Dist-PU: Positive-Unlabeled Learning from a Label Distribution Perspective

Yunrui Zhao[1] Qianqian Xu[2,*] Yangbangyan Jiang[3,4]
Peisong Wen[1,2] Qingming Huang[1,2,5,*]
[1] School of Computer Science and Technology, University of Chinese Academy of Sciences
[2] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS
[3] State Key Laboratory of Information Security, Institute of Information Engineering, CAS
[4] School of Cyber Security, University of Chinese Academy of Sciences
[5] Key Laboratory of Big Data Mining and Knowledge Management, University of Chinese Academy of Sciences
zhaoyunrui20@mails.ucas.ac.cn {xuqianqian,wenpeisong20z}@ict.ac.cn
jiangyangbangyan@iie.ac.cn qmhuang@ucas.ac.cn

Abstract

Positive-Unlabeled (PU) learning tries to learn binary classifiers from a few labeled positive examples with many unlabeled ones. Compared with ordinary semi-supervised learning, this task is much more challenging due to the absence of any known negative labels. While existing cost-sensitive-based methods have achieved state-of-the-art performances, they explicitly minimize the risk of classifying unlabeled data as negative samples, which might result in a negative-prediction preference of the classifier. To alleviate this issue, we resort to a label distribution perspective for PU learning in this paper. Noticing that the label distribution of unlabeled data is fixed when the class prior is known, it can be naturally used as learning supervision for the model. Motivated by this, we propose to pursue the label distribution consistency between predicted and ground-truth label distributions, which is formulated by aligning their expectations. Moreover, we further adopt the entropy minimization and Mixup regularization to avoid the trivial solution of the label distribution consistency on unlabeled data and mitigate the consequent confirmation bias. Experiments on three benchmark datasets validate the effectiveness of the proposed method.

1. Introduction

With the advent of big data, deep neural networks have attracted extensive attention, since their performance has reached or even surpassed the human level in various tasks [21, 35, 33]. Specifically, such great success usually relies on supervision by a large amount of labeled data. However, it is hard to obtain intact label information in many real-world applications, even those with only binary options. For example, observed interactions between users and items in recommendation systems are labeled positives. Since many unconcerned factors like lack of exposure or other coincidences could account for missing interactions, we cannot view all the unobserved interactions as negatives. Similar scenarios include Alzheimer's disease recognition [11], malicious URL detection [45] and particle picking in cryo-electron micrographs [4], where we only have access to a few labeled positives with plenty of unlabeled data. Such a great demand motivates us to learn from positive and unlabeled data, also known as PU learning.

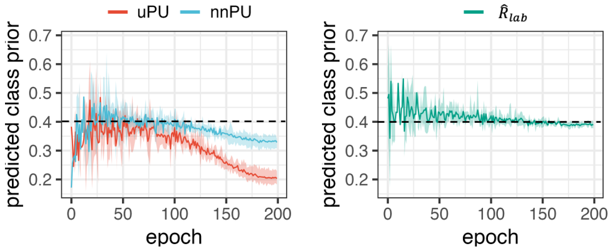


Figure 1. Predicted label distributions of uPU [12], nnPU [27], and Ours on the training data of CIFAR-10 with 5 repeats and Vehicles as the positive class. Compared with uPU and nnPU, the proposed label distribution alignment scheme ($\hat{R}_{lab}$) rectifies the negative-prediction preference, and achieves a predicted class prior consistent with the ground-truth (black dashed lines).

Researchers have developed numerous PU algorithms over the past decades. One prevalent research line would be cost-sensitive PU learning. These methods [12, 14, 27, 38] often assume the availability of class prior and minimize the risk of classifying unlabeled data as negative instances. By reweighting the importance of the positive and the negative risks, they could get an either unbiased or consistent risk estimator for PU learning.

Despite the great success the cost-sensitive methods have achieved, explicitly optimizing the risk that classifies unlabeled data to the negative class would essentially lead a flexible model (*e.g.*, deep neural networks) to overfit, and further result in a negative-prediction preference of the classifier, as illustrated in the left half of Fig. 1. As the training epoch increases, the predicted class prior probability $\Pr(\hat{y} = 1)$, *i.e.*, the expectation of predicted label distribution, tends to decrease from the ground-truth class prior. On the other hand, we notice that given the class prior, the underlying label distribution of data is immediately determined. Such a label distribution could be a natural supervision for the model. More specifically, the distribution of the model's predicted labels is supposed to be consistent with that of ground-truth ones. Following this intuition, we propose the label distribution alignment for PU learning, which aligns the expectations of the predicted and the ground-truth labels to ensure the label distribution consistency. In particular, the expectation of predicted labels is estimated by sigmoid outputs from a deep network, enabling the end-to-end learning of the proposed framework. Compared with the cost-sensitive methods, the proposed label distribution alignment scheme could rectify the negative-prediction preference (see Fig. 1).

Nevertheless, merely pursuing the label distribution consistency might suffer from a trivial solution that all the predicted scores of unlabeled data are equal to the class prior. To avoid this issue, we employ the entropy minimization technique, which encourages the model to produce scores much closer to zero or one. Meanwhile, Mixup [44] is further adopted to alleviate the confirmation bias caused by the model's overfitting on its early predictions. To summarize, the contributions of this paper are three-fold:

- We propose a PU learning framework based on label distribution alignment called **Dist-PU**. Different with existing methods, **Dist-PU** aligns the expectation of the model's predicted labels with that of ground-truth ones, and thus mitigates the negative-prediction preference. We also present a generalization bound for label distribution alignment as the theoretical guarantee.
- We further incorporate entropy minimization and Mixup to avoid the trivial solution of label distribution alignment and the consequent confirmation bias.
- Extensive experiments are conducted on Fashion-MNIST, CIFAR-10 and Alzheimer datasets, where **Dist-PU** outperforms existing state-of-the-art models in most cases.

2. Related work

PU learning models mainly fall into two categories. The current mainstream of PU methods adopts the framework of cost-sensitive learning. During the optimization step, samples relate to different importance weights. The unbiased risk estimator of PU learning, known as uPU, was proposed by Plessis, Niu, and Sugiyama [12]. Later, the authors of uPU found that a convex surrogate loss could reduce the computational cost [14]. Since then, works have surged to enhance this technique. Due to the deep models' strong fitting ability, the empirical risk of training data could go negative. Therefore, the non-negative risk estimator, known as nnPU, was proposed by Kiryo [27]. Besides, Self-PU [11] introduces self-supervision to nnPU via auxiliary tasks including model calibration and distillation with a self-paced curriculum; ImbPU [38] extends nnPU to imbalanced data by magnifying the weights of minority class. However, all the above methods assume identical distributions between labeled positives and ground-truth ones. PUSB [25] relaxes such an assumption by maintaining the order-preserving property. Furthermore, aPU [19] deals with an arbitrary positive shift between source and target distributions.

Another branch of PU learning employs two heuristic steps. Such methods [30, 42] first identify reliable negative or positive examples from the unlabeled data, thus yielding (semi-) supervised learning in the second step. The two-step methods differ in ways of assigning labels to unlabeled data. Graph-based methods [46, 9, 43] measure distances between samples through graphs to affirm the labels of the unlabeled data. RP [34] learns from confident examples whose predicted scores are near zero or one. PUbN [23] pretrains a model with nnPU to recognize some negatives. It then combines the positive risk, the unlabeled risk, and the negative risk to learn the final classifier. GenPU [22] is established from a generative learning perspective. It leverages the GAN framework and uses the generated data to train the final classifier. KLDCE[16] firstly translates PU learning into a label noise problem and weakens its side effect secondly via centroid estimation of the corrupted negative set. PULNS [32] incorporates reinforcement learning to obtain an effective negative sample selector.

Most PU learning methods are established on a known class prior. To this end, there emerge some class prior estimation algorithms designed for PU data. PE [13] uses partial matching and minimizes the Pearson divergence between the unlabeled and the labeled distributions. Pen-L1 [15] corrects the overestimate of PE. KM1 and KM2 [36] model the positive distribution with the distance between kernel embeddings. TIcE [2] approximates the class prior through a decision tree induction. CAPU [7] estimates the class prior and learns a classifier jointly. Other works pursue for PU learning without class prior. VPU [10] proposes a variational principle for PU learning with Mixup regularization. PAN [24] revises the architecture of GAN and proposes a new objective based on KL-distance. Instance-dependent PU learning [17] treats the class label as a hidden variable, aquiring the classifier under the EM framework.

Readers are referred to a recent survey of PU learning [3] for a more comprehensive study.

3. Methodology

3.1. Problem setting

In binary classification, the input space is $\mathcal{X} \subseteq \mathbb{R}^d$ with d dimensions, and the label space is $\mathcal{Y} = \{0, 1\}$ with 0 for negative and 1 for positive. Let $p(\boldsymbol{x}, y)$ be the underlying probability density of $(\mathcal{X}, \mathcal{Y})$, and $p(\boldsymbol{x})$ be the marginal distribution of the input. Then the sets of positive and negative samples could be denoted as:

$$\begin{aligned} \boldsymbol{X_P} &= \{\boldsymbol{x}\}^{n_P} \sim p_P(\boldsymbol{x}), \\ \boldsymbol{X_N} &= \{\boldsymbol{x}\}^{n_N} \sim p_N(\boldsymbol{x}), \end{aligned} \tag{1}$$

where $p_P(\boldsymbol{x})$ and $p_N(\boldsymbol{x})$ denote the class-conditional distribution of positive and negative samples respectively. Based on this, the whole sample set $\boldsymbol{X} = \boldsymbol{X_P} \cup \boldsymbol{X_N}$ is formulated as:

$$\boldsymbol{X} = \{\boldsymbol{x}\}^n \sim p(\boldsymbol{x}), \tag{2}$$

$$p(\boldsymbol{x}) = \pi_P \cdot p_P(\boldsymbol{x}) + \pi_N \cdot p_N(\boldsymbol{x}), \tag{3}$$

where $n = n_P + n_N$, $\pi_P = \Pr(y = 1)$ is the class prior probability and $\pi_N = 1 - \pi_P$.

We aim to learn a function $f \in \mathcal{F} : \mathbb{R}^d \to \mathbb{R}$ to minimize the average prediction error, also known as the expected risk:

$$R = \mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}\left[\mathbb{1}(\hat{y}, y)\right], \tag{4}$$

with $\hat{y} = f(\boldsymbol{x}) \in \{0, 1\}$ denoting the predicted label of $\boldsymbol{x}$ by f; $\mathbb{1}(\hat{y}, y)$ refers to the zero-one error, which equals 0 when $\hat{y} = y$, and otherwise 1. Unfortunately, $\mathbb{1}(\hat{y}, y)$ is discontinuous, making the model hard to optimize. Therefore, a surrogate loss $l(\cdot, \cdot)$ defined on $f(\boldsymbol{x})$ and y is usually used for risk minimization.

PU learning is a special case of binary classification, in the way that only a small portion of positive data are labeled. Formally, the training set $\boldsymbol{X_{PU}} = \boldsymbol{X_L} \cup \boldsymbol{X_U}$ where $\boldsymbol{X_L}$ or $\boldsymbol{X_U}$ represents the labeled positive or the unlabeled subset respectively. Under the Selected Completely at Random (SCAR) assumption and the case-control scenario [3], positives are labeled uniformly at random and independently of their features, while the unlabeled data are i.i.d drawn from the real marginal distribution:

$$\boldsymbol{X_L} = \{\boldsymbol{x}\}^{n_L} \sim p_P(\boldsymbol{x}), \tag{5}$$

$$\boldsymbol{X_U} = \{\boldsymbol{x}\}^{n_U} \sim p_U(\boldsymbol{x}) = p(\boldsymbol{x}). \tag{6}$$

3.2. Label distribution alignment

The key to PU learning lies in the ways of incorporating unlabeled data $\boldsymbol{X_U}$ into the training process. Without knowing any label in $\boldsymbol{X_U}$, we cannot directly calculate a loss like $\ell(f(\boldsymbol{x}), y)$. As a solution, a large proportion of existing methods rely on the risk of classifying unlabeled data as positive or negative samples. Yet an essential fact might be ignored that the expectation of all the ground-truth labels over the entire data distribution is exactly the class prior, *i.e.*, $\mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}[y] = \pi_P$. Motivated by this, we propose to use the class prior π_P to guide the learning of the model. More specifically, we pursue the consistency between the predicted and ground-truth label distributions of labeled and unlabeled data, offering relatively reliable supervision. As shown in Fig. 2, for labeled positives $\boldsymbol{X_L}$, the expectation of predicted labels should be equal to 1. Meanwhile, in $\boldsymbol{X_U}$, positives are expected to take over π_P proportion, and π_N for negatives. Thus, the expectation over $\boldsymbol{X_U}$ should be matched to π_P. By aligning the expectations of predicted and ground-truth label distributions, the label distribution consistency could be achieved. Such a mechanism is called as *label distribution alignment* in this paper.

In order to infer the formulation of the label distribution alignment, we first reformulate the expected risk R of Eq. (4) as a combination of the expected risks on positive and negative classes:

$$R = \pi_P \mathbb{E}_{\boldsymbol{x}\sim p_P(\boldsymbol{x})}\left[\mathbb{1}(\hat{y}, 1)\right] + \pi_N \mathbb{E}_{\boldsymbol{x}\sim p_N(\boldsymbol{x})}\left[\mathbb{1}(\hat{y}, 0)\right]. \tag{7}$$

Since $\hat{y}, y \in \{0, 1\}$, we have $\mathbb{1}(\hat{y}, y) = |\hat{y} - y|$. Moreover, for all the positive samples, we can rewrite its loss by $\mathbb{1}(\hat{y}, y) = 1 - \hat{y}$. Likewise, the zero-one loss of negative samples could be written as $\mathbb{1}(\hat{y}, y) = \hat{y}$. Then R in Eq.(7) could be reformulated as:

$$\begin{aligned} R =& \pi_P \mathbb{E}_{\boldsymbol{x}\sim p_P(\boldsymbol{x})}[1 - \hat{y}] + \pi_N \mathbb{E}_{\boldsymbol{x}\sim p_N(\boldsymbol{x})}[\hat{y}], \\ =& \pi_P \underbrace{\left|\mathbb{E}_{\boldsymbol{x}\sim p_P(\boldsymbol{x})}[\hat{y}] - 1\right|}_{R_P} + \pi_N \underbrace{\left|\mathbb{E}_{\boldsymbol{x}\sim p_N(\boldsymbol{x})}[\hat{y}]\right|}_{R_N}, \end{aligned} \tag{8}$$

where R_P and R_N are the expected risk of positive and negative samples, respectively. Such a reformulation naturally induces the principle of the label distribution consistency we pursue, *i.e.*, the expectation of $\hat{y}$ should be equal to that of y. In other words, given a dataset, the class probabilities of the predicted labels should be consistent with the ground-truth class priors.

With this principle in mind, we need to find a way to estimate R_P and R_N in PU learning. In this setting, R_P could be evaluated directly from the labeled positive samples, while the situation is not the same for R_N due to the absence of any known negative labels. Instead, we may resort to the label distribution consistency over unlabeled data. According to Eq. (6), the distribution of the unlabeled data is the same with the real distribution. Therefore, we first concentrate on the difference between the expectation of the predicted labels and that of the underlying ground-truth labels concerning the real distribution:

$$\begin{aligned} & \mathbb{E}_{\boldsymbol{x}\sim p_U(\boldsymbol{x})}[\hat{y}] - \mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}[y] \\ =& \pi_P \mathbb{E}_{\boldsymbol{x}\sim p_P(\boldsymbol{x})}[\hat{y}] - \pi_P + \pi_N \mathbb{E}_{\boldsymbol{x}\sim p_N(\boldsymbol{x})}[\hat{y}] \\ =& -\pi_P R_P + \pi_N R_N. \end{aligned} \tag{9}$$

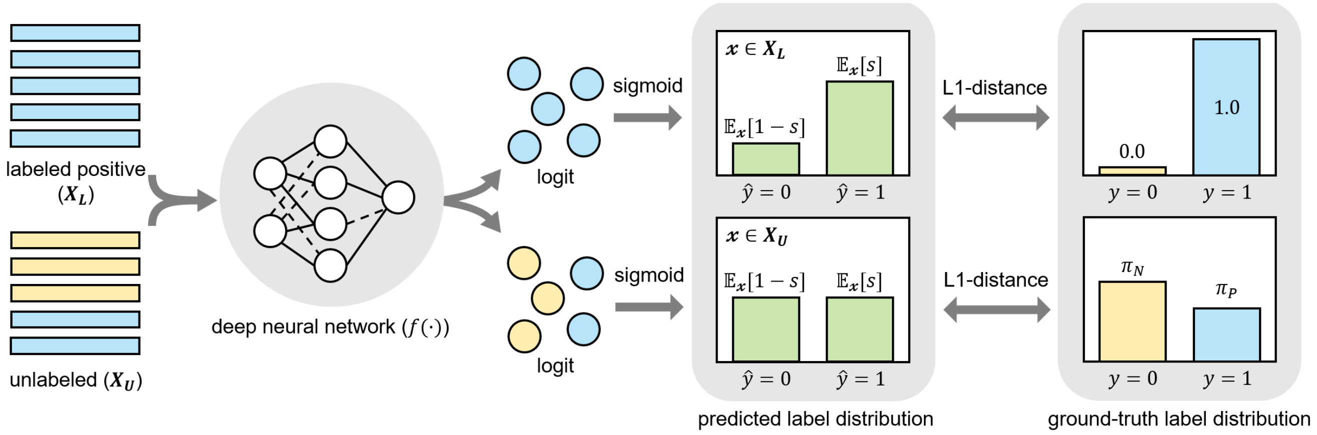


Figure 2. Overview of the proposed label distribution alignment framework. We aim at matching the predicted label distributions over labeled positive and unlabeled data with their ground-truth distributions by aligning the corresponding expectations. Specifically, a labeled positive subset ($\boldsymbol{X_L}$) and unlabeled data ($\boldsymbol{X_U}$) are fed into a deep neural network (f) followed by a sigmoid function. Using the output score s, we could approximately estimate the expectation of $\hat{y}$ by $\mathbb{E}_{\boldsymbol{x}}[s]$. Note that the expectation of the ground-truth label y's distribution of $\boldsymbol{X_L}$ is 1, while that of $\boldsymbol{X_U}$ is equal to the class prior π_P. Then, we train f by minimizing the L1-distances between the expectations of the predicted and the ground-truth labels from $\boldsymbol{X_L}$ and $\boldsymbol{X_U}$, respectively.

Then we could easily represent R_N by R_P and the expectations of label distributions over unlabeled data:

$$\begin{aligned}\pi_N R_N =&\pi_P R_P + \mathbb{E}_{\boldsymbol{x}\sim p_U(\boldsymbol{x})}\left[\hat{y}\right] - \mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}\left[y\right]\\ =&\pi_P R_P + \mathbb{E}_{\boldsymbol{x}\sim p_U(\boldsymbol{x})}\left[\hat{y}\right] - \pi_P. \qquad (10)\end{aligned}$$

By substituting Eq. (8) into Eq. (10), we could derive an equivalent form of R in a label distribution alignment manner, which does not involve any explicit calculation using negative labels:

$$R = 2\pi_P R_P + \mathbb{E}_{\boldsymbol{x}\sim p_U(\boldsymbol{x})}\left[\hat{y}\right] - \pi_P. \qquad (11)$$

However, due to insufficient label information and weak supervision for unlabeled data, the expectation of the predicted labels might be unstable during the training process. It could deviate far from the ground-truth prior, even making the difference of $\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\hat{y}\right]$ and π_P negative. In this case, continuing minimizing this risk might worsen the situation. On the other hand, taking the label distribution consistency into account, we wish that $\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\hat{y}\right]$ is exactly π_P. Therefore, we propose a variant risk by putting an absolute function into R:

$$R_{lab} = 2\pi_P R_P + \underbrace{\left|\mathbb{E}_{\boldsymbol{x}\sim p_U(\boldsymbol{x})}\left[\hat{y}\right] - \pi_P\right|}_{R_U}. \qquad (12)$$

Though by this equation we decompose the risk over the entire data distribution into terms which can be estimated using labeled positive and unlabeled samples, there still exist some practical issues when applying the Empirical Risk Minimization (ERM) principle [39] for its optimization, *i.e.*, the differentiability of $\hat{y}$. It is common and straightforward to determine the predicted label via the sign of the model's output score $f(\boldsymbol{x})$. Namely, $\hat{y} = \frac{1}{2}(\text{sgn}(f(\boldsymbol{x}))+1)$. This process suffers from the non-differentiability of the sign function. As a result, the induced empirical objective could not be updated in an end-to-end manner. To mitigate this issue, we turn to directly consider $\mathbb{E}_{\boldsymbol{x}}\left[\hat{y}\right]$ and compute it by the conditional probability of $\hat{y}$ given $\boldsymbol{x}$:

$$\mathbb{E}_{\boldsymbol{x}}\left[\hat{y}\right] = \mathbb{E}_{\boldsymbol{x}}\left[\Pr\left(\hat{y}=1|\boldsymbol{x}\right)\cdot 1 + \Pr\left(\hat{y}=0|\boldsymbol{x}\right)\cdot 0\right]. \qquad (13)$$

The problem then shifts to how to aquire the conditional probability. Through a sigmoid function, we can approximately calculate $\Pr\left(\hat{y}=1|\boldsymbol{x}\right)$ by s using $f(\boldsymbol{x})$ as follows:

$$s = \frac{1}{1+\exp\left[-f(\boldsymbol{x})\right]}. \qquad (14)$$

In this manner, we could alternatively employ the differentiable $\mathbb{E}_{\boldsymbol{x}}\left[s\right]$ instead of the original expectation of hard predicted labels, furtherly making the optimization compatible with a gradient descent style.

To sum up, we obtain the optimization objective of our label distribution alignment as follows:

$$\hat{R}_{lab} = 2\pi_P \underbrace{\left|\frac{1}{n_L}\sum_{\boldsymbol{x}\in\boldsymbol{X_L}} s - 1\right|}_{\hat{R}_L} + \underbrace{\left|\frac{1}{n_U}\sum_{\boldsymbol{x}\in\boldsymbol{X_U}} s - \pi_P\right|}_{\hat{R}_U}, \qquad (15)$$

where $\hat{R}_L$ and $\hat{R}_U$ denote the empirical risk estimator of R_P and R_U, respectively.

Though the label distribution alignment risk $\hat{R}_{lab}$ is obviously biased, the following proposition shows that the original risk R can be upper bounded by $\hat{R}_{lab}$ together with sample sizes and model complexity based terms.

Proposition 1. *For a class of b-uniformly bounded functions $\mathcal{F}$ with the VC dimension $\mathcal{V}$, with a probability at least $1-\delta$, it holds that:*

$$\begin{aligned} R \leq & 2\hat{R}_{lab} + 8\pi_P \cdot C\sqrt{\frac{\mathcal{V}}{n_L}} + 12\pi_P\sqrt{\frac{\ln 4/\delta}{2n_L}} \\ & + 4C\sqrt{\frac{\mathcal{V}}{n_U}} + 6\sqrt{\frac{\ln 4/\delta}{2n_U}}, \end{aligned} \tag{16}$$

where C is a universal constant.

Therefore, optimizing the upper bound naturally leads to a minimization of R. Since all the terms except $\hat{R}_{lab}$ only depend on the dataset sizes and the complexity of the model, here we could minimize $\hat{R}_{lab}$ to indirectly optimize R. Moreover, we also see that using more labeled positive or unlabeled samples, or a less-complex model (*i.e.*, smaller $\mathcal{V}$) would reduce the generalization gap. Proof is provided in supplementary materials due to the space limitation.

3.3. Entropy minimization

So far, Eq. (15) seems to serve our pursuit well. However, we would like to remind that Eq. (15) might induce a trivial solution. Consider the trivial solution that every s is caculated as π_P regradless of the inputs, $\hat{R}_U$ will reach 0 as well, which is of course against our hope. We hope that on the premise of label distribution consistency, s is as close to either 0 or 1 as possible. To this aim, we introduce the *entropy minimization* technique to concentrate the probability distribution on a single class, thus avoiding the trivial solution. Entropy minimization appears widely in semi-supervised learning [18, 41, 6, 5]. It originates from the well-known clustering hypothesis that the data near the hyperplane of the classification decision is sparse, and the data in the place of the same category cluster is dense. According to this hypothesis, we expect low entropy of the scores obtained from the unlabeled data:

$$L_{ent} = -\frac{1}{n_U}\sum_{\boldsymbol{x}\in\boldsymbol{X_U}}\left[(1-s)\log(1-s) + s\log s\right]. \tag{17}$$

3.4. Confirmation bias

Although the incorporation of entropy minimization could result in a sharper bimodal distribution for predicted scores of unlabeled data, it might also further increase over-confidence of the model. Such over-confidence is especially fatal during the early training stage. Incorrect predictions might get reinforced as the training process steps forward, while the label distribution consistency and entropy minimization make the situation even worse. The phenomenon of overfitting wrongly predicted data, known as *confirmation bias*, is ubiquitous in semi-supervised learning [1, 29]. In this paper, we resort to the popular Mixup regularization [44] to deal with the confirmation bias.

Mixup is motivated by the Vicinal Risk Minimization (VRM) principle [8]. It randomly combines training pairs $\boldsymbol{x_1}, \boldsymbol{x_2} \in \boldsymbol{X_{PU}}$ convexly with the mixing proportion λ chosen from a Beta distribution:

$$\lambda \sim \text{Beta}(\alpha, \alpha), \tag{18}$$

$$\lambda' = \max(\lambda, 1-\lambda), \tag{19}$$

$$\boldsymbol{x'} = \lambda'\boldsymbol{x_1} + (1-\lambda')\boldsymbol{x_2}. \tag{20}$$

Then the regularization term leads the mixed sample $\boldsymbol{x'}$ to have a consistent prediction with the linear interpolation of the associated targets:

$$\begin{aligned} L_{mix} = \frac{1}{n}\sum_{\boldsymbol{x_1},\boldsymbol{x_2}\in\boldsymbol{X_{PU}}} & [\lambda' l_{bce}(s', s_1) \\ & + (1-\lambda')l_{bce}(s', s_2)], \end{aligned} \tag{21}$$

where $l_{bce}(t', t) = -(1-t)\log(1-t') - t\log(t')$ is the binary cross entropy loss; s' is the predicted score of $\boldsymbol{x'}$, while $s_1(s_2)$ refers to the corresponding soft label (*i.e.*, predicted score) of $\boldsymbol{x_1}(\boldsymbol{x_2})$.

The reasons why Mixup can alleviate confirmation bias are three-folds. Firstly, using predicted scores as soft labels in Eq. (21) reduces the model's over-confidence, balancing the trade-off between entropy minimization and confirmation bias. Secondly, Mixup loss is more robust to prediction errors. For instance, considering two examples that are predicted as false negative and false positive, respectively, their Mixup loss will still be helpful especially when λ' in Eq. (19) is near 0.5. Thirdly, Mixup could be regarded as a kind of data augmentation [37], which extends the training dataset by additional virtual examples as shown in Eq. (20). With the training distribution enlarged, the model generalization gets improved.

Moreover, we incorporate entropy minimization for the mixed data as well:

$$L'_{ent} = -\frac{1}{n}\sum_{\boldsymbol{x_1},\boldsymbol{x_2}\in\boldsymbol{X_{PU}}} l_{bce}(s', s'). \tag{22}$$

Finally, we present the overall objective of the label distribution alignment with entropy minimization and Mixup, denoted as **Dist-PU**, as follows:

$$L_{dist} = \hat{R}_{lab} + \mu L_{ent} + \nu L_{mix} + \gamma L'_{ent}, \tag{23}$$

where μ ajusts the importance of L_{ent}; ν and γ control the strength of L_{mix} and L'_{ent}, respectively.

Table 1. Summary of used datasets and their corresponding models.

Dataset	Input Size	n_L	n_U	# Testing	π_P	Positive Class	Model
F-MNIST	28×28	500	60,000	10,000	0.4	Top (*i.e.*, 0, 2, 4 and 6)	6-layer MLP
CIFAR-10	$3 \times 32 \times 32$	1,000	50,000	10,000	0.4	Vehicles (*i.e.*, 0, 1, 8 and 9)	13-layer CNN
Alzheimer	$3 \times 224 \times 224$	769	5,121	1,279	0.5	Alzheimer's Disease	ResNet-50 [20]

Table 2. Comparative results on F-MNIST, CIFAR-10, and Alzheimer. Best and second best values are both highlighted. ✓(✗) denotes that our Dist-PU is significantly better (worse) than the corresponding methods revealed by the paired t-test with confidence level 95%.

Dataset	Method	ACC (%)	Prec. (%)	Rec. (%)	F1 (%)	AUC (%)	AP (%)
F-MNIST	naive	91.07 (0.93) ✓	90.16 (2.25) ✓	87.28 (2.08) ✓	88.66 (1.14) ✓	96.94 (0.67) ✓	94.27 (1.40) ✓
	uPU	94.02 (0.30) ✓	92.50 (1.26) ✓	92.59 (0.80) ✓	92.53 (0.31) ✓	97.34 (0.54) ✓	96.60 (0.52) ✓
	nnPU	94.44 (0.49) ✓	91.69 (1.13) ✓	94.69 (0.84)	93.16 (0.57) ✓	97.53 (0.48) ✓	96.39 (0.98) ✓
	RP	92.37 (1.08) ✓	88.58 (1.56) ✓	92.94 (2.38) ✓	90.69 (1.39) ✓	97.14 (0.58) ✓	94.39 (1.31) ✓
	PUSB	94.50 (0.36) ✓	93.12 (0.44) ✓	93.12 (0.44) ✓	93.12 (0.44) ✓	97.31 (0.50) ✓	96.28 (0.96) ✓
	PUbN	94.82 (0.16) ✓	92.92 (0.50) ✓	94.24 (0.93)	93.57 (0.24) ✓	94.72 (0.28) ✓	89.87 (0.18) ✓
	Self-PU	94.75 (0.25) ✓	91.73 (0.80) ✓	95.50 (0.61)	93.57 (0.28) ✓	97.62 (0.31) ✓	96.14 (0.70) ✓
	aPU	94.71 (0.34) ✓	92.71 (0.50) ✓	94.20 (1.06)	93.44 (0.45) ✓	97.67 (0.40) ✓	96.64 (0.48) ✓
	VPU	92.26 (1.11) ✓	89.04 (2.00) ✓	92.01 (2.00) ✓	90.48 (1.35) ✓	97.38 (0.44) ✓	95.57 (0.62) ✓
	ImbPU	94.54 (0.42) ✓	92.81 (1.53) ✓	93.66 (1.67)	93.21 (0.52) ✓	97.67 (0.81) ✓	96.73 (0.86) ✓
	Dist-PU	95.40 (0.34)	94.18 (0.90)	94.34 (1.00)	94.25 (0.43)	98.57 (0.24)	97.90 (0.30)
CIFAR-10	naive	84.92 (0.89) ✓	83.59 (0.89) ✓	77.57 (3.23) ✓	80.43 (1.56) ✓	92.29 (0.63) ✓	88.23 (0.79) ✓
	uPU	88.35 (0.45) ✓	87.18 (2.39) ✓	83.23 (2.68) ✓	85.10 (0.56) ✓	94.91 (0.62) ✓	92.62 (1.11) ✓
	nnPU	88.89 (0.45) ✓	86.18 (1.15) ✓	86.05 (1.42) ✓	86.10 (0.58) ✓	95.12 (0.52) ✓	92.42 (1.38) ✓
	RP	88.73 (0.15) ✓	86.01 (1.01) ✓	85.82 (1.51) ✓	85.90 (0.32) ✓	95.17 (0.23) ✓	92.92 (0.56) ✓
	PUSB	88.95 (0.41) ✓	86.19 (0.51) ✓	86.19 (0.51) ✓	86.19 (0.51) ✓	95.13 (0.52) ✓	92.44 (1.34) ✓
	PUbN	89.83 (0.30) ✓	87.85 (0.98) ✓	86.56 (1.87) ✓	87.18 (0.54) ✓	89.28 (0.54) ✓	81.41 (0.37) ✓
	Self-PU	89.28 (0.72) ✓	86.16 (0.78) ✓	87.21 (2.35) ✓	86.67 (1.06) ✓	95.47 (0.58) ✓	93.28 (1.01) ✓
	aPU	89.05 (0.52) ✓	86.29 (1.30) ✓	86.37 (0.79) ✓	86.32 (0.56) ✓	95.09 (0.42) ✓	92.41 (1.23) ✓
	VPU	87.99 (0.48) ✓	86.72 (1.41) ✓	82.71 (2.84) ✓	84.63 (0.91) ✓	94.51 (0.41) ✓	92.00 (0.73) ✓
	ImbPU	89.41 (0.46) ✓	86.69 (0.87) ✓	86.87 (0.82) ✓	86.77 (0.56) ✓	95.52 (0.27) ✓	93.45 (0.45) ✓
	Dist-PU	91.88 (0.52)	89.87 (1.09)	89.84 (0.81)	89.85 (0.62)	96.92 (0.45)	95.49 (0.72)
Alzheimer	naive	61.45 (3.81) ✓	62.51 (5.87) ✓	61.44 (12.5) ✓	61.05 (4.65) ✓	66.26 (6.39) ✓	63.28 (5.98) ✓
	uPU	68.48 (2.15) ✓	69.65 (3.50)	66.13 (6.13) ✓	67.62 (2.78) ✓	73.75 (2.94) ✓	69.53 (3.23) ✓
	nnPU	68.33 (2.13) ✓	68.01 (2.33)	69.48 (7.15) ✓	68.55 (3.16) ✓	72.90 (2.80) ✓	69.45 (2.87) ✓
	RP	61.61 (3.20) ✓	61.89 (4.54) ✓	64.60 (15.89)	62.10 (5.61) ✓	66.13 (3.28) ✓	63.82 (2.29) ✓
	PUSB	69.21 (2.39)	69.16 (2.39)	69.26 (2.39) ✓	69.21 (2.39) ✓	74.43 (2.41) ✓	70.00 (1.56) ✓
	PUbN	69.98 (1.34) ✓	69.42 (2.49)	72.02 (8.42) ✓	70.38 (3.19)	69.98 (1.34) ✓	63.84 (0.95) ✓
	Self-PU	70.88 (0.72) ✓	69.32 (2.54)	75.43 (5.07)	72.09 (1.09) ✓	75.89 (1.79) ✓	71.68 (3.84)
	aPU	68.52 (1.75) ✓	66.21 (0.91) ✓	75.71 (8.20)	70.46 (3.35)	73.80 (2.59) ✓	70.71 (3.70) ✓
	VPU	67.44 (0.65) ✓	64.74 (1.12) ✓	76.68 (3.60)	70.16 (1.08) ✓	73.12 (0.85) ✓	71.11 (0.75)
	ImbPU	68.18 (0.83) ✓	67.54 (2.52) ✓	70.64 (6.54) ✓	68.83 (1.94) ✓	73.81 (0.71) ✓	70.46 (1.07) ✓
	Dist-PU	71.57 (0.62)	68.48 (1.16)	80.09 (5.10)	73.74 (1.64)	77.13 (0.69)	73.33 (1.47)

4. Experiment

4.1. Experimental settings

Datasets. We conduct experiments on three benchmarks: F-MNIST [40] for fashion product classification, CIFAR-10 [28] for vehicle class identification, together with the Alzheimer dataset[1] for the recognition of the Alzheimer's Disease. More details are concluded in Tab.1.

Evaluation metrics. For each model, we report six metrics on the test set for a more comprehensive comparison, including accuracy (ACC), Precision (Prec.), Recall (Rec.), F1, Area Under ROC Curve (AUC) and Average Precision (AP). Experiments are repeated with five random seeds, then the mean and the standard deviation of each metric are recorded.

Implementation details. All the experiments are run on Geoforce RTX 3090 implemented by PyTorch. Backbones of each dataset are summarized in Tab.1. For CIFAR-10, we only normalize the input images with mean = (0.485, 0.456, 0.406) and std = (0.229, 0.224, 0.225) without any extra data preprocessing techniques. Besides, we clamp the logits between -10 and 10 to avoid the potantial NaN error

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

Table 3. Ablation results on CIFAR-10 with ✓indicating the enabling of the corresponding loss term.

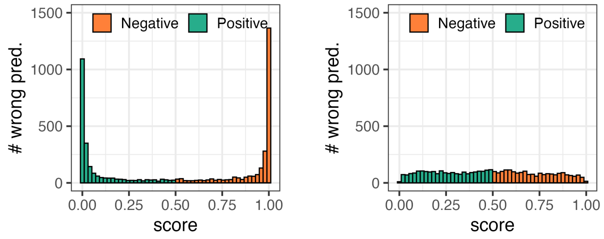
Variant	$\hat{R}_{lab}$	L_{ent} / L'_{ent}	L_{mix}	ACC (%)	Prec. (%)	Rec. (%)	F1 (%)	AUC (%)	AP (%)
I				84.92 (0.89)	83.59 (0.89)	77.57 (3.23)	80.43 (1.56)	92.29 (0.63)	88.23 (0.79)
II	✓			89.30 (0.48)	86.96 (0.86)	86.19 (1.12)	86.57 (0.52)	95.40 (0.52)	93.22 (0.97)
III		✓		87.50 (0.24)	83.55 (0.17)	85.62 (0.78)	84.57 (0.36)	93.61 (0.33)	89.36 (0.98)
IV			✓	86.97 (0.23)	84.50 (4.35)	83.22 (6.47)	83.57 (1.11)	94.37 (0.22)	91.21 (0.73)
V	✓	✓		89.41 (0.48)	86.40 (0.98)	87.30 (1.75)	86.83 (0.69)	95.61 (0.44)	93.51 (0.46)
VI	✓		✓	91.62 (0.47)	90.96 (1.07)	87.78 (0.32)	89.34 (0.54)	96.80 (0.44)	95.54 (0.35)
VII		✓	✓	87.67 (0.52)	85.01 (1.78)	84.07 (1.90)	84.51 (0.60)	94.32 (0.35)	90.65 (0.80)
VIII	✓	✓	✓	91.88 (0.52)	89.87 (1.09)	89.84 (0.81)	89.85 (0.62)	96.92 (0.45)	95.49 (0.72)

from Eq. (17,22). The training batch size is set as 256 for F-MNIST and CIFAR-10, while 128 for Alzheimer. We use Adam [26] as the optimizer with a cosine annealing scheduler [31], where the initial learning rate is set as 5×10^{-4}; weight decay is set as 5×10^{-3}. Dist-PU first experiences a warm-up stage of several epochs without Mixup, then trains with Mixup for another 60 epochs, where μ is also with a cosine annealing scheduling to mitigate overfitting. Moreover, the values of hyperparameters μ, ν, γ and α are searched within the range of $[0, 0.1]$, $[0, 10]$, $[0, 0.3]$ and $[0.1, 10]$, respectively.

4.2. Comparision with state-of-the-art methods

Competitors. We compare our Dist-PU with a naive baseline which constructs the negative class with randomly sampled unlabeled data, together with 9 competitive PU algorithms including uPU [12], nnPU [27], RP [34], PUSB [25], PUbN [23], Self-PU [11], aPU [19], VPU [10], and ImbPU [38]. Due to the space limitation, the detailed descriptions are provided in supplementary materials.

Results. The results on all the datasets are recorded in Tab.2. It shows that our proposed Dist-PU outperforms the competitors by a significant margin for all the datasets in terms of most metrics, surpassing the second-best roughly by 1% to 3% on average. This validates the effectiveness of our proposed method. Moreover, our Dist-PU also achieves a relative balance between precision and recall metrics. Besides, some observations could be made: (1) We see that the performance of classic cost-sensitive PU approaches including uPU, nnPU, aPU and ImbPU fluctuate over different datasets. This might be because their objectives consist of the loss classifying unlabeled data to the negative class, thus they may suffer from overfitting and tend to make negative predictions. On the contrary, our Dist-PU alleviates this problem via the proposed label distribution alignment. (2) Self-PU and PUbN are the most competitive baselines. Their success can be partly attributed to some extra designs. For example, Self-PU adopts mentor net and self-paced learning, while PUbN relies on the pre-trained model by nnPU. In the ablation study, we will show that our method still performs well without Mixup regularization. (3) Some competitors such as VPU, performs relatively less promising. Notably, considering that VPU does not use the class prior information, it might not be able to outperform other baselines given the accurate class prior. Nevertheless, sensitivity analysis in supplementary materials shows that our Dist-PU is relatively robust to misspecified prior.



(a) w/o Mixup

(b) w/ Mixup

Figure 3. Histogram of wrongly-predicted scores (s) from the training set on CIFAR-10.

4.3. Ablation studies

In order to analyze the impact of each module in Dist-PU, *i.e.*, label distribution alignment, entropy minimization, and Mixup regurlarization, we conduct ablation studies on CIFAR-10. The results are shown in Tab.3. From the table, we could get the following observations: (1) Our label distribution alignment plays an essential role in Dist-PU. By comparing the results of the variants I w/ II, IV w/ VI, and VII w/ VIII, we can find that our label distribution alignment leads to a significant improvement from those without label distribution alignment. (2) Entropy minimization exhibits a minor enhancement, which can be observed from comparisons of variants II w/ V and VI w/ VIII. (3) Mixup contributes to the performance especially when label distribution alignment is incorporated. As shown by variants III and VII, Mixup brings few gains for entropy minimization without label distribution alignment. However, with label distribution alignment, the increase from Mixup is much more noticeable based on comparisons of variants II with VI and V with VIII. This also reflects another advantage of label distribution alignment in a way that it promotes the effect of Mixup.

For better understanding the impact of Mixup on allevi-

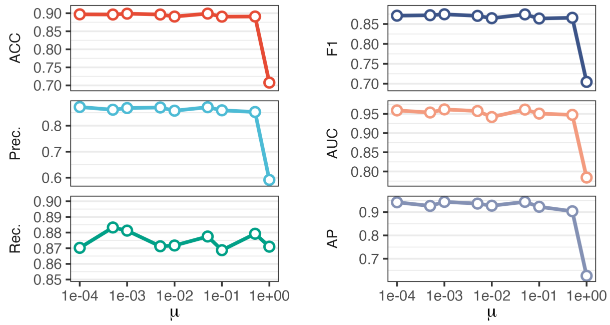


Figure 4. Influences of different μ on CIFAR-10.

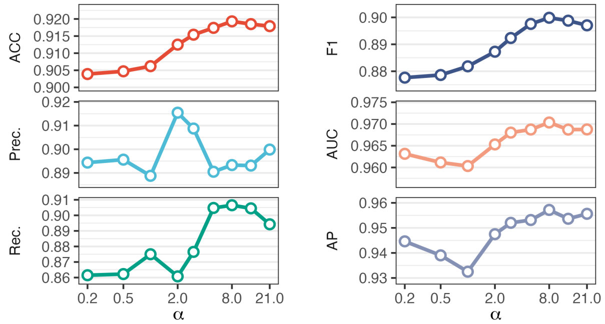


Figure 5. Influences of different α on CIFAR-10.

ating confirmation bias, we plot the scores of wrong predictions on the training data of CIFAR-10 in Fig. 3. As the result shows, with Mixup, the total number of wrong predictions becomes less. Besides, the score distribution is more smooth with Mixup, which means the confidence of wrong predictions is lower in general. This unveils some intrinsic mechanisms of Mixup in reducing confirmation bias.

4.4. Effectiveness of hyper-parameters

Effectiveness of μ for entropy minimization on $\mathcal{X}_{\mathcal{U}}$. Parameter μ controls the concentration degree of the predicted score distribution of the unlabeled data. To get rid of the interference of other loss terms, we conduct experiments with different μ without Mixup. In Fig. 4, we can see that when μ locates between 0 and 0.1, all the metrics except Rec. are relatively stable. However, a μ larger than 0.5 heavily degenerates the model performance.

Effectiveness of α for Beta distribution in Mixup. Parameter α controls the shape of Beta distribution in Mixup. The higher α is, the narrower its curve will be and λ will be more likely near 0.5. Fig. 5 shows that rationally larger α helps improve the performance. Understandably, a larger α means more chance to get a λ near 0.5 and a better-mixed target for the mixed data, especially when its source labels are different, thus better reducing confirmation bias.

Effectiveness of ν and γ for Mixup. Parameter ν controls the power of Mixup on alleviating confirmation bias.

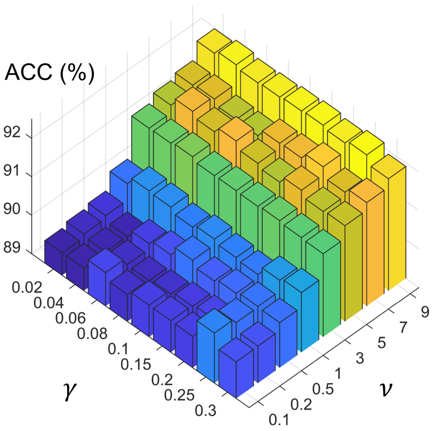


Figure 6. Influences of different ν and γ on CIFAR-10.

In Fig. 6, we see that the larger ν is, the better confirmation bias is weakened and the higher ACC our algorithm could reach. It also shows that a moderately larger γ, which leads to lower entropy of the mixed predictions, has positive effects on the performance.

5. Conclusion

This paper proposes a novel PU learning method from a label distribution alignment perspective called **Dist-PU**. Specifically, Dist-PU pursues the label distribution consistency between the predictions and the ground-truth labels over the labeled positive and unlabeled data. It then leverages entropy minimization to make the positive-negative distributions more separable. With Mixup to mitigate the confirmation bias, Dist-PU consistently outperforms the state-of-the-art methods over most metrics on real-world datasets including F-MNIST, CIFAR-10 and Alzheimer. Ablation studies and sensitivity analysis further demonstrate the effectiveness of each module in Dist-PU. We hope that the proposed label distribution alignment scheme could provide some enlightenment on other weakly supervised scenarios as well, especially for those with unlabeled or inaccurately labeled data but of known label distributions.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0102000, in part by National Natural Science Foundation of China: U21B2038, 61931008, 61836002, 6212200758 and 61976202, in part by the Fundamental Research Funds for the Central Universities, in part by Youth Innovation Promotion Association CAS, in part by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDB28000000.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 1–8, 2020. 5

[2] Jessa Bekker and Jesse Davis. Estimating the class prior in positive and unlabeled data through decision tree induction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 2712–2719, 2018. 2

[3] Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: a survey. *Machine Learning*, 109(4):719–760, 2020. 2, 3

[4] Tristan Bepler, Andrew Morin, Micah Rapp, Julia Brasch, Lawrence Shapiro, Alex J Noble, and Bonnie Berger. Positive-unlabeled convolutional neural networks for particle picking in cryo-electron micrographs. *Nature Methods*, 16(11):1153–1160, 2019. 1

[5] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *Proceedings of the 8th International Conference on Learning Representations*, 2020. 5

[6] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019. 5

[7] Shizhen Chang, Bo Du, and Liangpei Zhang. Positive unlabeled learning with class-prior approximation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 2014–2021, 2020. 2

[8] Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *Advances in Neural Information Processing Systems*, pages 416–422, 2000. 5

[9] Sneha Chaudhari and Shirish K. Shevade. Learning from positive and unlabelled examples using maximum margin clustering. In *Proceedings of the 19th International Conference on Neural Information Processing*, volume 7665, pages 465–473, 2012. 2

[10] Hui Chen, Fangqing Liu, Yin Wang, Liyue Zhao, and Hao Wu. A variational approach for learning from positive and unlabeled data. In *Advances in Neural Information Processing Systems*, 2020. 2, 7

[11] Xuxi Chen, Wuyang Chen, Tianlong Chen, Ye Yuan, Chen Gong, Kewei Chen, and Zhangyang Wang. Self-pu: Self boosted and calibrated positive-unlabeled training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 1510–1519, 2020. 1, 2, 7

[12] Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *Advances in Neural Information Processing Systems*, pages 703–711, 2014. 1, 2, 7

[13] Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Class-prior estimation for learning from positive and unlabeled data. In *Proceedings of The 7th Asian Conference on Machine Learning*, volume 45, pages 221–236, 2015. 2

[14] Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1386–1394, 2015. 1, 2

[15] Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Class-prior estimation for learning from positive and unlabeled data. *Machine Learning*, 106(4):463–492, 2017. 2

[16] Chen Gong, Hong Shi, Tongliang Liu, Chuang Zhang, Jian Yang, and Dacheng Tao. Loss decomposition and centroid estimation for positive and unlabeled learning. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):918–932, 2019. 2

[17] Chen Gong, Qizhou Wang, Tongliang Liu, Bo Han, Jane J You, Jian Yang, and Dacheng Tao. Instance-dependent positive and unlabeled learning with labeling bias estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[18] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, pages 529–536, 2004. 5

[19] Zayd Hammoudeh and Daniel Lowd. Learning from positive and unlabeled data with arbitrary positive shift. In *Advances in Neural Information Processing Systems*, 2020. 2, 7

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[21] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory F. Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *CoRR*, abs/1712.00409, 2017. 1

[22] Ming Hou, Brahim Chaib-Draa, Chao Li, and Qibin Zhao. Generative adversarial positive-unlabeled learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2255–2261, 2018. 2

[23] Yu-Guan Hsieh, Gang Niu, and Masashi Sugiyama. Classification from positive, unlabeled and biased negative data. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 2820–2829, 2019. 2, 7

[24] Wenpeng Hu, Ran Le, Bing Liu, Feng Ji, Jinwen Ma, Dongyan Zhao, and Rui Yan. Predictive adversarial learning from positive and unlabeled data. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 7806–7814, 2021. 2

[25] Masahiro Kato, Takeshi Teshima, and Junya Honda. Learning from positive and unlabeled data with a selection bias. In *Preceedings of the 7th International Conference on Learning Representations*, 2019. 2, 7

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015. 7

[27] Ryuichi Kiryo, Gang Niu, Marthinus Christoffel du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with

non-negative risk estimator. In *Advances in Neural Information Processing Systems*, pages 1675–1685, 2017. 1, 2, 7

[28] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 6

[29] Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *Proceedings of the 8th International Conference on Learning Representations*, 2020. 5

[30] Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. Partially supervised classification of text documents. In *Proceedings of the Nineteenth International on Machine Learning*, pages 387–394, 2002. 2

[31] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *Proceedings of the 5th International Conference on Learning Representations*, 2017. 7

[32] Chuan Luo, Pu Zhao, Chen Chen, Bo Qiao, Chao Du, Hongyu Zhang, Wei Wu, Shaowei Cai, Bing He, Saravanakumar Rajmohan, and Qingwei Lin. PULNS: positive-unlabeled learning with effective negative sample selector. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 8784–8792, 2021. 2

[33] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of European Conference on Computer Vision*, volume 11206, pages 185–201, 2018. 1

[34] Curtis G. Northcutt, Tailin Wu, and Isaac L. Chuang. Learning with confident examples: Rank pruning for robust classification with noisy labels. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, 2017. 2, 7

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67, 2020. 1

[36] Harish G. Ramaswamy, Clayton Scott, and Ambuj Tewari. Mixture proportion estimation via kernel embeddings of distributions. In *Proceedings of the 33nd International Conference on Machine Learning*, volume 48, pages 2052–2060, 2016. 2

[37] Patrice Y. Simard, Yann LeCun, John S. Denker, and Bernard Victorri. Transformation invariance in pattern recognition-tangent distance and tangent propagation. In *Neural Networks: Tricks of the Trade*, volume 1524, pages 239–27. Springer, 1996. 5

[38] Guangxin Su, Weitong Chen, and Miao Xu. Positive-unlabeled learning from imbalanced data. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 2995–3001, 2021. 1, 2, 7

[39] Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998. 4

[40] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, 2017. 6

[41] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, 2020. 5

[42] Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. PEBL: web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):70–81, 2004. 2

[43] Bangzuo Zhang and Wanli Zuo. Reliable negative extracting based on knn for learning from positive and unlabeled examples. *Journal of Computers*, 4(1):94–101, 2009. 2

[44] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the 6th International Conference on Learning Representations*, 2018. 2, 5

[45] Ya-Lin Zhang, Longfei Li, Jun Zhou, Xiaolong Li, Yujiang Liu, Yuanchao Zhang, and Zhi-Hua Zhou. POSTER: A PU learning based system for potential malicious URL detection. In *Proceedings of the Conference on Computer and Communications Security*, pages 2599–2601, 2017. 1

[46] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, pages 321–328, 2003. 2