Positive and Unlabeled Learning with Controlled Probability Boundary Fence

Changchun Li [1 2] **Yuanchao Dai** [1 2] **Lei Feng** [3] **Ximing Li** [1 2] **Bing Wang** [1 2] **Jihong Ouyang** [1 2]

Abstract

Positive and Unlabeled (PU) learning refers to a special case of binary classification, and technically, it aims to induce a binary classifier from a few labeled positive training instances and loads of unlabeled instances. In this paper, we derive a theorem indicating that the probability boundary of the asymmetric disambiguation-free expected risk of PU learning is controlled by its asymmetric penalty, and we further empirically evaluated this theorem. Inspired by the theorem and its empirical evaluations, we propose an easy-to-implement two-stage PU learning method, namely **P**ositive and **U**nlabeled **L**earning with **C**ontrolled **P**robability **B**oundary **F**ence (**PUL-CPBF**). In the first stage, we train a set of weak binary classifiers concerning different probability boundaries by minimizing the asymmetric disambiguation-free empirical risks with specific asymmetric penalty values. We can interpret these induced weak binary classifiers as a probability boundary fence. For each unlabeled instance, we can use the predictions to locate its class posterior probability and generate a stochastic label. In the second stage, we train a strong binary classifier over labeled positive training instances and all unlabeled instances with stochastic labels in a self-training manner. Extensive empirical results demonstrate that PUL-CPBF can achieve competitive performance compared with the existing PU learning baselines.

[1]College of Computer Science and Technology, Jilin University, China [2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, China [3]Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore. Correspondence to: Ximing Li <liximing86@gmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

The traditional binary classification, *a.k.a.,* **P**ositive and **N**egative (**PN**) learning, is a common case of supervised learning, whose goal, as its name suggests, is to induce a binary classifier from a set of labeled positive and negative training instances (Bekker & Davis, 2020). However, manually annotating sufficient numbers of positive training instances is often costly and even impractical in many real-world scenarios, therefore the datasets consisting of a few labeled positive training instances and loads of unlabeled ones can be available only (Yang et al., 2012; Ren et al., 2014; Hsieh et al., 2015). Take the diagnosis of Alzheimer's disease for example. Due to the infrequency and long incubation of Alzheimer's disease, one can only access a few diagnosed patients (**positive**) and loads of undiagnosed individuals (**unlabeled**), which can be either diseased or healthy. Learning with such kind of datasets refers to the paradigm of **P**ositive and **U**nlabeled (**PU**) learning, which has drawn increasing interest recently to meet the urgent practical demands (Bekker & Davis, 2020).

Many PU learning methods have been developed during the past decade (Chen et al., 2020a; Li et al., 2022; Zhao et al., 2022; Zhu et al., 2023). Naturally, a straightforward methodology is to induce the binary classifier by minimizing the disambiguation-free empirical risk, where all unlabeled instances are directly treated as pseudo-negative instances. However, it empirically suffers from a disambiguation-free boundary deviation phenomenon, where the boundary learned by the disambiguation-free empirical risk tends to deviate from the supervised boundary towards the positive side (Li et al., 2022), resulting in worse performance. To promote the disambiguation-free setting, some PU learning studies formulate a variety of unbiased risk estimators of PN learning, and the representatives include uPU (du Plessis et al., 2014; 2015), nnPU (Kiryo et al., 2017), and Dist-PU (Zhao et al., 2022) *etc.* In parallel, another branch of PU learning concentrates on estimating pseudo-labels for unlabeled instances, and using them to induce the binary classifier in a self-training manner (Liu et al., 2002; Zhang & Zuo, 2009; Chaudhari & Shevade, 2012; Luo et al., 2021; Wang et al., 2023).

Our motivation and contribution. Our motivation begins with an interest in the disambiguation-free boundary deviation phenomenon presented in (Li et al., 2022). Our original goal is to investigate whether this phenomenon can be supported by a certain theory. We analyzed it from the perspective of the Bayes optimal classifier expressed by the probability boundary. We derive a theorem indicating that the probability boundary of the asymmetric disambiguation-free expected risk is controlled by the asymmetric penalty. We conducted several preliminary experiments to evaluate this theorem, and the empirical results indicated that we can roughly control the probability boundary by adjusting the asymmetric penalty values, as presented in Fig.1.

Inspired by the theorem and its empirical evaluations, we propose a novel two-stage PU learning method, namely **P**ositive and **U**nlabeled **L**earning with **C**ontrolled **P**robability **B**oundary **F**ence (**PUL-CPBF**). Specifically, in the first stage, we train a set of weak binary classifiers concerning different probability boundaries by minimizing the asymmetric disambiguation-free empirical risks with specific asymmetric penalty values. We can interpret these induced weak binary classifiers as a probability boundary fence, and use them to predict each unlabeled instance, locate its class posterior probability, and generate a stochastic label. In the second stage, we train a strong binary classifier over labeled positive training instances and all unlabeled instances with stochastic labels in a self-training manner. Generally, PUL-CPBF is easy-to-implement, where any well-established tricks can be directly applied in the second stage. We conduct extensive experiments to evaluate the effectiveness of PUL-CPBF on benchmark datasets. Empirical results indicate that PUL-CPBF can achieve competitive performance compared with the existing PU learning baselines.

In a nutshell, the contributions of this paper are outlined as follows:

- We derive a theorem indicating that the probability boundary of the asymmetric disambiguation-free risk is controlled by the asymmetric penalty, and empirically evaluate the theorem.
- Inspired by the theorem, we propose a novel two-stage PU learning method named **PUL-CPBF**, which assigns stochastic labels for unlabeled instances by using the probability boundary fence.
- We conduct extensive experiments to evaluate PUL-CPBF on benchmark datasets, and the empirical results indicate that PUL-CPBF can be competitive with the existing PU learning baselines.

2. Formulation and Analysis

In this section, we introduce the problem formulations and analyze the asymmetric disambiguation-free expected risk of PU learning.

2.1. Problem Formulation

Formulation of PN learning. Broadly speaking, PN learning aims to induce a binary classifier from a set of labeled positive and negative training instances. Formally, let $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} = \{-1, +1\}$ be the d-dimensional feature space and label space, respectively. Consider a positive dataset $\mathcal{D}_p$ and a negative dataset $\mathcal{D}_n$ drawn from the class conditional distributions over positive and negative data, respectively:

$$\mathcal{D}_p \coloneqq \{(\mathbf{x}_i^p \in \mathcal{X}, +1)\}_{i=1}^{n_p}, \quad \mathbf{x}_i^p \overset{i.i.d.}{\sim} p_p(\mathbf{x});$$
$$\mathcal{D}_n \coloneqq \{(\mathbf{x}_i^n \in \mathcal{X}, -1)\}_{i=1}^{n_n}, \quad \mathbf{x}_i^n \overset{i.i.d.}{\sim} p_n(\mathbf{x}),$$

where $p_p(\mathbf{x}) = p(\mathbf{x}|y = +1)$, $p_n(\mathbf{x}) = p(\mathbf{x}|y = -1)$, and $p(\mathbf{x}|y)$ is the conditional distribution of $\mathbf{x}$ given y; $\mathbf{x}^p$ and $\mathbf{x}^n$ denote the labeled positive and negative training instances, respectively; n_p and n_n are the numbers of labeled positive and negative training instances, respectively. Let π denote the positive class prior $p(y = +1)$. PN learning induces a binary classifier $f\colon \mathbb{R}^d \to \mathcal{Y}$ from $\mathcal{D}_p \cup \mathcal{D}_n$. Its expected risk with respect to the data distribution is formulated as follows:

$$\mathcal{R}_{pn}^{\ell_{0-1}}(f) = \pi \mathbb{E}_p[\ell_{0-1}(f(\mathbf{x}^p))] + (1-\pi)\mathbb{E}_n[\ell_{0-1}(-f(\mathbf{x}^n))], \tag{1}$$

where $\ell_{0-1}(z) = -\frac{1}{2}\text{sign}(z) + \frac{1}{2}$ is the zero-one loss; $\mathbb{E}_p[\cdot]$ and $\mathbb{E}_n[\cdot]$ are the expectations with respect to $p_p(\mathbf{x})$ and $p_n(\mathbf{x})$, respectively. One can minimize the expected risk in Eq.1 to achieve the Bayes optimal classifier of PN learning, formulated below:

$$f^*_{\text{Bayes}}(\mathbf{x}) = \text{sign}\Big[p_\pi(y = +1|\mathbf{x}) - 0.5\Big]. \tag{2}$$

where $p_\pi(y = +1|\mathbf{x}) = \frac{\pi p_p(\mathbf{x})}{p(\mathbf{x})}$ is the class posterior probability, and $p(\mathbf{x}) = \pi p_p(\mathbf{x}) + (1 - \pi) p_n(\mathbf{x})$.

Formulation of PU learning. PU learning refers to a special case of binary classification, and it aims to induce a binary classifier from a few labeled positive training instances and loads of unlabeled instances. In this work, we concentrate on the selected completely at random assumption and the two-sample problem setting (Niu et al., 2016), also known as the case-control scenario (Bekker & Davis, 2020). Denote by n_p and n_u the sizes of labeled positive and unlabeled datasets. Note that as two random variables, n_p and n_u are fully independent in the two-sample problem setting (Niu et al., 2016). Without loss of generality, we assume that n_p and n_u are independently drawn from two Binomial distributions $B(N, \pi_p)$ and $B(N, \pi_u)$ respectively, where $\{N, \pi_p, \pi_u\}$ are the parameters of Binomial distributions. And consider a labeled positive dataset $\mathcal{D}_p$

and an unlabeled dataset $\mathcal{D}_u$ independently drawn as:

$$\begin{aligned} &n_p \sim B(N, \pi_p), \quad n_u \sim B(N, \pi_u), \\ &\mathcal{D}_p := \{(\mathbf{x}_i^p \in \mathcal{X}, +1)\}_{i=1}^{n_p}, \quad \mathbf{x}_i^p \overset{i.i.d.}{\sim} p_p(\mathbf{x}); \\ &\mathcal{D}_u := \{\mathbf{x}_i^u \in \mathcal{X}\}_{i=1}^{n_u}, \qquad \mathbf{x}_i^u \overset{i.i.d.}{\sim} p(\mathbf{x}), \end{aligned}$$

where $\mathbf{x}^u$ denotes the unlabeled instance; and n_u is the number of unlabeled instances. The aim of PU learning is to induce a binary classifier f: $\mathbb{R}^d \to \mathcal{Y}$ from $\mathcal{D}_p \cup \mathcal{D}_u$, however, it is intractable due to the lack of labeled negative training instances. A straightforward solution is to treat all unlabeled instances as pseudo-negative instances, rewritten as $\mathcal{D}_u := \{(\mathbf{x}_i^u \in \mathcal{X}, -1)\}_{i=1}^{n_u}$, and formulate the following disambiguation-free expected risk:

$$\mathcal{R}_{pu}^{\ell_{0-1}}(f) = \widehat{\pi}\mathbb{E}_p[\ell_{0-1}(f(\mathbf{x}^p))] + (1-\widehat{\pi})\mathbb{E}_u[\ell_{0-1}(-f(\mathbf{x}^u))], \tag{3}$$

where $\widehat{\pi} = \frac{\pi_p}{\pi_p + \pi_u}$; and $\mathbb{E}_u[\cdot]$ is the expectation with respect to $p(\mathbf{x})$.

To handle the asymmetric error, an asymmetric extension of disambiguation-free expected risk can be formulated as follows (Scott, 2012):

$$\begin{aligned} \mathcal{R}_{apu}^{\ell_{0-1}}(f) = (1-\beta)\widehat{\pi}\mathbb{E}_p[\ell_{0-1}(f(\mathbf{x}^p))] \\ + \beta(1-\widehat{\pi})\mathbb{E}_u[\ell_{0-1}(-f(\mathbf{x}^u))], \end{aligned} \tag{4}$$

where $\beta \in (0, 1)$ denotes the asymmetric penalty.

2.2. Analysis of Asymmetric Disambiguation-free Risk

We can expand and rearrange the risk of Eq.4 as follows:

$$\begin{aligned} \mathcal{R}_{\text{apu}}^{\ell_{0-1}}(f) &= (1-\beta)\widehat{\pi}\mathbb{E}_p[\ell_{0-1}(f(\mathbf{x}))] \\ &\quad + \beta(1-\widehat{\pi})\mathbb{E}_{\widehat{p}_n(\mathbf{x})}[\ell_{0-1}(-f(\mathbf{x}))] \\ &= ((1-\beta)\widehat{\pi} - \beta\pi(1-\widehat{\pi}))\mathbb{E}_p[\ell_{0-1}(f(\mathbf{x}))] \\ &\quad + \beta(1-\pi)(1-\widehat{\pi})\mathbb{E}_n[\ell_{0-1}(-f(\mathbf{x}))] \\ &\quad + \beta\pi(1-\widehat{\pi}). \end{aligned} \tag{5}$$

The derivation depends on the fact $\ell_{0-1}(z) + \ell_{0-1}(-z) = 1$. Note that the term $\beta\pi(1-\widehat{\pi})$ is a constant given a specific PU data distribution. Minimizing this risk is equivalent to minimizing a homogeneous form of the expected risk of PN learning. Accordingly, we can analyze its corresponding Bayes optimal classifier and present the following theorem.

Theorem 2.1. *Given $\widehat{\pi}$ and the positive class prior π in the asymmetric disambiguation-free expected risk in Eq.(5), and $0 < \beta < \frac{\widehat{\pi}}{\widehat{\pi}+\pi-\widehat{\pi}\pi}$, the following equation holds:*

$$f^*_{\text{Bayes-apu}} = \text{sign}\Big[p_\pi(y = +1|\mathbf{x}) - \alpha\Big],$$

where $\alpha = \frac{\beta\pi(1-\widehat{\pi})}{(1-\beta)\widehat{\pi}}$, and accordingly $\beta = \frac{\alpha\widehat{\pi}}{\alpha\widehat{\pi}+\pi-\pi\widehat{\pi}}$.

Proof. We first constrain the coefficients $((1-\beta)\widehat{\pi} - \beta\pi(1-\widehat{\pi}))$ and $\beta(1-\pi)(1-\widehat{\pi})$ in Eq.(5) to be positive, and then derive the result in Theorem 2.1 by employing the asymmetric expected risk of PN learning.

Firstly, by constraining the coefficients $((1-\beta)\widehat{\pi} - \beta\pi(1-\widehat{\pi})) > 0$ and $\beta(1-\pi)(1-\widehat{\pi}) > 0$ in Eq.(5) with $\beta, \pi, \widehat{\pi} \in (0, 1)$, we have

$$0 < \beta < \frac{\widehat{\pi}}{\widehat{\pi} + \pi - \widehat{\pi}\pi}. \tag{6}$$

Secondly, consider the asymmetric expected risk of PN learning suggested in (Scott, 2012):

$$\begin{aligned} \mathcal{R}_{\text{apn}}^{\ell_{0-1}} = (1-\alpha)\pi\mathbb{E}_p[\ell_{0-1}(f(\mathbf{x}^p))] \\ + \alpha(1-\pi)\mathbb{E}_n[\ell_{0-1}(-f(\mathbf{x}^n))]. \end{aligned} \tag{7}$$

Normalizing $\gamma = ((1-\beta)\widehat{\pi} - \beta\pi(1-\widehat{\pi})) + \beta(1-\pi)(1-\widehat{\pi})$ and $(1-\alpha)\pi + \alpha(1-\pi)$ to one and equating the expected risks of Eq.(7) and Eq.(5), we have

$$\frac{(1-\beta)\widehat{\pi} - \beta\pi(1-\widehat{\pi})}{\gamma} = \frac{(1-\alpha)\pi}{(1-\alpha)\pi + \alpha(1-\pi)}. \tag{8}$$

And by solving the above equation based on the given labeled positive proportion $\widehat{\pi}$ of the PU data and the positive class prior π, the Bayes optimal classifier is given by:

$$f^*_{\text{Bayes-apu}}(\mathbf{x}) = \text{sign}\left[\frac{\pi p_p(\mathbf{x})}{p(\mathbf{x})} - \alpha\right], \ \alpha = \frac{\beta\pi(1-\widehat{\pi})}{(1-\beta)\widehat{\pi}}. \tag{9}$$

Combining Eqs.(6) and (9), the final conclusion of Theorem 2.1 is given. □

Discussion. According to Theorem 2.1, we can interpret α as the probability boundary of the corresponding Bayes optimal classifier. And we can control α by adjusting β in the asymmetric disambiguation-free risk.

2.3. Empirical Evaluation of Theorem 2.1

To evaluate whether we can control α by adjusting β, we conducted several preliminary experiments on 2 benchmark PU datasets, *i.e.,* CIFAR-10-1 and F-MNIST-1. We employ a binary classifier f consisting of a pre-trained backbone and a classification layer. More setting details can be found in the experiment part.

Given any dataset $\mathcal{D}_p \cup \mathcal{D}_u$, we can specify the asymmetric disambiguation-free expected risk of Eq.4 as the following empirical risk:

$$\begin{aligned} \mathcal{L}_\beta(\mathcal{D}_l, \mathcal{D}_u) = &\frac{(1-\beta)\widehat{\pi}}{n_l}\sum_{i=1}^{n_l}\ell_{log}(f(\mathbf{x}_i^p)) \\ &+ \frac{\beta(1-\widehat{\pi})}{n_u}\sum_{i=1}^{n_u}\ell_{log}(-f(\mathbf{x}_i^u)), \end{aligned} \tag{10}$$

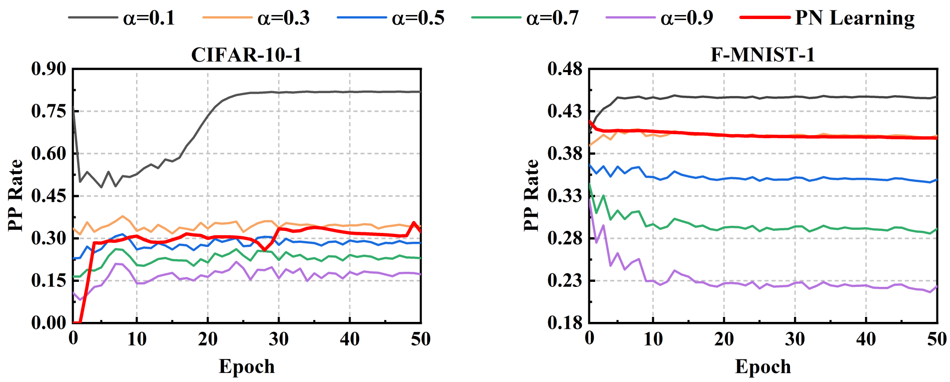

Figure 1: Rates of training instances Predicted as Positive (PP Rate), computed by the binary classifiers trained by the asymmetric disambiguation-free empirical risks with different α values and the empirical risk of PN learning. Best viewed in color.

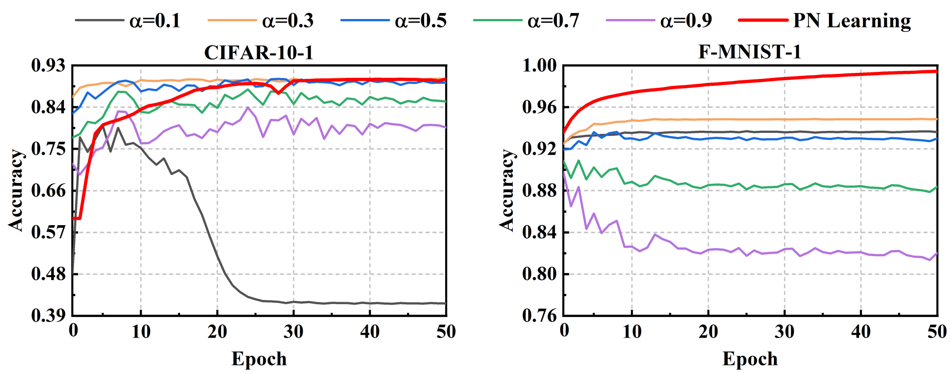

Figure 2: Empirical classification accuracy scores predicted by the binary classifiers trained by the asymmetric disambiguation-free empirical risks with different α values and the empirical risk of PN learning. Best viewed in color.

where $\ell_{log}(z) = \log(1 + exp(-z))$ is the logistic loss, used as the surrogate one for ℓ_{0-1} since it is intractable to minimize. For datasets whose class prior π is unknown, we apply the well-established KM2 method (Ramaswamy et al., 2016) to estimate their π. And because π_p and π_u are unknown, we can approximate $\widehat{\pi}$ with the proportion of labeled positive instances $\frac{n_p}{n_p+n_u}$ in practice. Then we can adjust the values of β to vary α from the range $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. For each synthetic PU dataset, we independently minimize these 5 asymmetric disambiguation-free empirical risks with the corresponding values of β. To evaluate different risks in the same feature space, we freeze the pre-trained backbone and optimize the parameter of the classification layer only. Additionally, to build reference results, we also optimize the classification layer by minimizing the empirical risk of PN learning over the same datasets with ground-truth labels.

The empirical results are presented in Fig.1. We present the rate of training instances Predicted as Positive (PP rate) for the binary classifier concerning different values of α. We can observe that the order of PP rate is strictly consistent with the order of α. That is to say, higher values of α imply less number of training instances can be predicted as positive. These empirical results suggest empirical support for Theorem 2.1.

Discussion on the results of the optimal β. In theory, the asymmetric disambiguation-free expected risk with the specific β (*i.e.,* optimal β) leading to $\alpha = 0.5$ is equivalent to the expected risk of PN learning. However, our preliminary empirical results (partially shown in Fig.2) indicated that the binary classifiers trained by their empirical risks showed a significant performance gap in most cases. Besides, the empirical results also showed that the binary classifier trained with the optimal β was not always optimal compared with the ones trained with other β. For example, $\alpha = 0.5$ performs worse than $\alpha = 0.3$ on FashionMNIST-1. This phenomenon may be caused by various reasons, such as the surrogate loss of the zero-one loss and less accurate estimations of π and $\widehat{\pi}$, resulting in the difference between theory and practice. Therefore we argue that inducing binary classifiers with the optimal β can be an efficient candidate of PU learning but not a stable one.

3. Proposed PUL-CPBF Method

In this section, we introduce the proposed **PUL-CPBF** method for PU learning.

3.1. Overview of PUL-CPBF

Inspired by Theorem 2.1 and its empirical evaluations, given any PU dataset, we can efficiently train a set of weak binary classifiers concerning different α values, denoted by f_α, by minimizing the asymmetric disambiguation-free empirical risks with specific values of β. Given these trained weak binary classifiers, we can apply them to form a probability boundary fence. For example, given the trained ones $f_{0.9}$ and $f_{0.7}$, an unlabeled instance $\mathbf{x}^u$ predicted by $f_{0.9}(\mathbf{x}^u) = -1$ and $f_{0.7}(\mathbf{x}^u) = +1$ must locate between their classification boundaries, so its class posterior probability $p_\pi(y = +1|\mathbf{x}^u)$ locates in the probability boundary range $(0.7, 0.9)$. Accordingly, we can assign $\mathbf{x}^u$ a **stochastic label** uniformly drawn from $(0.7, 0.9)$. With labeled positive training instances and all unlabeled instances with stochastic labels, we can then train a strong binary classifier in a self-training manner.

Upon the aforementioned ideas, we develop PUL-CPBF consisting of two stages, named generating stochastic labels with probability boundary fence and self-training with stochastic labels. In the following subsections, we will introduce the two stages in more detail.

3.2. Generating Stochastic Labels with Probability Boundary Fence

Given a PU dataset $\mathcal{D}_p \cup \mathcal{D}_u$, we independently train a set of weak binary classifiers $\{f_{\alpha_j}\}_{j=1}^m$ with different values of $\{\alpha_j\}_{j=1}^m$. To induce these weak binary classifiers in the same feature space, they share the same pre-trained backbone parameterized by $\mathbf{\Theta}$ and use specific classification layers parameterized by $\{\mathbf{W}_j\}_{j=1}^m$. In this stage, we freeze $\mathbf{\Theta}$ for efficiency. In terms of each f_{α_j}, we optimize $\mathbf{W}_j$ by minimizing the asymmetric disambiguation-free empirical risk with the specific value of β_j computed by Theorem 2.1 based on α_j:

$$\mathcal{L}_{\beta_j}(\mathcal{D}_l, \mathcal{D}_u; \mathbf{W}_j) = \frac{(1-\beta_j)\widehat{\pi}}{n_l} \sum_{i=1}^{n_l} \ell_{log}(f_{\alpha_j}(\mathbf{x}_i^p)) + \frac{\beta_j(1-\widehat{\pi})}{n_u} \sum_{i=1}^{n_u} \ell_{log}(-f_{\alpha_j}(\mathbf{x}_i^u)). \tag{11}$$

We can use all trained weak binary classifiers $\{f^*_{\alpha_j}\}_{j=1}^m$ to express the probability boundary fence. For each unlabeled instance $\mathbf{x}_i^u$, we use $\{f^*_{\alpha_j}\}_{j=1}^m$ to generate its stochastic label $y_i^s \in (0, 1)$. To be specific, we predict $\mathbf{x}_i^u$ by leveraging each one of $\{f^*_{\alpha_j}\}_{j=1}^m$, and then use the predictions to judge the **probability boundary range** (p_i^l, p_i^r), where its class posterior probability $p_\pi(y = +1|\mathbf{x}_i^u)$ is located:

$$(p_i^l, p_i^r) = \begin{cases} (0, \alpha_1), & \text{if } f^*_{\alpha_1}(\mathbf{x}_i^u) = -1; \\ (\alpha_j, \alpha_{j+1}), & \text{if } f^*_{\alpha_j}(\mathbf{x}_i^u) = +1,\ f^*_{\alpha_{j+1}}(\mathbf{x}_i^u) = -1; \\ (\alpha_m, 1), & \text{if } f^*_{\alpha_m}(\mathbf{x}_i^u) = +1. \end{cases} \tag{12}$$

We can then uniformly draw y_i^s from (p_i^l, p_i^r). Performing this process for all unlabeled instances, we can reform $\mathcal{D}_u$ as a dataset with stochastic labels, denoted by $\widehat{\mathcal{D}}_u := \{(\mathbf{x}_i^u, y_i^s)\}_{i=1}^{n_u}$.

3.3. Self-training with Stochastic Labels

In this stage, we train a strong binary classifier f^s over $\mathcal{D}_p \cup \widehat{\mathcal{D}}_u$ in a self-training manner. Specifically, f^s consists of the same pre-trained backbone parameterized by $\mathbf{\Theta}$ and a classification layer parameterized by $\mathbf{W}$. we optimize $\{\mathbf{\Theta}, \mathbf{W}\}$ by minimizing the following empirical risk:

$$\mathcal{L}(\mathcal{D}_l, \widehat{\mathcal{D}}_u; \mathbf{\Theta}, \mathbf{W}) = \frac{1}{n_l} \sum_{i=1}^{n_l} \ell_{ce}(f^s(\mathbf{x}_i^p), +1) + \frac{1}{n_u} \sum_{i=1}^{n_u} \ell_{ce}(f^s(\mathbf{x}_i^u), y_i^s) + \mathcal{R}, \tag{13}$$

where $\ell_{ce}(z_1, z_2) = -z_2 \log(z_1) - (1 - z_2) \log(1 - z_1)$ denotes the cross-entropy loss; and $\mathcal{R}$ is the regularization term. Due to the stochastic labels of unlabeled instances may be imprecise, we iteratively refine each stochastic label y_i^s by leveraging the sharpened prediction score of the current classifier as follows:

$$y_i^s = (1-\eta) y_i^s + \eta q_i, \quad q_i = \frac{(f^s(\mathbf{x}_i^u))^T}{(f^s(\mathbf{x}_i^u))^T + (1 - f^s(\mathbf{x}_i^u))^T}, \tag{14}$$

where $f^s(\mathbf{x}_i^u) \in (0, 1)$ denotes the prediction score of the current classifier for $\mathbf{x}_i^u$; η is a smoothing parameter; and T is the temperature parameter.

In this work, we specify $\mathcal{R}$ in Eq.13 by the consistency regularization term with data augmentation. Due to the space limit, we refer the readers to more details in (Sohn et al., 2020).

4. Related Works

Weakly-supervised learning surveyed by (Zhou, 2018) is classified into three kinds of weak supervision, including incomplete labels (van Engelen & Hoos, 2020; Li et al., 2021a; Pei et al., 2020; 2024; Yang et al., 2023b;a), inaccurate labels (Li et al., 2020b; Nguyen et al., 2020), and inexact labels (Feng et al., 2020a;b; Li & Wang, 2020; Li et al., 2020a; 2021b). Generally, PU learning belongs to the

Table 1: Specification of datasets and corresponding backbones. #Train: the number of training instances. #Test: the number of test instances.

Dataset	#Train	#Test	Input size	Backbone
F-MNIST	60,000	10,000	28×28	LeNet-5
CIFAR-10	50,000	10,000	3×32×32	7-layer CNN
STL-10	105,000	8,000	3×96×96	7-layer CNN
Alzheimer	5,121	1,279	3×224×224	ResNet-50

learning paradigm with incomplete labels, since no labeled negative instances are available. To the best of our knowledge, the existing PU learning methods can be grouped into two branches, including cost-sensitive methods and pseudo-labeling methods.

The cost-sensitive methods treat all unlabeled instances as noisy negative instances and propose various empirical risks of PU learning to correct the estimation bias. The straightforward methodology is the naive disambiguation-free empirical risk, however, it is a bias risk estimator of the empirical risk of PN learning. uPU rearranges the disambiguation-free empirical risk to formulate an unbiased risk estimator (du Plessis et al., 2014; 2015). nnPU, *i.e.,* the non-negative extension of uPU, further tackles the negative risk problem caused by overfitting, especially when applying deep learning backbones (Kiryo et al., 2017). Dist-PU drives the label distribution consistency between the predicted and ground-truth label distributions to mitigate the negative-prediction preference issue (Zhao et al., 2022). There are some other representative methods such as self-PU promoted by three self-supervision techniques (Chen et al., 2020c), PULD with margin-based label disambiguation (Zhang et al., 2019), and LLSVM performing label refinement by using a hat loss (Gong et al., 2019).

In parallel, the pseudo-labeling methods generate pseudo-labels for unlabeled instances before training the binary classifiers. Some early studies generate pseudo-labels by leveraging heuristic strategies based on traditional machine learning methods, such as Naïve Bayes (Liu et al., 2002), 1-DNF (Yu et al., 2002; 2004; Peng et al., 2008), kNN (Zhang & Zuo, 2009), and k-means (Chaudhari & Shevade, 2012), *etc.* Some recent methods promote the precision of pseudo-labels by further exploiting the well-established techniques. For example, PULNS formulates PU learning under the reinforcement learning framework, which incorporates a negative instance selector to generate pseudo-labels (Luo et al., 2021). Based on the observation that the predictive trends of unlabeled positive and negative instances present different patterns, HolisticPU transforms the pseudo-labeling process into an interesting predictive trend detection problem (Wang et al., 2023).

Besides the PU learning methods mentioned above, there are several other interesting methods. Some PU learning methods are built on the framework of adversarial generative networks (Hou et al., 2018; Chiaroni et al., 2018; Guo et al., 2020; Na et al., 2020; Hu et al., 2021), generating negative instances with the generator and then training the binary classifier with them. Additionally, some other PU learning methods (Wei et al., 2020; Li et al., 2022) turn to data augmentation techniques such as mixup (Zhang et al., 2018). For example, P^3Mix enriches and promotes the precise supervision of unlabeled instances by using a heuristic mixup partner selection method based on the observed disambiguation-free boundary deviation phenomenon (Li et al., 2022).

Orthogonal to the existing PU learning methods, in this work we suggest a new concept of probability boundary fence expressed by a set of weak binary classifiers trained with the asymmetric disambiguation-free empirical risks with specific asymmetric penalty values. We can use them to generate stochastic labels for unlabeled instances. The idea is supported by a proven theorem, and the proposed PUL-CPBF method can be easy-to-implement since any well-established tricks can be directly applied to promote PU learning with stochastic labels.

5. Experiments

In this section, we present the empirical results on benchmark PU learning settings.

5.1. Experimental Settings

Datasets In the experiments, we employ 3 prevalent benchmark datasets, including FashionMNIST (F-MNIST) (Xiao et al., 2017),[1] CIFAR-10 (Krizhevsky, 2016),[2] and STL-10 (Coates et al., 2011),[3] and a real-world dataset on Alzheimer diagnosis (Alzheimer).[4] The statistics of those

[1]https://github.com/zalandoresearch/fashion-mnist

[2]http://www.cs.toronto.edu/~kriz/cifar.html

[3]https://cs.stanford.edu/~acoates/stl10

[4]Dubey, S. Alzheimer's Dataset. Available online: https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images

Table 2: Positive and negative label groups of datasets and the statistics of those PU training sets. #Valid: the number of validation instances.

Dataset	Positive Class	Negative Class	π	n_l	n_u	#Valid
F-MNIST-1	0, 2, 4, 6	1, 3, 5, 7, 8, 9	0.4	1,000	59,500	500
F-MNIST-2	1, 3, 5, 7, 8, 9	0, 2, 4, 6	0.6	1,000	59,500	500
CIFAR-10-1	airplane, truck, automobile, ship	bird, cat, deer, dog, frog, horse	0.4	1,000	49,500	500
CIFAR-10-2	bird, cat, deer, dog, frog, horse	airplane, truck, automobile, ship	0.6	1,000	49,500	500
STL-10-1	0, 2, 3, 8, 9	1, 4, 5, 6, 7	–	1,000	104,500	500
STL-10-2	1, 4, 5, 6, 7	0, 2, 3, 8, 9	–	1,000	104,500	500
Alzheimer	Demented	Non-Demented	0.5	1,000	5,121	–

datasets are described in Table 1. Note that some of them contain multiple category labels, therefore following the processing of conventions (Kato et al., 2019), we pre-process them as binary classification datasets by grouping those category labels into two disjoint sets as positive and negative, respectively. For the training set of each dataset, we form its PU version(s) composed of a few labeled positive instances drawn from its positive dataset, a certain number of validation instances drawn from the full dataset, and unlabeled instances, *i.e.,* the remaining instances eliminating their labels. The statistics of those PU training sets are described in Table 2.

Baseline methods In the experiments, we employ 10 existing PU learning methods as baselines. They include uPU (du Plessis et al., 2014), nnPU (Kiryo et al., 2017), Self-PU (Chen et al., 2020c), VPU (Chen et al., 2020a), PULNS (Luo et al., 2021), PAN (Hu et al., 2021), Dist-PU (Zhao et al., 2022), P^{3}Mix (Li et al., 2022), Robust-PU (Zhu et al., 2023), and HolisticPU (Wang et al., 2023). In terms of P^{3}Mix, its two versions named P^{3}Mix-E and P^{3}Mix-C are all employed. Besides, we also use the PN learning method as a special baseline. In terms of all comparing methods, we apply the backbones for different datasets as follows: LeNet-5 for F-MNIST, 7-layer CNN for CIFAR-10 and STL-10, and ResNet-50 for Alzheimer. The details are presented in Table 1. Additionally, the positive class prior π is required by uPU, nnPU, Self-PU, and PUL-CPBF. Therefore, in terms of STL-10 whose π is unknown, we apply the well-established KM2 method (Ramaswamy et al., 2016) to estimate its π before training uPU, nnPU, Self-PU, and PUL-CPBF.

In terms of Robust-PU,[5] HolisticPU,[6] and PUL-CPBF, we run their source codes 5 times for each dataset and report the average results. Besides the three methods mentioned above, the results of all other comparing methods are from the public literature.

[5] https://github.com/woriazzc/robust-pu
[6] https://github.com/wxr99/HolisticPU

Evaluation metrics We employ the classification accuracy as the main criterion. And for the biasedly selected Alzheimer dataset, we also provide additional metrics, including F1 score, Recall, Precision, and Area Under ROC Curve (AUC) for a more comprehensive comparison. All metrics are calculated by using the Scikit-Learn tool (Pedregosa et al., 2011).[7]

Implementation details. We implement in-house code for PUL-CPBF by using Pytorch (Paszke et al., 2019). We employ the stochastic gradient descent optimizer and select the learning rate from $\{0.001, 0.0015, 0.002, 0.0025, 0.003\}$ and weight decay from $\{5e^{-5}, 1e^{-4}, 5e^{-4}, 1e^{-3}, 5e^{-3}\}$. The probability boundary range is set to $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, and the positive class prior of the PU data $\widehat{\pi}$ is estimated by the proportion of labeled positive training instances. Given specific π, $\widehat{\pi}$, and α, we then compute the corresponding value of β for each case. The epoch numbers of the first and second stages of PUL-CPBF are all set to 25. The batch sizes of the first and second stages are set to 32 and 16, respectively. We also clamp the logits between -10 and 10 to avoid the potential NaN error in Eqs.(11) and (13) following (Zhao et al., 2022). Besides, the backbones for each dataset are all pre-trained by contrastive learning (Chen et al., 2020b).

5.2. Results and Analysis

The empirical results of all baseline methods across 7 benchmark PU datasets are presented in Tables 3 and 4. Overall speaking, we can obverse that PUL-CPBF performs the highest scores compared with all PU learning baseline methods. It can be seen that PUL-CPBF consistently outperforms the recent competitor HolisticPU in all settings. For example, in terms of STL-10-1 and STL-10-2, the accuracy improvements of PUL-CPBF to HolisticPU are about 0.9 and 1.3, respectively. These results imply that the stochastic labels generated by PUL-CPBF are precise to the ground-truth ones. Compared with other recent competitors such as P^{3}Mix

[7] https://scikit-learn.org/stable/

Table 3: Results of classification accuracy (mean±std) on PU datasets formed by F-MNIST, CIFAR-10, and STL-10. The highest scores among PU learning methods are indicated in **bold**.

Method	F-MNIST-1	F-MNIST-2	CIFAR-10-1	CIFAR-10-2	STL-10-1	STL-10-2
uPU (du Plessis et al., 2015)	81.6±1.2	85.7±2.6	76.5±2.5	71.6±1.4	76.7±3.8	78.2±4.1
nnPU (Kiryo et al., 2017)	91.4±0.6	90.2±0.7	84.7±2.4	83.7±0.6	77.1±4.5	80.4±2.7
Self-PU (Chen et al., 2020c)	90.8±0.4	89.1±0.7	85.1±0.8	83.9±2.6	78.5±1.1	80.8±2.1
VPU (Chen et al., 2020a)	92.6±1.2	90.5±0.8	86.8±1.2	82.5±1.1	78.4±1.1	82.9±0.7
PULNS (Luo et al., 2021)	91.0±0.5	89.1±0.8	87.2±0.6	83.7±2.9	80.2±0.8	83.6±0.7
PAN (Hu et al., 2021)	87.7±2.4	89.9±3.2	87.0±0.3	82.8±1.0	77.7±2.5	79.8±1.4
Dist-PU (Zhao et al., 2022)	95.1±0.2	94.9±0.4	87.8±0.8	80.8±0.8	78.4±2.5	83.0±3.0
P^{3}Mix-E (Li et al., 2022)	92.6±0.4	91.8±0.2	88.2±0.4	84.7±0.5	80.2±0.9	83.7±0.7
P^{3}Mix-C (Li et al., 2022)	92.8±0.6	90.4±0.1	88.7±0.4	87.9±0.5	80.7±0.7	84.1±0.3
Robust-PU (Zhu et al., 2023)	90.0±0.5	85.5±0.7	80.0±0.6	85.2±1.1	79.6±0.9	80.4±0.8
HolisticPU (Wang et al., 2023)	96.2±0.1	96.0±0.3	91.0±0.3	90.4±0.5	82.5±0.5	84.0±1.2
PUL-CPBF (**Ours**)	**96.7±0.3**	**96.5±0.2**	**91.4±0.2**	**91.0±0.3**	**83.4±0.7**	**85.4±1.2**
PN learning	97.7±0.1	97.7±0.1	91.9±0.1	91.9±0.1	86.0±0.6	86.0±0.6

Table 4: The classification performance (mean±std) on the Alzheimer benchmark. The highest scores among PU learning methods are indicated in **bold**.

Method	F1 score	Accuracy	Recall	Precision	AUC
uPU (du Plessis et al., 2015)	67.6±2.8	68.5±2.2	66.1±6.1	69.7±3.5	73.8±2.9
nnPU (Kiryo et al., 2017)	68.6±3.2	68.3±2.1	69.5±7.2	68.0±2.3	72.9±2.8
Self-PU (Chen et al., 2020c)	72.1±1.1	70.9±0.7	75.4±5.1	69.3±2.5	75.9±1.8
VPU (Chen et al., 2020a)	70.2±1.1	67.4±0.7	76.7±3.6	64.7±1.1	73.1±0.9
Dist-PU (Zhao et al., 2022)	73.7±1.6	71.6±0.6	**80.1±5.1**	68.5±1.2	77.1±0.7
HolisticPU (Wang et al., 2023)	74.5±2.4	72.8±0.9	79.5±5.8	70.2±1.6	77.1±2.3
PUL-CPBF (**Ours**)	**75.3±1.4**	**73.4±0.7**	79.8±5.3	**71.4±1.3**	**81.1±0.9**
PN Learning	77.4±1.1	74.9±1.2	83.3±1.5	72.3±0.8	82.8±0.7

and Dist-PU, PUL-CPBF also significantly performs better in all settings, where, for example, the improvements of PUL-CPBF are achieved about $2.7 \sim 10.2$ across CIFAR-10-1 and CIFAR-10-2. From the dataset perspective, we can observe that PUL-CPBF is more stable than most PU learning baselines across both balanced PU datasets formed by STL-10 and relatively imbalanced ones formed by F-MNIST and CIFAR-10. These empirical results indirectly imply the robustness of PUL-CPBF supported by Theorem 2.1, which has taken the positive class prior into consideration. Surprisingly, the accuracy scores of PUL-CPBF are even approaching those of PN learning, where the performance gap is only about $0.5 \sim 2.6$ across all benchmark PU datasets. The competitive performance compared with PN learning further indicates the effectiveness of PUL-CPBF.

Furthermore, as illustrated in Table 4, our PUL-CPBF also consistently outperforms other methods in most cases and achieves comparable performance on the Alzheimer dataset to the SOTA baselines DistPU and HolisticPU, in which various regularization techniques and data augmentation strategies are employed. The balanced good performance of our PUL-CPBF on the biasedly selected real-world Alzheimer dataset across all evaluation metrics further demonstrates its effectiveness.

5.3. Evaluation of Probability Boundary Range

We empirically evaluate different probability boundary ranges of α across 6 benchmark PU datasets. The empirical results are presented in Table 5. Overall speaking, we can observe that our PUL-CPBF is insensitive to the probability boundary range of α, while the accuracy scores of dense probability boundary ranges are slightly higher than those of sparse ones. For example, the accuracy scores of $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ are consistently higher than those of $\{0.2, 0.5, 0.8\}$ and $\{0.1, 0.5, 0.9\}$. Considering the trade-off between efficiency and effectiveness, we suggest that the probability boundary range of α is set to $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ in practice.

Table 5: Classification accuracy results of different probability boundary ranges of α (mean$\pm$std) on PU datasets formed by F-MNIST, CIFAR-10, and STL-10. The highest scores are indicated in **bold**.

Probability boundary range	F-MNIST-1	F-MNIST-2	CIFAR-10-1	CIFAR-10-2	STL-10-1	STL-10-2
$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$	96.6±0.2	**96.6±0.3**	91.2±0.1	**91.1±0.3**	**83.5±0.6**	85.3±0.9
$\{0.1, 0.3, 0.5, 0.7, 0.9\}$	**96.7±0.3**	96.5±0.2	**91.4±0.2**	91.0±0.3	83.4±0.7	**85.4±1.2**
$\{0.2, 0.5, 0.8\}$	96.5±0.4	96.1±0.3	91.3±0.1	90.8±0.4	83.1±0.6	85.2±0.9
$\{0.1, 0.5, 0.9\}$	96.1±0.2	96.0±0.3	90.7±0.2	90.6±0.2	82.8±0.7	84.7±0.8

Table 6: Real running time (seconds) on PU datasets formed by F-MNIST, CIFAR-10, and STL-10.

Method	F-MNIST-1	F-MNIST-2	CIFAR-10-1	CIFAR-10-2	STL-10-1	STL-10-2
PUL-CPBF	213.47	214.34	333.28	333.53	421.39	428.77
HolisticPU	414.72	419.94	450.72	450.12	761.47	777.84
Dist-PU	47.97	47.47	298.84	299.05	386.05	397.36

5.4. Time Efficiency

We compare the running time between a cost-sensitive method Dist-PU,[8] a pseudo-labeling method HolisticPU, and our PUL-CPBF. For each comparing method, we independently perform 50 epochs in total regardless of classification results. Specifically, for HolisticPU and PUL-CPBF, the epoch numbers of the first and second stages are all set to 25, and for Dist-PU, the epoch numbers of the warmup and training stages are all set to 25. All experiments are performed on a server with one Nvidia RTX4090 GPU.

We show the running time in Table 6. It can be clearly seen that PUL-CPBF is more efficient than HolisticPU, while only costs a little more time compared with Dist-PU due to employing data augmentation. We consider that although in PUL-CPBF we induce several weak binary classifiers, we freeze the pre-trained backbones but optimize the classification layers only. Therefore we consider that PUL-CPBF is an efficient and practical candidate for PU learning.

6. Conclusion

In this paper, we analyze the asymmetric disambiguation-free expected risk of PU learning, and indicate the probability boundary can be controlled by the asymmetric penalty. In the preliminary experiments, we observed the consistency between the analysis and empirical results. Inspired by these findings, we propose to build a probability boundary fence expressed by a set of weak binary classifiers, trained with the asymmetric disambiguation-free empirical risks with specific asymmetric penalty values. We can then assign each unlabeled instance a stochastic label by using the predictions of these weak binary classifiers. Finally, we can train a strong binary classifier with these stochastic labels in a self-training manner. Upon these ideas, we suggest an easy-to-implement PU learning method named PUL-CPBF. We comprehensively evaluate PUL-CPBF by comparing with the existing PU learning baselines on benchmark datasets. The experimental results indicate the effectiveness and efficiency of PUL-CPBF.

[8] https://github.com/Ray-rui/Dist-PU-Positive-Unlabeled-Learning-from-a-Label-Distribution-Perspective

Acknowledgements

We would like to acknowledge support for this project from the National Science and Technology Major Project (No.2021ZD0112501), the National Natural Science Foundation of China (No.62276113), and China Postdoctoral Science Foundation (No.2022M721321).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bekker, J. and Davis, J. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4):719–760, 2020.

Chaudhari, S. and Shevade, S. K. Learning from positive and unlabelled examples using maximum margin clustering. In *International Conference Neural Information Processing*, pp. 465–473, 2012.

Chen, H., Liu, F., Wang, Y., Zhao, L., and Wu, H. A variational approach for learning from positive and unlabeled data. In *Neural Information Processing Systems*, 2020a.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607, 2020b.

Chen, X., Chen, W., Chen, T., Yuan, Y., Gong, C., Chen, K., and Wang, Z. Self-pu: Self boosted and calibrated positive-unlabeled training. In *International Conference on Machine Learning*, pp. 1510–1519, 2020c.

Chiaroni, F., Rahal, M., Hueber, N., and Dufaux, F. Learning with A generative adversarial network from a positive unlabeled dataset for image classification. In *IEEE International Conference on Image Processing*, pp. 1368–1372, 2018.

Coates, A., Ng, A. Y., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 215–223, 2011.

du Plessis, M. C., Niu, G., and Sugiyama, M. Analysis of learning from positive and unlabeled data. In *Neural Information Processing Systems*, pp. 703–711, 2014.

du Plessis, M. C., Niu, G., and Sugiyama, M. Convex formulation for learning from positive and unlabeled data. In *International Conference on Machine Learning*, pp. 1386–1394, 2015.

Feng, L., Kaneko, T., Han, B., Niu, G., An, B., and Sugiyama, M. Learning with multiple complementary labels. In *International Conference on Machine Learning*, pp. 3072–3081, 2020a.

Feng, L., Lv, J., Han, B., Xu, M., Niu, G., Geng, X., An, B., and Sugiyama, M. Provably consistent partial-label learning. In *Neural Information Processing Systems*, pp. 10948–10960, 2020b.

Gong, C., Liu, T., Yang, J., and Tao, D. Large-margin label-calibrated support vector machines for positive and unlabeled learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3471–3483, 2019.

Guo, T., Xu, C., Huang, J., Wang, Y., Shi, B., Xu, C., and Tao, D. On positive-unlabeled classification in GAN. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8382–8390, 2020.

Hou, M., Chaib-Draa, B., Li, C., and Zhao, Q. Generative adversarial positive-unlabeled learning. In *International Joint Conference on Artificial Intelligence*, pp. 2255–2261, 2018.

Hsieh, C., Natarajan, N., and Dhillon, I. S. PU learning for matrix completion. In *International Conference on Machine Learning*, pp. 2445–2453, 2015.

Hu, W., Le, R., Liu, B., Ji, F., Ma, J., Zhao, D., and Yan, R. Predictive adversarial learning from positive and unlabeled data. In *AAAI Conference on Artificial Intelligence*, pp. 7806–7814, 2021.

Kato, M., Teshima, T., and Honda, J. Learning from positive and unlabeled data with a selection bias. In *International Conference on Learning Representations*, 2019.

Kiryo, R., Niu, G., du Plessis, M. C., and Sugiyama, M. Positive-unlabeled learning with non-negative risk estimator. In *Neural Information Processing Systems*, pp. 1675–1685, 2017.

Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*. Technical report, University of Toronto, 2016.

Li, C., Li, X., and Ouyang, J. Learning with noisy partial labels by simultaneously leveraging global and local consistencies. In *ACM International Conference on Information and Knowledge Management*, pp. 725–734, 2020a.

Li, C., Li, X., and Ouyang, J. Semi-supervised text classification with balanced deep representation distributions. In *Annual Meeting of the Association for Computational Linguistics*, pp. 5044–5053, 2021a.

Li, C., Li, X., Ouyang, J., and Wang, Y. Learning with noisy partial labels by simultaneously leveraging global and local consistencies. In *ACM International Conference on Information and Knowledge Management*, pp. 903—-912, 2021b.

Li, C., Li, X., Feng, L., and Ouyang, J. Who is your right mixup partner in positive and unlabeled learning. In *International Conference on Learning Representations*, 2022.

Li, J., Socher, R., and Hoi, S. C. H. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020b.

Li, X. and Wang, Y. Recovering accurate labeling information from partially valid data for effective multi-label learning. In *International Joint Conference on Artificial Intelligence*, pp. 1373–1380, 2020.

Liu, B., Lee, W. S., Yu, P. S., and Li, X. Partially supervised classification of text documents. In *International Conference Machine Learning*, pp. 387–394, 2002.

Luo, C., Zhao, P., Chen, C., Qiao, B., Du, C., Zhang, H., Wu, W., Cai, S., He, B., Rajmohan, S., and Lin, Q. PULNS: positive-unlabeled learning with effective negative sample selector. In *AAAI Conference on Artificial Intelligence*, pp. 8784–8792, 2021.

Na, B., Kim, H., Song, K., Joo, W., Kim, Y., and Moon, I. Deep generative positive-unlabeled learning under selection bias. In *ACM International Conference on Information and Knowledge Management*, pp. 1155–1164, 2020.

Nguyen, D. T., Mummadi, C. K., Ngo, T., Nguyen, T. H. P., Beggel, L., and Brox, T. SELF: learning to filter noisy labels with self-ensembling. In *International Conference on Learning Representations*, 2020.

Niu, G., du Plessis, M. C., Sakai, T., Ma, Y., and Sugiyama, M. Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In *Neural Information Processing Systems*, pp. 1199–1207, 2016.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems*, pp. 8024–8035, 2019.

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

Pei, H., Yang, B., Liu, J., and Chang, K. C.-C. Active surveillance via group sparse bayesian learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1133–1148, 2020.

Pei, H., Xiong, Y., Wang, P., Tao, J., Liu, J., Deng, H., Ma, J., and Guan, X. Memory disagreement: A pseudo-labeling measure from training dynamics for semi-supervised graph learning. In *Proceedings of the ACM on Web Conference*, pp. 434–445, 2024.

Peng, T., Zuo, W., and He, F. SVM based adaptive learning method for text classification from positive and unlabeled documents. *Knowledge and Information Systems*, 16(3): 281–301, 2008.

Ramaswamy, H. G., Scott, C., and Tewari, A. Mixture proportion estimation via kernel embeddings of distributions. In *International Conference on Machine Learning*, pp. 2052–2060, 2016.

Ren, Y., Ji, D., and Zhang, H. Positive unlabeled learning for deceptive reviews detection. In *Conference on Empirical Methods in Natural Language Processing*, pp. 488–498, 2014.

Scott, C. Calibrated asymmetric surrogate losses. *Electronic Journal of Statistics*, 6:958–992, 2012.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C., Cubuk, E. D., Kurakin, A., and Li, C. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Neural Information Processing Systems*, 2020.

van Engelen, J. E. and Hoos, H. H. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

Wang, X., Wan, W., Geng, C., Li, S., and Chen, S. Beyond myopia: Learning from positive and unlabeled data through holistic predictive trends. In *Neural Information Processing Systems*, 2023.

Wei, T., Shi, F., Wang, H., Tu, W., and Li, Y. Mixpul: Consistency-based augmentation for positive and unlabeled learning. *arXiv preprint arXiv:2004.09388*, 2020.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yang, L., Shi, R., Zhang, Q., Niu, B., Wang, Z., Cao, X., and Wang, C. Self-supervised graph neural networks via low-rank decomposition. In *Neural Information Processing Systems*, 2023a.

Yang, L., Zhang, Q., Shi, R., Zhou, W., Niu, B., Wang, C., Cao, X., He, D., Wang, Z., and Guo, Y. Graph neural networks without propagation. In *Proceedings of the ACM Web Conference*, pp. 469–477, 2023b.

Yang, P., Li, X., Mei, J., Kwoh, C. K., and Ng, S. Positive-unlabeled learning for disease gene identification. *Bioinformatics*, 28(20):2640–2647, 2012.

Yu, H., Han, J., and Chang, K. C. PEBL: positive example based learning for web page classification using SVM. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 239–248, 2002.

Yu, H., Han, J., and Chang, K. C. PEBL: web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):70–81, 2004.

Zhang, B. and Zuo, W. Reliable negative extracting based on knn for learning from positive and unlabeled examples. *Journal of Computers*, 4(1):94–101, 2009.

Zhang, C., Ren, D., Liu, T., Yang, J., and Gong, C. Positive and unlabeled learning with label disambiguation. In *International Joint Conference on Artificial Intelligence*, pp. 4250–4256, 2019.

Zhang, H., Cissé, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Zhao, Y., Xu, Q., Jiang, Y., Wen, P., and Huang, Q. Dist-pu: Positive-unlabeled learning from a label distribution perspective. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14441–14450, 2022.

Zhou, Z. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.

Zhu, Z., Wang, L., Zhao, P., Du, C., Zhang, W., Dong, H., Qiao, B., Lin, Q., Rajmohan, S., and Zhang, D. Robust positive-unlabeled learning via noise negative sample self-correction. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3663–3673, 2023.