Online Positive and Unlabeled Learning

Chuang Zhang1, Chen Gong1,3, Tengfei Liu4, Xun Lu4, Weiqiang Wang4 and Jian Yang1,2
1PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, China
2Jiangsu Key Lab of Image and Video Understanding for Social Security
3The Department of Computing, HongKong Polytechnic University
4Ant Financial Services Group
chen.gong@njust.edu.cn

Abstract

Positive and Unlabeled learning (PU learning) aims to build a binary classifier where only positive and unlabeled data are available for classifier training. However, existing PU learning methods all work on a batch learning mode, which cannot deal with the online learning scenarios with sequential data. Therefore, this paper proposes a novel positive and unlabeled learning algorithm in an online training mode, which trains a classifier solely on the positive and unlabeled data arriving in a sequential order. Specifically, we adopt an unbiased estimate for the loss induced by the arriving positive or unlabeled examples at each time. Then we show that for any coming new single datum, the model can be updated independently and incrementally by gradient based online learning method. Furthermore, we extend our method to tackle the cases when more than one example is received at each time. Theoretically, we show that the proposed online PU learning method achieves low regret even though it receives sequential positive and unlabeled data. Empirically, we conduct intensive experiments on both benchmark and real-world datasets, and the results clearly demonstrate the effectiveness of the proposed method.

1 Introduction

Traditional supervised learning usually employs labeled positive and negative examples for model training. However, the labeled negative data are not always available in many real-world applications. This has led to the development of Positive and Unlabeled learning (PU learning) [Denis et al., 2005], which aims to train a binary classifier from only positive and unlabeled data without the existence of negative data. Here the unlabeled data could be either positive or negative, but their ground-truth labels remain unknown to the learning algorithm throughout the training stage. Due to its usefulness and effectiveness, PU learning has been widely used in many real-world applications, such as software clone detection [Wei and Li, 2018], remote-sensed hyperspectral image classification [Li et al., 2011], etc.

Given its broad applicability as mentioned above, PU learning has attracted intensive research attention in recent years. A variety of effective algorithms have been proposed, such as [Liu et al., 2002; Kiryo et al., 2017; Gong et al., 2019b], etc. Although these existing methods have received encouraging performance on various datasets or tasks, they all work on a batch learning or offline learning mode, which cannot deal with the online learning scenarios with sequential data. Unfortunately, it is quite often that the data are presented in sequence for massive practical applications, so traditional batch learning algorithms which require all training data to be simultaneously observed will not work. Therefore, online learning [Rosenblatt, 1958; Crammer and Singer, 2003] is proposed to process the training examples that arrive in a sequential order, where a learner aims to learn and update the optimal classifier for processing the future data at each step. In other words, the classifier is required to be updated incrementally for any new training examples, so that the online learning algorithms do not need to observe all training examples for classifier training.

To make PU models applicable to sequential data, in this paper, we propose a novel Online Positive and Unlabeled (“OPU” for short) learning algorithm. The key challenge for our online model designing is how to avoid the bias incurred by the absence of negative data, and to update the model incrementally according to the sequential positive and unlabeled examples. In this paper, we propose to overcome this challenge by exploring the unbiased estimator for positive and unlabeled learning [Du Plessis et al., 2015] and utilizing an Online Gradient Descent (OGD) [Zinkevich, 2003] method to optimize our model. Specifically, we cast OPU learning as a sequential Empirical Risk Minimization problem, in which different unbiased loss functions are elaborately designed for arriving positive and unlabeled data, respectively. Then, we show that, for any coming new data point, the model can be updated independently and incrementally by OGD. Besides, we present theoretical proof of regret that bounds the difference between the solution computed by our OPU learning and the optimal solution learned at hindsight. Moreover, experimental results on both benchmark and real-world datasets are presented, confirming the effectiveness of our OPU algorithms.
2 Related Work

This section briefly reviews some typical works on PU learning and online learning, as they are related to this paper.

2.1 PU Learning

PU learning has been a popular research topic in machine learning community. Up to now, a variety of PU learning models have been proposed, and they can typically be categorized into three types according to how the unlabeled data are handled.

The methods belonging to the first category deploy a two-step strategy which firstly identifies some reliable negative examples from the unlabeled set, and then employs the reliable negative examples as well as the original positive examples to train a traditional supervised classifier. In this strategy, the first stage is critical and various methods have been developed to find the potential negative examples, such as “Spy Technique” [Liu et al., 2002], l-DNF method [Yu et al., 2002], and K-means based prototype method [Xiao et al., 2011]. The methods belonging to the second category convert PU learning problem to a one-sided label noise learning problem, which directly treat the unlabeled examples as noisy negative ones. Concretely, the original positive examples in the unlabeled set are regarded as mislabeled, while no negative examples are mislabeled as positive ones, and then some effective label noise learning algorithms can be deployed to solve the PU learning problem. The representative works are [Lee and Liu, 2003], [Shi et al., 2018], and [Gong et al., 2019b]. The methods belonging to the last category impose different weights on the loss values incurred by positive examples and also the pseudo-labeling on treating unlabeled examples as positive examples and negative examples, and thus transferring PU learning problem into a cost-sensitive learning problem. The typical works include [Elkan and Noto, 2008; Du Plessis et al., 2014; Du Plessis et al., 2015; Kiryo et al., 2017].

Apart from above-mentioned methods, other representative PU models usually rely on larger-margin theory [Gong et al., 2019a], multi-manifold data structure [Gong et al., 2019c], label disambiguation [Zhang et al., 2019], and sequential minimal optimization [Sansone et al., 2018]. However, as mentioned in the introduction, all above methods are only designed for batch learning mode and cannot be applied to deal with sequential data.

In fact, [Li et al., 2009] has studied PU learning for the data on the fly. However, both their setting and target are very different from ours as they only treat current data as positive while regarding all historical data as unlabeled, to model the interest drift of online customers. Differently, we require that the real labels of positive and unlabeled data should be specified in advance and will not change during the entire online learning process, so that our model coincides with the standard PU learning requirements.

2.2 Online Learning

Online learning [Rosenblatt, 1958; Kivinen et al., 2004; Finn et al., 2019] aims to learn from the data arriving in a sequential order. In general, existing online learning algorithms mainly fall into two categories according to the model updating strategy, namely the first-order-based online learning and the second-order-based online learning.

The first-order-based online learning algorithms only exploit the first order feature information. Representative works include Perceptron [Rosenblatt, 1958; Freund and Schapire, 1999], Relaxed Online Maximum Margin Algorithm [Li and Long, 2000], and Passive-Aggressive algorithm [Crammer et al., 2006]. Apparently, the performances of these algorithms are limited, for only the first-order information is deployed. In recent years, some second-order-based online learning algorithms have been elaborately designed to improve the performance of first-order-based algorithms. Generally, the second-order-based algorithms can significantly outperform the first-order-based algorithms by exploring the second-order information, such as the covariance matrix of the feature information. Representative works are the second-order Perceptron [Cesa-Bianchi et al., 2005], Confidence-Weighted learning [Dredze et al., 2008], and Adaptive Regularization of Weights Learning [Crammer et al., 2009].

However, the aforementioned algorithms are not applicable to the PU learning problem studied in this paper, and this motivates us to seek for a novel online training strategy for tackling the OPU learning problem.

3 Preliminaries on Batch-mode PU Learning

In this section, we review the formal setting for traditional batch-mode PU learning, which helps to explain our designed online algorithm in Section 4.

Consider a binary classification problem, where $x \in \mathbb{R}^d$ denotes a $d$-dimensional pattern and $y \in \{1, -1\}$ is the corresponding class label. Let $p(x, y)$ be the underlying joint density of $(x, y)$, and the class-conditional densities regarding positive class and negative class can be written as $p_p(x) = p(x|y = 1)$ and $p_n(x) = p(x|y = -1)$, respectively, where $p(x)$ denotes the marginal density regarding unlabeled data. Furthermore, given $\pi = p(y = 1)$ as the positive class-prior probability, we have that $p(y = -1) = 1 - \pi$. Since an unlabeled dataset consists of positive and negative examples, we know that $p(x) = \pi p(x|y = 1) + (1 - \pi) p(x|y = -1)$.

Assume that we have a positive dataset $\mathcal{P}$ and an unlabeled dataset $\mathcal{U}$ independent and identically drawn from an unknown distribution $\mathcal{D}$ as

$$
\mathcal{P} := \{(x_i^p)^{n_p}_{i=1} \sim p(x|y = 1), \quad \mathcal{U} := \{(x_j^n)^{n_n}_{j=1} \sim p(x),
$$

where $n_p$ and $n_n$ are the size of positive dataset and unlabeled dataset, respectively. The goal of PU learning is to learn a classifier $g(x)$ that assigns a label $\hat{y}$ to a new pattern $x$ as $\hat{y} = \text{sign}(g(x))$. It has been widely acknowledged that the Bayes optimal classifier $g^*(x)$ can be obtained by minimizing the following classification risk, namely

$$
R(g) = \mathbb{E}_{(x, y) \sim p(x, y)}[\ell_{0-1}(Y g(X))] = \pi \mathbb{E}_p[\ell_{0-1}(g(X))] + (1 - \pi) \mathbb{E}_n[\ell_{0-1}(-g(X))],
$$

where $(X, Y)$ are corresponding random variables of $(x, y)$, $\mathbb{E}_p[.] = \mathbb{E}_x \mathbb{E}_{y \sim p(y|x)}[.]$, $\mathbb{E}_n[.] = \mathbb{E}_x \mathbb{E}_{y \sim p(y=x)}[.]$, and $\ell_{0-1}(\cdot)$ is the zero-one loss formulated as $\ell_{0-1}(z) = \frac{1}{2} - \frac{1}{2} \text{sign}(z)$ with $z$ being the variable.
Thanks to the availability of positive and negative examples in the traditional fully supervised learning, the expectations \( \mathbb{E}_p[\ell_0(g(X))] \) and \( \mathbb{E}_n[\ell_0(-g(X))] \) in Eq. (1) can be estimated through the corresponding example averages. However, there are no negative examples available for PU learning, therefore \( \mathbb{E}_n[\ell_0(-g(X))] \) cannot be estimated directly. To solve this problem, [Du Plessis et al., 2015] proposed to estimate \( \mathbb{E}_n[\ell_0(-g(X))] \) indirectly from weighted \( \mathbb{E}_p[\ell_0(-g(X))] \) and \( \mathbb{E}_a[\ell_0(-g(X))] \). Based on the fact that \( p(x) = \pi p(x|y = 1) + (1 - \pi) p(x|y = -1) \), we can obtain

\[
\mathbb{E}_n[\ell_0(-g(X))] = \pi \mathbb{E}_p[\ell_0(-g(X))] + (1 - \pi) \mathbb{E}_a[\ell_0(-g(X))],
\]
(2)

where \( \mathbb{E}_a[\cdot] \) denotes the expectation over \( p(x) \). Therefore, the risk \( R(g) \) in PU learning can be approximated by

\[
R(g) = \pi \mathbb{E}_p[\ell_0(g(X))] - \ell_0(-g(X)) - \pi \mathbb{E}_p[\ell_0(-g(X))] \\
+ \mathbb{E}_a[\ell_0(-g(X))].
\]

From the equation above, we can see that the risk for PU learning consists of a composite loss for positive examples \( \ell_0(g(X)) - \ell_0(-g(X)) \), and an ordinary loss for unlabeled examples, i.e., \( \ell_0(-g(X)) \).

[Du Plessis et al., 2015] showed that a loss function \( \ell(z) \) will lead to a convex PU learning model if it satisfies the linear-odd property, namely \( \ell(z) - \ell(-z) = -z \), and the feasible \( \ell(z) \) can be double hinge loss, square loss, and logistic loss. In our method, the performance of different loss functions with linear-odd property will be discussed in Section 6.

4 The Proposed OPU Method
Let us consider the problem setting of online positive and unlabeled classification task. Let \( \{(x_t, y_t)\}_{t=1}^T \) be a sequence of input examples under a potential distribution \( D \), where \( x_t \) is a pattern of \( d \) dimension received at the \( t \)-th time, \( y_t \in \{1, 0\} \) is the observed class label of corresponding pattern, and \( T \) is the number of training rounds across the whole training stage. Here, we take \( y = 1 \) for positive examples and \( y = 0 \) for unlabeled examples. The goal of OPU classification is to learn a linear classifier \( g(x_t) = w_t \cdot x_t \), where \( w_t \in \mathbb{R}^d \) is the weight vector at the \( t \)-th time during the training stage.

4.1 Basic OPU with Single Coming Datum
We cast our OPU learning as sequential Empirical Risk Minimization problem, of which the main idea is to propose an unbiased estimate for the risk induced by the received training examples at each time. Specifically, our goal is to develop an OPU learning algorithm which approximates the risk presented in Eq. (3).

Given a surrogate loss function with linear-odd property \( \ell(z) \), Eq. (3) turns to be a convex optimization problem:

\[
R(g) = \pi \mathbb{E}_p[\ell(g(X))] - \ell_0(-g(X)) + \mathbb{E}_a[\ell_0(-g(X))] \\
= \pi \mathbb{E}_p[-g(X)] + \mathbb{E}_a[\ell_0(-g(X))],
\]
(4)

Algorithm 1 Basic OPU with single coming datum

**Input**: The penalty parameter \( \lambda \).

1. Initialize \( w_0 \leftarrow 0 \); 
2. for \( t = 1, 2, \ldots, T \) do 
3. Receive a training example \( (x_t, y_t) \); 
4. Set learning rate \( \eta_t = \frac{1}{t} \); 
5. if \( y_t = 1 \) then 
6. Calculate the gradient \( \nabla_t = -\pi g'(x_t) + \lambda w_t \); 
else 
7. Calculate the gradient \( \nabla_t = \ell'(g(x_t)) + \lambda w_t \); 
9. end if 
10. Update \( w_t \leftarrow \Pi_{\mathcal{W}}(w_{t-1} - \eta_t \nabla_t) \); 
11. end for 

**Output**: The latest classifier parameter \( w_T \).

which can be rewritten as an average of a set of training examples, namely

\[
R(g) = -\frac{\pi}{n_p} \sum_{i=1}^{n_p} g(x_i) + \frac{1}{n_u} \sum_{j=1}^{n_u} \ell(-g(x_j)).
\]

Now, we reformulate the risk estimation for each individual training example received at the \( t \)-th time, i.e.,

\[
R_t(g) = -\pi \mathbbm{1}_{y_t=1} g(x_t) + \mathbbm{1}_{y_t=0} \ell(-g(x_t)),
\]
(6)

where \( \mathbbm{1}_{x} \) is an indicator function which equals to one if its argument is true, and zero otherwise. Using this representation, we can directly apply the gradient descent based online learning method to solve OPU learning problem. Formally, given a linear-in-parameter classifier \( g(x_t) = w_t \cdot x_t \), the regularized objective function of Eq. (6) can be written as

\[
f(w_t) = R_t(g) + \frac{\lambda}{2} \|w_t\|^2_2
\]

where the first term and the second term are the losses induced by the received positive and negative examples, and the last term is an \( L_2 \) regularizer to avoid overfitting with \( \lambda \) being a nonnegative penalty parameter.

Then, we consider the unbiased gradient \( \nabla_t \) of the above objective function in presence of the \( t \)-th coming data, which is given by

\[
\nabla_t = -\pi \mathbbm{1}_{y_t=1} g'(x_t) + \mathbbm{1}_{y_t=0} \ell'(-g(x_t)) + \lambda w_t,
\]
(8)

where \( g'(\cdot) \) and \( \ell'(\cdot) \) are the corresponding derivatives. Therefore, we can update \( w_t \leftarrow \Pi_{\mathcal{W}}(w_{t-1} - \eta_t \nabla_t) \) using a step size of \( \eta_t \), where \( \Pi_{\mathcal{W}}(w) \) is a projection step defined as \( \Pi_{\mathcal{W}}(w) = \arg \min_{w' \in \mathcal{W}} \|w - w'\|_2 \), with \( \mathcal{W} \) being a feasible set of \( w \). After a predetermined number \( T \) of iterations, we output the \( w_T \) in the last iteration. Algorithm 1 concludes the overall OPU algorithm with single coming datum.

4.2 Extended OPU with Multiple Coming Data
Practically, we may also meet the cases that the data come in groups, so we extend our basic OPU learning algorithm to enable it to process a set of examples for one time.

To be exact, given a small training set \( I_t = \{(x_i, y_i)\}_{i=1}^b \) with the size of \( b \) \((b > 1)\) at time \( t \) \((t = 1, 2, \ldots, T)\), the regularized objective function can be formulated as

\[
f_{I_t}(w_t) = R_{I_t}(g) + \frac{\lambda}{2} \|w_t\|^2_2.
\]
(9)
where $R_{It}(g)$ is the risk averages of corresponding positive and unlabeled examples, defined as
$$R_{It}(g) = \frac{1}{b} \sum_{i=1}^{b} (-\Pi_{y_i=1} g(x_i) + \Pi_{y_i=0} \ell(-g(x_i)))$$

Now, we rewrite the objective function $f_t(w_t)$ by adding a conservative constraint $\frac{\gamma t}{2} ||w_t - w_{t-1}||^2$ [Li et al., 2014], and obtain as
$$\phi_t(w_t) = f_t(w_t) + \frac{\gamma t}{2} ||w_t - w_{t-1}||^2$$
$$= \frac{1}{b} \sum_{i=1}^{b} (-\Pi_{y_i=1} g(x_i) + \Pi_{y_i=0} \ell(-g(x_i)))$$
$$+ \frac{\lambda}{2} ||w_t||^2 + \frac{\gamma t}{2} ||w_t - w_{t-1}||^2. \quad (10)$$

where $\lambda$ is a nonnegative trade-off parameter and $\gamma t$ is a conservative coefficient related to $\lambda$. The first term is the unbiased estimator for PU learning aiming to achieve full utilization of this the currently coming set, the second term is an $L_2$ regularizer to avoid overfitting, and the last term is a conservative constraint which limits dramatic changes of the weight vector to avoid overutilization.

For $(x_i, y_i) \in I_t, i = 1, \ldots, b$, by computing the unbiased gradient of the above approximate objective Eq. (10), we have
$$\nabla f_t = \frac{1}{b} \sum_{i=1}^{b} (-\Pi_{y_i=1} g'(x_i) + \Pi_{y_i=0} \ell'(-g(x_i)))$$
$$+ \lambda w_t + \gamma t (w_t - w_{t-1}). \quad (11)$$

By following the gradient descent based online learning method, the learning parameter can be effectively updated as $w_t \leftarrow \Pi_{[0]}(w_{t-1} - \eta t \nabla f_t)$, with $\eta t$ being the learning rate.

Furthermore, compared with the basic OPU algorithm with single coming datum, another benefit brought by the extended OPU algorithm is that an $O(1/b)$ variance reduction can be achieved if $b (b > 1)$ training examples are presented each time [Dekel et al., 2012].

5 Theoretical Analysis

In this section, we are going to analyze the regret bound of the proposed OPU learning method, which measures the difference between the cumulative loss of our predictions and the cumulative loss of the optimal predictor $w_*$. To be specific, we first bound the regret of our basic OPU algorithm which updates the model with one example per time, and then we bound the regret of the extended OPU algorithm which updates the model with $b (b > 1)$ arrived examples.

5.1 Regret for Basic OPU Algorithm

When only one example is processed at each time, we have the following theorem for the regret bound:

**Theorem 1.** Let $f_1, \ldots, f_T$ be a sequence of $\lambda$-strongly convex functions. Let $W$ be a closed convex set and define $\Pi_W(w) = \arg \min_{w \in W} ||w - w||$. Let $w_0, \ldots, w_T$ be a sequence of vectors such that $w_0 \in W$ and for $t \geq 0$, $w_t = \Pi_W(w_{t-1} - \eta_t \nabla f_t)$, with $\nabla f_t$ being the unbiased gradient of $f_t$ at $w_t$. Assume that for all $t$, $||\nabla f_t|| \leq G$, we have
$$\text{Regret}_T(f(w)) \leq \frac{C^2}{2\lambda}(1 + \log T). \quad (12)$$

**Proof.** Let $w_* \in \arg \min_{w \in W} \sum_{t=1}^{T} f_t(w)$. By recalling the definition of regret, we have
$$\text{Regret}_T(f(w)) = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(w_*). \quad (13)$$

Since $f_t$ is $\lambda$-strongly convex, we get the following inequality,
$$f_t(w_*) \geq f_t(w_t) + \nabla f_t^T (w_t - w_*) + \frac{\lambda}{2} ||w_t - w_*||^2. \quad (14)$$

Rearranging Eq. (14), we get
$$f_t(w_*) - f_t(w_t) \geq \nabla f_t^T (w_t - w_*) + \frac{\lambda}{2} ||w_t - w_*||^2. \quad (15)$$

Following Zinkevich’s analysis [Zinkevich, 2003], we are going to upper-bound $\nabla f_t^T (w_t - w_*)$. According to the properties of projections [Hazan et al., 2007], we have,
$$||w_t - w_*||^2 = ||\Pi (w_{t-1} - \eta_t \nabla f_t) - w_*||^2 \leq ||w_{t-1} - \eta_t \nabla f_t - w_*||^2. \quad (16)$$

Hence,
$$||w_t - w_*||^2 \leq ||w_{t-1} - w_*||^2 + \eta_t^2 ||\nabla f_t||^2 - 2\eta_t \nabla f_t^T (w_{t-1} - w_*). \quad (17)$$

After rearranging the above equation and using the assumption $||\nabla f_t|| \leq G$, we obtain
$$2\nabla f_t^T (w_{t-1} - w_*) \leq \frac{1}{\eta t} \left(||w_{t-1} - w_*||^2 - ||w_t - w_*||^2\right) + \eta_t C^2. \quad (18)$$

By comparing Eq. (14) and Eq. (17), and summing up Eq. (17) from $t = 1$ to $T$ with $\eta_t = 1/(\lambda t)$, we have
$$\sum_{t=1}^{T} f_t(w_t) - f_t(w_*) \leq \frac{1}{2} \sum_{t=1}^{T} ||w_t - w_*||^2 \left(\frac{1}{\eta t} - \frac{1}{\eta t + \lambda t}\right) + \frac{C^2}{2\lambda} \sum_{t=1}^{T} \eta_t \leq 0 + \frac{C^2}{2}\sum_{t=1}^{T} \frac{1}{\lambda t} \leq \frac{C^2}{2\lambda} (1 + \log T), \quad (19)$$

which concludes the proof. \qed

5.2 Regret for Extended OPU Algorithm

When a set of examples are processed at each time, we have the following theorem for the regret bound:

**Theorem 2.** Let $\{I_t\}_{t=1}^{T}$ be the received training sets of the learning algorithm, with each training set consisting of $(b \geq 1)$ training examples. Let $f_1, \ldots, f_T$ be a sequence of $\lambda$-strongly convex functions. Let $W$ be a feasible set of $w$, and $w_0, \ldots, w_T$ be a sequence of vectors such that $w_0 \in W$ and $w_* \in \arg \min_{w \in W} \sum_{t=1}^{T} f_t(w)$. Assume that for all $t$, $\sup_{w \in W} ||\nabla f_t(w_t) - \nabla f_t(w)||^2 \leq A^2$, we have
$$\text{Regret}_T(f_t(w)) \leq \frac{\lambda||w_t - w_*||^2}{2\sqrt{b}} + \frac{A^2(1 + \log T)}{\lambda b}. \quad (19)$$

**Proof.** By using the definition of regret, in this case we have
$$\text{Regret}_T(f_t(w)) = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(w_*). \quad (20)$$
According to Theorem 1 in [Li et al., 2014], assumed that $f_t(w_t)$ is $\lambda$-convex for all $t$ and the updating parameter is chosen as $\gamma_t = \gamma + \lambda t$, we have that for all $w_\ast \in \mathcal{W}$,\
$$
\sum_{t=1}^{T} (f_t(w_t) - f_t(w_\ast)) \leq \frac{\gamma}{2} \|w_\ast - w_0\|_2^2 + \frac{A^2}{b} \sum_{t=1}^{T} \frac{1}{\gamma_t}.
$$
(21)

For strongly convex objective function, one may choose $\gamma = (\lambda A)/\left(\sqrt{b}\right)|w_\ast - w_0|_2$ to minimize the upper bound in Eq. (21). Since the variance decreases with $O(1/b)$ for each set of coming data, we see that the choice of $\gamma$ in Eq. (21) with $\gamma = O(1/\sqrt{b})$ is appropriate. We can easily get the following aggregated regret bound with simple algebra,

$$
\sum_{t=1}^{T} (f_t(w_t) - f_t(w_\ast)) \\
\leq \frac{\gamma}{2} \|w_\ast - w_0\|_2^2 + \frac{A^2}{b} \sum_{t=1}^{T} \frac{1}{\gamma_t} \\
= \frac{\gamma}{2} \|w_\ast - w_0\|_2^2 + \frac{A^2}{b} \sum_{t=1}^{T} \frac{1}{\gamma + \lambda t} \\
\leq \frac{\gamma}{2} \|w_\ast - w_0\|_2^2 + \frac{A^2}{b} \sum_{t=1}^{T} \frac{1}{\gamma t} \\
\leq \frac{\lambda}{2} \frac{\|w_\ast - w_0\|_2^2}{b} + \frac{A^2(1 + \log T)}{\lambda b},
$$
(22)

which completes the proof. \qed

From Theorems 1 and 2, we see that the regret bound for our basic OPU algorithm is the order of $O(\log T)$, and that for our extended OPU is $O((\log T)/b)$, which improves the regret bound of the basic OPU algorithm with $O(1/b)$. Therefore, the models finally obtained by our method are very close to the ideal solution $w_\ast$.

### 6 Experiments

In this section, we first study the performance of the compared methods on benchmark and real-world datasets, and then investigate the parametric sensitivity of the pre-tuned parameter in our model.

#### 6.1 Compared Algorithms

In our experiments, two different loss functions with linear-odd property are discussed, i.e., double hinge loss $\ell_{DH}(z) = \max\left(-z, \max\left(0, \frac{1}{2} - \frac{1}{z}\right)\right)$ and square loss $\ell_{SL}(z) = \frac{1}{2}(z - 1)^2 - \frac{1}{4}$. The following methods are compared:

- **UPU**: Unbiased PU learning, a batch-mode PU learning algorithm [Du Plessis et al., 2015], which can be deemed as the performance upper bound of our OPU algorithm.
- **OPU$_{DH}$**: The proposed basic OPU algorithm with single coming datum in Sec. 4.1 using double hinge loss.
- **OPU$_{SL}$**: The proposed basic OPU algorithm with single coming datum in Sec. 4.1 using square loss.
- **\(\tilde{\text{OPU}}_{DH}\)**: The proposed extended OPU algorithm with multiple coming data in Sec. 4.2 using double hinge loss.
- **\(\tilde{\text{OPU}}_{SL}\)**: The proposed extended OPU algorithm with multiple coming data in Sec. 4.2 using square loss.

#### 6.2 Benchmark Datasets

We conduct experiments on a variety of benchmark datasets from OpenML machine learning repositories1. To be specific, four binary datasets are adopted for algorithm evaluation including Vote, Australian, Mushroom, and Phishing, and their brief information is presented in Table 1.

For each dataset, we randomly choose $r = 20\%$, 30\%, and 40\% positive examples as well as all negative examples as unlabeled and leave the rest positive examples as labeled. Under each $r$, we conduct five-fold cross validation on every compared method and report the average accuracy and standard deviation over the five independent implementations. As a result, each model under a certain implementation is trained with 80\% examples and then tested on the rest 20\% examples. Moreover, all data features are normalized to $[-1, 1]$ in advance, and the formation of training set is kept identical for all compared methods to ensure fair comparison.

In our experiments, the class prior of positive examples $\pi$ is assumed to be known during training for all the compared methods, which is also assumed by the existing PU learning works such as [Du Plessis et al., 2015; Kiryo et al., 2017; Gong et al., 2019b]. In practice, $\pi$ can be obtained by experience or some domain-specific prior knowledge. Besides, the parameters of every algorithm have been carefully tuned on the validation set to achieve the best performance. In our OPU, we choose the regularization parameter $\lambda$ from $\{10^{-6}, \ldots, 10^{2}\}$. For UPU, the regularization parameter $\lambda$ is chosen from $\{10^{-3}, \ldots, 10^{1}\}$. For our extended OPU algorithms including OPU$_{DH}$ and OPU$_{SL}$, 5\% training examples of the whole examples are selected to form the set of input data at each time.

The obtained test accuracies are reported in Table 1. We can find that the extended OPU algorithms OPU$_{DH}$ and OPU$_{SL}$ usually achieve higher accuracies than the basic OPU algorithms OPU$_{DH}$ and OPU$_{SL}$, and the OPU algorithms using double hinge loss often outperform the OPU algorithms using square loss. Besides, the performances of our OPU algorithms are very close to the performance of batch-mode PU learning method (i.e., UPU), and in some cases our OPU algorithm can perform even better than UPU, which suggests the effectiveness of the proposed method.

#### 6.3 Real-world Datasets

Here, we investigate the performance of the compared methods on image classification tasks. Concretely, CIFAR-10 [Krizhevsky and Hinton, 2009] and SVHN [Netzer et al., 2011] are chosen to evaluate their performance. CIFAR-10 consists of 60000 $32 \times 32$ natural images in 10 classes with each class containing 6000 images. We choose the images of transportation tools (‘airplane’, ‘auto mobile’, ‘ship’, and ‘truck’) as negative, and regard the images of animals (‘bird’, ‘cat’, ‘deer’, ‘dog’, ‘frog’, and ‘horse’) as positive. Therefore, there are 24000 positive examples and 36000 negative examples in total. SVHN consists of 99289 $32 \times 32$ digit images in 10 classes i.e., the digits ‘0’-’9’, where the negative set is formed by the digit images ‘1’-‘5’, and the rest digit images compose the positive set. As a result, we get 34699 positive

1https://www.openml.org/
examples and 64590 negative examples. Note that the training set and the test set are split in advance with 50000 training examples and 64590 negative examples. The best record among the compared methods is presented in Table 2, where we can see that although our OPU algorithms receive the sequential data sequentially, their performances are very similar to the compared UPU learning algorithm with all data observed for model training. Therefore, the proposed OPU algorithms are effective in handling real-world data.

6.4 Parametric Sensitivity

This section investigates the sensitivity of our OPU algorithms to the tuning parameter $\lambda$ and the positive class prior $\pi$ appearing in the objective function Eq. (7) and Eq. (10) of our OPU learning model. Specifically, we examine the test accuracy of the extended OPU algorithm with double hinge loss (i.e., $\text{OPU}_{DH}$) on the two real-world datasets when $r = 30\%$. The proposed $\text{OPU}_{SL}$, $\text{OPU}_{DH}$, and $\text{OPU}_{SL}$ all exhibit similar results to $\text{OPU}_{DH}$, so here we omit their parametric sensitivity curves due to the lack of space.

**Influence of $\lambda$:** The experimental results with $\lambda$ changing from $10^{-6}$ to $10^{2}$ are reported in Figure. 1 (a), we see that $\lambda$ is critical for our OPU algorithm to obtain satisfactory performance. To be specific, the choice of $\lambda = 10^{-2}$ usually leads to high classification accuracy.

**Influence of $\pi$:** As mentioned previously, the positive class prior $\pi$ might be unknown in practice, so it should be estimated from experience or some domain-specific prior knowledge. However, such estimation can be inaccurate, therefore we investigate how the classification accuracy is influenced by the inaccurate $\pi$. More specifically, we tested our $\text{OPU}_{DH}$ by replacing $\pi$ with inaccurate $\hat{\pi} \in \{0.6\pi, 0.8\pi, \ldots, 1.4\pi\}$ and inserting $\hat{\pi}$ to the learning method. The experimental results are presented in Figure. 1 (b), which suggests that the performance of the proposed method will not severely decrease when the estimation is slightly deviated from the real $\pi$.

7 Conclusion

This paper proposed a novel OPU learning paradigm to deal with the sequential positive and unlabeled data. We show that with an unbiased estimate for the loss induced by the received positive or unlabeled example at each time, our OPU model can be effectively solved by gradient based online learning methods. Furthermore, two OPU algorithms with different formations of input data are developed, and their regret bounds have also been theoretically proved. The experimental results on both benchmark and real-world datasets clearly demonstrate the effectiveness of the proposed method. Since our method relies on the class prior $\pi$, we may design an online strategy to estimate $\pi$ incrementally in the future.

Acknowledgements

This research is supported by NSF of China (Nos: 61973162, U1713208), NSF of Jiangsu Province (No: BK20171430), the Fundamental Research Funds for the Central Universities (No: 30918011319), the Summit of the Six Top Talents Program (No: DZXX027), the Young Elite Scientists Sponsorship Program by CAST (No: 2018QNR001), the Ant Financial Science Funds for Security Research of Ant Financial, and the Program for Changjiang Scholars.
References


[AR]