlabeling as the ratio of labeled instances to true positive instances; i.e., measure of unlabeling = $\frac{\sum_{s_i=1} s_i}{\sum_{y_i=1} y_i}$. If the measure of unlabeling is 1 then all positive instances are labeled, and if it is 0 no positive instances are labeled. We compared *BurstPU* with three different window sizes (10, 50, 100) against *nnPU*, *SAR-EM*, a naive classifier which treats unlabeled instances as negative instances, along with a standard classifier that is always given the true class labels during training.

Results are shown in Figure 3. BurstPU outperforms all other methods, and performs the best with a window size of 100. Counter-intuitively, both BurstPU and SAR-EM perform

	Without $\hat{T}_i$	With $\hat{T}_i$
Phone in Bag	0.304	**0.269**
Phone in Hand	0.234	**0.217**
Phone in Pocket	**0.134**	0.139
Phone on Table Facing Up	0.199	**0.198**
Phone on Table Facing Down	**0.161**	0.180
Sitting	0.211	**0.210**
Walking	0.230	**0.222**
Laying Down	0.262	**0.259**

TABLE II: The MSE for each class in the *ITW HAR* dataset for BurstPU with and without the hyperparameter $\hat{T}_i$ (described in Equation 1). Using this feature increases the performance of BurstPU for the majority of classes.

better than the classifier that was given perfectly labeled data while they were trained on data with moderate levels of unlabeling. However, we note that the phenomenon of PU methods outperforming standard positive-negative classifiers has been given theoretical justification by [20].

2) Classification with PU Data: In this experiment we evaluate the ability of BurstPU to perform classification on an real-world PU dataset, *ITW HAR*. We do not introduce additional unlabeling into the dataset as *ITW HAR* already has mislabeled data. We train all methods on the user-annotated data. We then evaluate all methods on a held-out test set that consists of 30% of the total dataset. The splits where conducted on a user level. As *ITW HAR* is a multi-label dataset, we train each method as a binary classifier for each activity separately. To evaluate the performance of each method we measure the MSE of the predicted class probabilities returned by each method with the class labels of the testing set. Results are shown in Figure 4. For all but one class in *ITW HAR*, BurstPU significantly outperforms all other compared methods. The next best method is *SAR-EM*. This indicates that there might be feature-level biases in the user's labeling behavior for certain classes in addition to the sequential labeling dependencies captured by BurstPU.

3) Importance of T_i: BurstPU's burst likelihood estimator is given an estimate of the number of instances since the closest previous predicted negative instance occurred, as described in section 4.1. In order to validate the importance of this input, we evaluate the performance of BurstPU on the *ITW HAR* dataset for a version of BurstPU with and without this input. Results are shown in Table II. For most classes in the *ITW HAR* dataset, including this extra input decreased the MSE and thus increased classification performance of BurstPU.

VI. Conclusion

We have identified burst labeling, a new PU unlabeling pattern that had not previously been identified. We proposed BurstPU, a framework for PU learning under the burst labeling scenario. Additionally, we provided a theoretical error bound for our BurstPU method. We also demonstrated its effectiveness through extensive experimental evaluation. Future research includes the development of new PU burst labeling learning algorithms not using EM as well as of PU learning under alternate bias types.

Acknowledgements: This work was funded by the DARPA WASH program HR001117S0032.

References

[1] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE Pervasive Computing*, vol. 16, no. 4, 2017.

[2] J. Bekker, *Learning from Positive and Unlabeled Data*. PhD thesis, KU Leuven, 2018.

[3] 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*, vol. 102, 2017.

[4] M. N. Nguyen, X.-L. Li, and S.-K. Ng, "Positive unlabeled learning for time series classification," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[5] I. Rodríguez-Moreno, J. M. Martínez-Otzeta, B. Sierra, I. Rodriguez, and E. Jauregi, "Video activity recognition: State-of-the-art," *Sensors*, vol. 19, no. 14, p. 3160, 2019.

[6] N. Aafaq, A. Mian, W. Liu, S. Z. Gilani, and M. Shah, "Video description: A survey of methods, datasets, and evaluation metrics," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–37, 2019.

[7] R. L. Winslow, S. Granite, and C. Jurado, "Waveformecg: a platform for visualizing, annotating, and analyzing ecg data," *Computing in science & engineering*, vol. 18, no. 5, pp. 36–46, 2016.

[8] H. Mansoor, W. Gerych, L. Buquicchio, K. Chandrasekaran, E. Agu, and E. Rundensteiner, "Delfi: Mislabelled human context detection using multi-feature similarity linking," in *2019 IEEE Visualization in Data Science (VDS)*, IEEE, 2019.

[9] J. Bekker and J. Davis, "Learning from positive and unlabeled data under the selected at random assumption," Journal of Machine Learning Research, 2018.

[10] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2008.

[11] M. Du Plessis, G. Niu, and M. Sugiyama, "Convex formulation for learning from positive and unlabeled data," in *International Conference on Machine Learning*, 2015.

[12] R. Kiryo, G. Niu, M. C. du Plessis, and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," in *Advances In Neural Information Processing Systems (NeurIPS)*, 2017.

[13] M. Kato, T. Teshima, and J. Honda, "Learning from positive and unlabeled data with a selection bias," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[14] S. Jain, M. White, and P. Radivojac, "Recovering true classifier performance in positive-unlabeled learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[15] L. A. Kontorovich, K. Ramanan, *et al.*, "Concentration inequalities for dependent random variables via the martingale method," *The Annals of Probability*, vol. 36, no. 6, 2008.

[16] K. Zhang and W. Fan, "Forecasting skewed biased stochastic ozone days: analyses, solutions and beyond," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 299–326, 2008.

[17] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models," *Energy and Buildings*, vol. 112, pp. 28–39, 2016.

[18] X. L. Zhang, H. Begleiter, B. Porjesz, W. Wang, and A. Litke, "Event related potentials during object recognition tasks," *Brain Research Bulletin*, vol. 38, no. 6, pp. 531–538, 1995.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[20] G. Niu, M. C. du Plessis, T. Sakai, Y. Ma, and M. Sugiyama, "Theoretical comparisons of positive-unlabeled learning against positive-negative learning," in *Advances In Neural Information Processing Systems (NeurIPS)*, pp. 1199–1207, 2016.