Online Positive and Unlabeled Learning

Chuang Zhang[1] , **Chen Gong** [1,3] , **Tengfei Liu**[4] , **Xun Lu**[4] , **Weiqiang Wang**[4] and **Jian Yang**[1,2]
[1]PCA 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
[2]Jiangsu Key Lab of Image and Video Understanding for Social Security
[3]The Department of Computing, HongKong Polytechnic University
[4]Ant 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 **O**nline **P**ositive and **U**nlabeled ("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 s-trategy, 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], 1-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 $\mathbf{x} \in \mathbb{R}^d$ denotes a d-dimensional pattern and $y \in \{1, -1\}$ is the corresponding class label. Let $p(\mathbf{x}, y)$ be the underling joint density of $(\mathbf{x}, y)$, and the class-conditional densities regarding positive class and negative class can be written as $p_{\mathrm{p}}(\mathbf{x}) = p(\mathbf{x}|y = 1)$ and $p_{\mathrm{n}}(\mathbf{x}) = p(\mathbf{x}|y = -1)$, respectively, where $p(\mathbf{x})$ 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 $p(\mathbf{x}) = \pi p(\mathbf{x}|y = 1) + (1 - \pi)p(\mathbf{x}|y = -1)$.

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

$$\mathcal{P} := \{\mathbf{x}_i^{\mathrm{p}}\}_{i=1}^{n_{\mathrm{p}}} \sim p(\mathbf{x}|y = 1), \quad \mathcal{U} := \left\{\mathbf{x}_j^{\mathrm{u}}\right\}_{j=1}^{n_{\mathrm{u}}} \sim p(\mathbf{x}),$$

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

$$\begin{aligned} R(g) &= \mathbb{E}_{(\mathbf{X},Y)\sim p(\mathbf{x},y)}\left[\ell_{0-1}(Yg(\mathbf{X}))\right] \\ &= \pi\mathbb{E}_{\mathrm{p}}[\ell_{0-1}(g(\mathbf{X}))] + (1-\pi)\mathbb{E}_{\mathrm{n}}[\ell_{0-1}(-g(\mathbf{X}))], \end{aligned} \tag{1}$$

where $(\mathbf{X}, Y)$ are corresponding random variables of $(\mathbf{x}, y)$, $\mathbb{E}_{\mathrm{p}}[\cdot] = \mathbb{E}_{\mathbf{X}\sim p_{\mathrm{p}}(x)}[\cdot]$, $\mathbb{E}_{\mathrm{n}}[\cdot] = \mathbb{E}_{\mathbf{X}\sim p_{\mathrm{n}}(x)}[\cdot]$, and $\ell_{0-1}(\cdot)$ is the zero-one loss formulated as $\ell_{0-1}(z) = \frac{1}{2} - \frac{1}{2}\,\mathrm{sign}(z)$ with z being the variable.

Thanks to the availability of positive and negative examples in the traditional fully supervised learning, the expectations $\mathbb{E}_{\mathrm{p}}\left[\ell_{0-1}(g(\mathbf{X}))\right]$ and $\mathbb{E}_{\mathrm{n}}\left[\ell_{0-1}(-g(\mathbf{X}))\right]$ in Eq. (1) can be estimated through the corresponding example averages. However, there are no negative examples available for PU learning, therefore $\mathbb{E}_{\mathrm{n}}\left[\ell_{0-1}(-g(\mathbf{X}))\right]$ cannot be estimated directly. To solve this problem, [Du Plessis *et al.*, 2015] proposed to estimate $\mathbb{E}_{\mathrm{n}}\left[\ell_{0-1}(-g(\mathbf{X}))\right]$ indirectly from weighted $\mathbb{E}_{\mathrm{p}}\left[\ell_{0-1}(-g(\mathbf{X}))\right]$ and $\mathbb{E}_{\mathrm{u}}\left[\ell_{0-1}(-g(\mathbf{X}))\right]$. Based on the fact that $p(\mathbf{x}) = \pi p(\mathbf{x}|y=1) + (1-\pi)p(\mathbf{x}|y=-1)$, we can obtain

$$\begin{aligned}\mathbb{E}_{\mathrm{u}}\left[\ell_{0-1}(-g(\mathbf{X}))\right] =& \pi\mathbb{E}_{\mathrm{p}}\left[\ell_{0-1}(-g(\mathbf{X}))\right] \\ &+ (1-\pi)\mathbb{E}_{\mathrm{n}}\left[\ell_{0-1}(-g(\mathbf{X}))\right],\end{aligned} \tag{2}$$

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

$$\begin{aligned}R(g) =& \pi\mathbb{E}_{\mathrm{p}}\left[\ell_{0-1}(g(\mathbf{X}))\right] \\ &+ \mathbb{E}_{\mathrm{u}}\left[\ell_{0-1}(-g(\mathbf{X}))\right] - \pi\mathbb{E}_{\mathrm{p}}\left[\ell_{0-1}(-g(\mathbf{X}))\right] \\ =& \pi\mathbb{E}_{\mathrm{p}}\left[\ell_{0-1}(g(\mathbf{X})) - \ell_{0-1}(-g(\mathbf{X}))\right] \\ &+ \mathbb{E}_{\mathrm{u}}\left[\ell_{0-1}(-g(\mathbf{X}))\right].\end{aligned} \tag{3}$$

From the equation above, we can see that the risk for PU learning consists of a composite loss for positive examples *i.e.*, $\ell_{0-1}(g(\mathbf{X})) - \ell_{0-1}(-g(\mathbf{X}))$, and an ordinary loss for unlabeled examples, *i.e.*, $\ell_{0-1}(-g(\mathbf{X}))$.

[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 $\{(\mathbf{x}_t, y_t)\,|t = 1, \ldots, T\}$ be a sequence of input examples under a potential distribution $\mathcal{D}$, where $\mathbf{x}_t \in \mathbb{R}^d$ 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 $g(\mathbf{x}_t) = \mathbf{w}_t^\top \cdot \mathbf{x}_t$, where $\mathbf{w}_t \in \mathbb{R}^d$ 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 loss 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 as

$$\begin{aligned}R(g) &= \pi\mathbb{E}_{\mathrm{p}}\left[\ell(g(\mathbf{X})) - \ell(-g(\mathbf{X}))\right] + \mathbb{E}_{\mathrm{u}}\left[\ell(-g(\mathbf{X}))\right] \\ &= \pi\mathbb{E}_{\mathrm{p}}[-g(\mathbf{X})] + \mathbb{E}_{\mathrm{u}}[\ell\left(-g\left(\mathbf{X}\right)\right)],\end{aligned} \tag{4}$$

Algorithm 1 Basic OPU with single coming datum

Input: The penalty parameter λ;
Initialize $\mathbf{w}_0 \leftarrow \mathbf{0}$;
for $t = 1, 2, \ldots, T$ **do**
  Receive a training example $(\mathbf{x}_t, y_t)$;
  Set learning rate $\eta_t = \frac{1}{\lambda t}$;
  if $y_t = 1$ **then**
    Calculate the gradient $\nabla_t = -\pi g'(\mathbf{x}_t) + \lambda\mathbf{w}_t$;
  else
    Calculate the gradient $\nabla_t = \ell'(-g(\mathbf{x}_t)) + \lambda\mathbf{w}_t$;
  end if
  Update $\mathbf{w}_t \leftarrow \Pi_{\mathcal{W}}(\mathbf{w}_{t-1} - \eta_t\nabla_t)$;
end for
Output: The latest classifier parameter $\mathbf{w}_T$.

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

$$R(g) = -\frac{\pi}{n_{\mathrm{p}}}\sum\nolimits_{i=1}^{n_{\mathrm{p}}} g(\mathbf{x}_i) + \frac{1}{n_{\mathrm{u}}}\sum\nolimits_{j=1}^{n_{\mathrm{u}}} \ell\left(-g\left(\mathbf{x}_j\right)\right). \tag{5}$$

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

$$R_t(g) = -\pi\mathbb{1}_{y_t=1}g(\mathbf{x}_t) + \mathbb{1}_{y_t=0}\ell\left(-g\left(\mathbf{x}_t\right)\right), \tag{6}$$

where $\mathbb{1}_{(\cdot)}$ 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 $g(\mathbf{x}_t) = \mathbf{w}_t^\top \cdot \mathbf{x}_t$, the regularized objective function of Eq. (6) can be written as

$$\begin{aligned}f(\mathbf{w}_t) &= R_t(g) + \frac{\lambda}{2}\|\mathbf{w}_t\|_2^2 \\ &= -\pi\mathbb{1}_{y_t=1}g(\mathbf{x}_t) + \mathbb{1}_{y_t=0}\ell\left(-g\left(\mathbf{x}_t\right)\right) + \frac{\lambda}{2}\|\mathbf{w}_t\|_2^2,\end{aligned} \tag{7}$$

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 λ being a nonnegative penalty parameter.

Then, we consider the unbiased gradient ∇_t of the above objective function in presence of the t-th coming data, which is given by

$$\nabla_t = -\pi\mathbb{1}_{y_t=1}g'(\mathbf{x}_t) + \mathbb{1}_{y_t=0}\ell'\left(-g\left(\mathbf{x}_t\right)\right) + \lambda\mathbf{w}_t, \tag{8}$$

where $g'(\cdot)$ and $l'(\cdot)$ are the corresponding derivatives. Therefore, we can update $\mathbf{w}_t \leftarrow \Pi_{\mathcal{W}}(\mathbf{w}_{t-1} - \eta_t\nabla_t)$ using a step size of η_t, where $\Pi_{\mathcal{W}}(\mathbf{w})$ is a projection step defined as $\Pi_{\mathcal{W}}(\mathbf{w}) = \arg\min_{\mathbf{w}'\in\mathcal{W}}\|\mathbf{w} - \mathbf{w}'\|$, with $\mathcal{W}$ being a feasible set of $\mathbf{w}$. After a predetermined number T of iterations, we output the $\mathbf{w}_T$ 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 $I_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^b$ with the size of b ($b > 1$) at time t ($t = 1, 2, \ldots, T$), the regularized objective function can be formulated as

$$f_{I_t}(\mathbf{w}_t) = R_{I_t}(g) + \frac{\lambda}{2}\|\mathbf{w}_t\|_2^2, \tag{9}$$

where $R_{I_t}(g)$ is the risk averages of corresponding positive and unlabeled examples, defined as

$$R_{I_t}(g) = \frac{1}{b}\sum_{i=1}^{b}(-\pi\mathbb{1}_{y_i=1}g(\mathbf{x}_i) + \mathbb{1}_{y_i=0}\ell(-g(\mathbf{x}_i))),$$

Now, we rewrite the objective function $f_{I_t}(\mathbf{w}_t)$ by adding a conservative constraint $\frac{\gamma_t}{2}\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2$ [Li *et al.*, 2014], and obtain as

$$\begin{aligned}\phi_{I_t}(\mathbf{w}_t) =& f_{I_t}(\mathbf{w}_t) + \frac{\gamma_t}{2}\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 \\ =&\frac{1}{b}\sum_{i=1}^{b}(-\pi\mathbb{1}_{y_i=1}g(\mathbf{x}_i) + \mathbb{1}_{y_i=0}\ell(-g(\mathbf{x}_i))) \\ &+ \frac{\lambda}{2}\|\mathbf{w}_t\|_2^2 + \frac{\gamma_t}{2}\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2, \end{aligned} \tag{10}$$

where λ is a nonnegative trade-off parameter and γ_t is a conservative coefficient related to λ. The first term is the unbiased loss 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 $(\mathbf{x}_i, y_i) \in I_t, i = 1, \dots, b$, by computing the unbiased gradient of the above approximate objective Eq. (10), we have

$$\begin{aligned}\nabla_{I_t} =& \frac{1}{b}\sum_{i=1}^{b}(-\pi\mathbb{1}_{y_i=1}g'(\mathbf{x}_i) + \mathbb{1}_{y_i=0}\ell'\left(-g\left(\mathbf{x}_i\right)\right)) \\ &+ \lambda\mathbf{w}_t + \gamma_t(\mathbf{w}_t - \mathbf{w}_{t-1}). \end{aligned} \tag{11}$$

By following the gradient descent based online learning method, the learning parameter can be effectively updated as $\mathbf{w}_t \leftarrow \Pi_{\mathcal{W}}(\mathbf{w}_{t-1} - \eta_t\nabla_{I_t})$, with η_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 $\mathcal{O}(1/b)$ 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 $\mathbf{w}_\star$. 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 $f_1, \dots, f_T$ be a sequence of λ-strongly convex functions. Let $\mathcal{W}$ be a closed convex set and define $\Pi_{\mathcal{W}}(\mathbf{w}) = \arg\min_{\mathbf{w}'\in\mathcal{W}}\|\mathbf{w} - \mathbf{w}'\|$. Let $\mathbf{w}_0, \dots, \mathbf{w}_T$ be a sequence of vectors such that $\mathbf{w}_0 \in \mathcal{W}$ and for $t \geq 0$, $\mathbf{w}_t = \Pi_{\mathcal{W}}\left(\mathbf{w}_{t-1} - \eta_t\nabla_t\right)$, with ∇_t being the unbiased gradient of f_t at $\mathbf{w}_t$. Assume that for all t, $\|\nabla_t\| \leq G$, we have*

$$\text{Regret}_T(f(\mathbf{w})) \leq \frac{G^2}{2\lambda}(1 + \log T). \tag{12}$$

Proof. Let $\mathbf{w}_\star \in \arg\min_{\mathbf{w}\in\mathcal{W}}\sum_{t=1}^{T}f_t(\mathbf{w})$. By recalling the definition of regret, we have

$$\text{Regret}_T(f(\mathbf{w})) = \sum_{t=1}^{T}f_t\left(\mathbf{w}_t\right) - \sum_{t=1}^{T}f_t\left(\mathbf{w}_\star\right). \tag{13}$$

Since f_t is λ-*strongly convex*, we get the following inequality,

$$f_t\left(\mathbf{w}_\star\right) \geq f_t\left(\mathbf{w}_t\right) + \nabla_t^\top\left(\mathbf{w}_t - \mathbf{w}_\star\right) + \frac{\lambda}{2}\left\|\mathbf{w}_t - \mathbf{w}_\star\right\|^2. \tag{14}$$

Rearranging Eq. (14), we get

$$f_t\left(\mathbf{w}_\star\right) - f_t\left(\mathbf{w}_t\right) \geq \nabla_t^\top\left(\mathbf{w}_t - \mathbf{w}_\star\right) + \frac{\lambda}{2}\left\|\mathbf{w}_t - \mathbf{w}_\star\right\|^2.$$

Following Zinkevichs analysis [Zinkevich, 2003], we are going to upper-bound $\nabla_t^\top\left(\mathbf{w}_t - \mathbf{w}_\star\right)$. According to the properties of projections [Hazan *et al.*, 2007], we have,

$$\begin{aligned}\left\|\mathbf{w}_t - \mathbf{w}_\star\right\|^2 &= \left\|\Pi\left(\mathbf{w}_{t-1} - \eta_t\nabla_t\right) - \mathbf{w}_\star\right\|^2 \\ &\leq \left\|\mathbf{w}_{t-1} - \eta_t\nabla_t - \mathbf{w}_\star\right\|^2. \end{aligned} \tag{15}$$

Hence,

$$\begin{aligned}\left\|\mathbf{w}_t - \mathbf{w}_\star\right\|^2 \leq& \left\|\mathbf{w}_{t-1} - \mathbf{w}_\star\right\|^2 + \eta_t^2\left\|\nabla_t\right\|^2 \\ &-2\eta_t\nabla_t^\top\left(\mathbf{w}_{t-1} - \mathbf{w}_\star\right). \end{aligned} \tag{16}$$

After rearranging the above equation and using the assumption $\|\nabla_t\| \leq G$, we obtain

$$2\nabla_t^\top(\mathbf{w}_{t-1} - \mathbf{w}_\star) \leq \frac{1}{\eta_t}(\|\mathbf{w}_{t-1} - \mathbf{w}_\star\|^2 - \|\mathbf{w}_t - \mathbf{w}_\star\|^2) + \eta_t G^2. \tag{17}$$

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

$$\begin{aligned}&\sum_{t=1}^{T}f_t\left(\mathbf{w}_t\right) - f_t\left(\mathbf{w}_\star\right) \\ \leq&\frac{1}{2}\sum_{t=1}^{T}\left\|\mathbf{w}_t - \mathbf{w}_\star\right\|^2\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \lambda\right) + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t \\ \leq& 0 + \frac{G^2}{2}\sum_{t=1}^{T}\frac{1}{\lambda t} \leq \frac{G^2}{2\lambda}(1 + \log T), \end{aligned} \tag{18}$$

which concludes the proof. □

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 ($b >$ 1) training examples. Let $f_{I_1}, \dots, f_{I_T}$ be a sequence of λ-strongly convex functions. Let $\mathcal{W}$ be a feasible set of $\mathbf{w}$, and $\mathbf{w}_0, \dots, \mathbf{w}_T$ be a sequence of vectors such that $\mathbf{w}_0 \in \mathcal{W}$ and $\mathbf{w}_\star \in \arg\min_{\mathbf{w}\in\mathcal{W}}\sum_{t=1}^{T}f_{I_t}(\mathbf{w})$. Assume that for all t, $\sup_{\mathbf{w}\in\mathcal{W}}\|\nabla f_{I_t}(\mathbf{w}_t) - \nabla f(\mathbf{w})\|_2^2 \leq A^2$, we have*

$$\text{Regret}_T(f_I(\mathbf{w})) \leq \frac{\lambda\left\|\mathbf{w}_\star - \mathbf{w}_0\right\|_2^2}{2\sqrt{b}} + \frac{A^2(1 + \log T)}{\lambda b}. \tag{19}$$

Proof. By using the definition of regret, in this case we have

$$\text{Regret}_T(f_I(\mathbf{w})) = \sum_{t=1}^{T}f_{I_t}\left(\mathbf{w}_t\right) - \sum_{t=1}^{T}f_{I_t}\left(\mathbf{w}_\star\right). \tag{20}$$

According to Theorem 1 in [Li *et al.*, 2014], assumed that $f_{I_t}(\mathbf{w}_t)$ is λ-*convex* for all t and the updating parameter is chosen as $\gamma_t = \gamma + \lambda t$, we have that for all $\mathbf{w}_\star \in \mathcal{W}$,

$$\sum\nolimits_{t=1}^{T}(f_{I_t}(\mathbf{w}_t)-f_I(\mathbf{w}_\star)) \leq \frac{\gamma}{2}\left\|\mathbf{w}_\star-\mathbf{w}_0\right\|_2^2+\frac{A^2}{b}\sum\nolimits_{t=1}^{T}\frac{1}{\gamma_t}. \quad (21)$$

For strongly convex objective function, one may choose $\gamma = (\lambda A)/(\sqrt{b}\left\|\mathbf{w}_\star - \mathbf{w}_0\right\|_2)$ to minimize the upper bound in Eq. (21). Since the variance decreases with $\mathcal{O}(1/b)$ for each set of coming data, we see that the choice of γ in Eq. (21) with $\gamma = \mathcal{O}(1/\sqrt{b})$ is appropriate. We can easily get the following aggregated regret bound with simple algebra,

$$\begin{aligned}
&\sum\nolimits_{t=1}^{T}(f_{I_t}(\mathbf{w}_t) - f_I(\mathbf{w}_\star)) \\
&\leq \frac{\gamma}{2}\left\|\mathbf{w}_\star - \mathbf{w}_0\right\|_2^2 + \frac{A^2}{b}\sum\nolimits_{t=1}^{T}\frac{1}{\gamma_t} \\
&= \frac{\gamma}{2}\left\|\mathbf{w}_\star - \mathbf{w}_0\right\|_2^2 + \frac{A^2}{b}\sum\nolimits_{t=1}^{T}\frac{1}{\gamma + \lambda t} \\
&\leq \frac{\gamma}{2}\left\|\mathbf{w}_\star - \mathbf{w}_0\right\|_2^2 + \frac{A^2}{b}\sum\nolimits_{t=1}^{T}\frac{1}{\lambda t} \\
&\leq \frac{\lambda\left\|\mathbf{w}_\star - \mathbf{w}_0\right\|_2^2}{2\sqrt{b}} + \frac{A^2(1+\log T)}{\lambda b},
\end{aligned} \quad (22)$$

which completes the proof. □

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

6 Experiments

In this section, we first study the performance of the proposed method 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 $\ell_{DH}(z) = \max\left(-z, \max\left(0, \frac{1}{2} - \frac{1}{2}z\right)\right)$ and square loss $\ell_{SL}(z) = \frac{1}{4}(z-1)^2 - \frac{1}{4}$. 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.
- $\overline{\textbf{OPU}}_{DH}$: The proposed extended OPU algorithm with multiple coming data in Sec. 4.2 using double hinge loss.
- $\overline{\textbf{OPU}}_{SL}$: 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 repositories[1]. 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 π 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, π 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 λ from $\{10^{-6}, \ldots, 10^{2}\}$. For UPU, the regularization parameter λ is chosen from $\{10^{-3}, \ldots, 10^{1}\}$. For our extended OPU algorithms including $\overline{\text{OPU}}_{DH}$ and $\overline{\text{OPU}}_{SL}$, 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 $\overline{\text{OPU}}_{DH}$ and $\overline{\text{OPU}}_{SL}$ 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 32×32 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 $32 * 32$ 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

[1]https://www.openml.org/

Dataset	(n, d)	r	UPU	OPU_{DH}	OPU_{SL}	$\overline{\text{OPU}}_{DH}$	$\overline{\text{OPU}}_{SL}$
Vote	(435, 16)	20%	0.920±0.087	0.902±0.029	0.887±0.024	0.908±0.008	**0.911±0.022**
		30%	0.940±0.012	0.894±0.025	0.864±0.040	**0.908±0.026**	0.903±0.029
		40%	0.959±0.006	0.889±0.026	0.877±0.041	**0.901±0.024**	0.898±0.038
Australian	(690, 14)	20%	0.772±0.045	0.808±0.012	0.758±0.102	**0.864±0.023**	0.844±0.016
		30%	0.827±0.025	0.832±0.047	0.720±0.104	**0.864±0.037**	0.850±0.027
		40%	0.843±0.027	0.743±0.125	0.621±0.120	**0.861±0.031**	0.724±0.110
Mushroom	(8124, 112)	20%	0.857±0.013	0.810±0.004	0.787±0.105	**0.836±0.078**	0.816±0.067
		30%	0.800±0.006	0.766±0.056	0.768±0.028	**0.825±0.036**	0.789±0.046
		40%	0.753±0.008	0.740±0.012	0.751±0.014	**0.840±0.067**	0.762±0.037
Phishing	(11055, 68)	20%	0.867±0.010	0.827±0.007	0.842±0.012	**0.856±0.005**	0.856±0.017
		30%	0.893±0.013	0.838±0.003	0.835±0.004	**0.881±0.004**	0.871±0.012
		40%	0.910±0.009	0.826±0.004	0.812±0.007	**0.890±0.004**	0.867±0.007

Table 1: The accuracies of various methods on four OpenML benchmark datasets when $r = 20\%$, 30%, and 40%. The best record among the proposed OPU algorithms under each r is marked in **bold**. n and d are the amount of training data and feature dimension accordingly.

Dataset	r	UPU	OPU_{DH}	OPU_{SL}	$\overline{\text{OPU}}_{DH}$	$\overline{\text{OPU}}_{SL}$
CIFAR-10	20%	0.848	0.810	0.840	0.824	**0.844**
	30%	0.842	0.827	0.792	**0.832**	0.803
	40%	0.836	**0.827**	0.799	**0.827**	0.815
SVHN	20%	0.838	0.833	0.783	**0.835**	0.812
	30%	0.846	0.808	0.798	**0.818**	0.805
	40%	0.847	0.820	0.813	**0.828**	0.819

Table 2: The test accuracies on the adopted real-world datasets. The best record among the proposed OPU algorithms is marked in **bold**.

examples and 64590 negative examples. Note that the training set and the test set are split in advance with 50000 training examples and 10000 test examples for *CIFAR-10*, and 73257 training examples and 20632 test examples for *SVHN*.

In our experiments, we extract 512-dimensional GIST features for each image. Similar to the previous experiments, for each dataset, the situations $r = 20\%, 30\%,$ and 40% are studied. The parameters of every method have been carefully tuned to achieve the best performance. The test accuracies achieved by the compared methods are 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 λ and the positive class prior π 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.*, $\overline{\text{OPU}}_{DH}$) on the two real-world datasets when $r = 30\%$. The proposed $\overline{\text{OPU}}_{SL}$, OPU_{DH}, and OPU_{SL} all exhibit similar results to $\overline{\text{OPU}}_{DH}$, so here we omit their parametric sensitivity curves due to the lack of space.

Influence of λ: The experimental results with λ changing from 10^{-6} to 10^{0} are reported in Figure. 1 (a), we see that λ 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 π: As mentioned previously, the positive class prior π 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 π. More specifically, we tested our $\overline{\text{OPU}}_{DH}$ by replacing π 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 π.

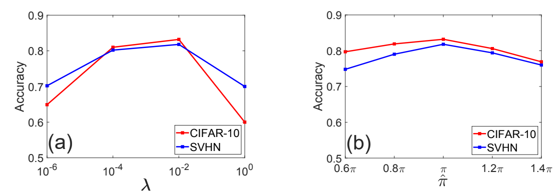

Figure 1: The parametric sensitivity of λ (a), and π (b) for $\overline{\text{OPU}}_{DH}$ on *CIFAR-10* and *SVHN* datasets.

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 π, we may design an online strategy to estimate π 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: 2018QNRC001), the Ant Financial Science Funds for Security Research of Ant Financial, and the Program for Changjiang Scholars.

References

[Cesa-Bianchi *et al.*, 2005] Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. A second-order perceptron algorithm. *SIAM Journal on Computing*, 34(3):640–668, 2005.

[Crammer and Singer, 2003] Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *JMLR*, 3(Jan):951–991, 2003.

[Crammer *et al.*, 2006] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *JMLR*, 7(Mar):551–585, 2006.

[Crammer *et al.*, 2009] Koby Crammer, Alex Kulesza, and Mark Dredze. Adaptive regularization of weight vectors. In *NeurIPS*, pages 414–422, 2009.

[Dekel *et al.*, 2012] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *JMLR*, 13(Jan):165–202, 2012.

[Denis *et al.*, 2005] François Denis, Rémi Gilleron, and Fabien Letouzey. Learning from positive and unlabeled examples. *Theoretical Computer Science*, 348(1):70–83, 2005.

[Dredze *et al.*, 2008] Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. In *ICML*, pages 264–271. ACM, 2008.

[Du Plessis *et al.*, 2014] Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *NeurIPS*, pages 703–711, 2014.

[Du Plessis *et al.*, 2015] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *ICML*, pages 1386–1394, 2015.

[Elkan and Noto, 2008] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *SIGKDD*, pages 213–220. ACM, 2008.

[Finn *et al.*, 2019] Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. *arXiv preprint arXiv:1902.08438*, 2019.

[Freund and Schapire, 1999] Yoav Freund and Robert E Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

[Gong *et al.*, 2019a] Chen Gong, Tongliang Liu, Jian Yang, and Dacheng Tao. Large-margin label-calibrated support vector machines for positive and unlabeled learning. *IEEE T-NNLS*, 2019.

[Gong *et al.*, 2019b] Chen Gong, Hong Shi, Tongliang Liu, Chuang Zhang, Jian Yang, and Dacheng Tao. Loss decomposition and centroid estimation for positive and unlabeled learning. *IEEE T-PAMI*, 2019.

[Gong *et al.*, 2019c] Chen Gong, Hong Shi, Jie Yang, and Jian Yanga. Multi-manifold positive and unlabeled learning for visual analysis. *IEEE T-CSVT*, 2019.

[Hazan *et al.*, 2007] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[Kiryo *et al.*, 2017] Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *NeurIPS*, pages 1675–1685, 2017.

[Kivinen *et al.*, 2004] Jyrki Kivinen, Alexander J Smola, and Robert C Williamson. Online learning with kernels. *IEEE T-SP*, 52(8):2165–2176, 2004.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[Lee and Liu, 2003] Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *ICML*, volume 3, pages 448–455, 2003.

[Li and Long, 2000] Yi Li and Philip M Long. The relaxed online maximum margin algorithm. In *NeurIPS*, pages 498–504, 2000.

[Li *et al.*, 2009] Xiao-Li Li, Philip S Yu, Bing Liu, and See-Kiong Ng. Positive unlabeled learning for data stream classification. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 259–270. SIAM, 2009.

[Li *et al.*, 2011] Wenkai Li, Qinghua Guo, and Charles Elkan. A positive and unlabeled learning algorithm for one-class classification of remote-sensing data. *T-GRS*, 49(2):717–725, 2011.

[Li *et al.*, 2014] Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J Smola. Efficient mini-batch training for stochastic optimization. In *SIGKDD*, pages 661–670. ACM, 2014.

[Liu *et al.*, 2002] Bing Liu, Wee Sun Lee, Philip S Yu, and Xiaoli Li. Partially supervised classification of text documents. In *ICML*, volume 2, pages 387–394. Citeseer, 2002.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[Rosenblatt, 1958] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[Sansone *et al.*, 2018] Emanuele Sansone, Francesco GB De Natale, and Zhi-Hua Zhou. Efficient training for positive unlabeled learning. *IEEE T-PAMI*, 2018.

[Shi *et al.*, 2018] Hong Shi, Shaojun Pan, Jian Yang, and Chen Gong. Positive and unlabeled learning via loss decomposition and centroid estimation. In *IJCAI*, pages 2689–2695, 2018.

[Wei and Li, 2018] Huihui Wei and Ming Li. Positive and unlabeled learning for detecting software functional clones with adversarial training. In *IJCAI*, pages 2840–2846, 2018.

[Xiao *et al.*, 2011] Yanshan Xiao, Bo Liu, Jie Yin, Longbing Cao, Chengqi Zhang, and Zhifeng Hao. Similarity-based approach for positive and unlabelled learning. In *IJCAI*, 2011.

[Yu *et al.*, 2002] Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. Pebl: positive example based learning for web page classification using svm. In *SIGKDD*, pages 239–248. ACM, 2002.

[Zhang *et al.*, 2019] Chuang Zhang, Dexin Ren, Tongliang Liu, Jian Yang, and Chen Gong. Positive and unlabeled learning with label disambiguation. In *IJCAI*, pages 4250–4256, 2019.

[Zinkevich, 2003] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.