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.

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 \( P(y = 1) \), 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 emerged 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.


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 $p(x, y)$ be the underlying probability distribution of $(\mathcal{X}, \mathcal{Y})$, and $p(x)$ be the marginal distribution of the input. Then the sets of positive and negative samples could be denoted as:

$$X_P = \{x\}^{n_P} \sim p_P(x),$$
$$X_N = \{x\}^{n_N} \sim p_N(x),$$

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

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

$$p(x) = \pi_P \cdot p_P(x) + \pi_N \cdot p_N(x),$$

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

$$R = \mathbb{E}_{(x,y) \sim p(x,y)} [\mathbb{I}(\hat{y}, y)],$$

with $\hat{y} = f(x) \in \{0, 1\}$ denoting the predicted label of $x$ by $f$; $\mathbb{I}(\hat{y}, y)$ refers to the zero-one error, which equals 0 when $\hat{y} = y$, and otherwise 1. Unfortunately, $\mathbb{I}(\hat{y}, y)$ is discontinuous, making the model hard to optimize. Therefore, a surrogate loss $l(\cdot, \cdot)$ defined on $f(x)$ 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 $X_{PU} = X_L \cup X_U$ where $X_L$ or $X_U$ 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:

$$X_L = \{x\}^{n_L} \sim p_P(x),$$
$$X_U = \{x\}^{n_U} \sim p_U(x) = p(x).$$

3.2. Label distribution alignment

The key to PU learning lies in the ways of incorporating unlabeled data $X_U$ into the training process. Without knowing any label in $X_U$, we cannot directly calculate a loss like $\ell(f(x), y)$. 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., $\mathbb{E}_{(x,y) \sim p(x,y)} [y] = \pi_P$. Motivated by this, we propose to use the class prior $\pi_P$ 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 $X_L$, the expectation of predicted labels should be equal to 1. Meanwhile, in $X_U$, positives are expected to take over $\pi_P$ proportion, and $\pi_N$ for negatives. Thus, the expectation over $X_U$ should be matched to $\pi_P$. 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:

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

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

$$R = \pi_P \mathbb{E}_{x \sim p_P(x)} [1 - \hat{y}] + \pi_N \mathbb{E}_{x \sim p_N(x)} [\hat{y}],$$

$$= \pi_P \left[ \mathbb{E}_{x \sim p_P(x)} [\hat{y}] - 1 \right] + \pi_N \left[ \mathbb{E}_{x \sim p_N(x)} [\hat{y}] \right],$$

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:

$$\mathbb{E}_{x \sim p_U(x)} [\hat{y}] - \mathbb{E}_{x \sim p(x,y)} [y]$$

$$= \pi_P \mathbb{E}_{x \sim p_P(x)} [\hat{y}] - \pi_P + \pi_N \mathbb{E}_{x \sim p_N(x)} [\hat{y}]$$

$$= - \pi_P R_P + \pi_N R_N.$$
Then we could easily represent $R_N$ by $R_P$ and the expectations of label distributions over unlabeled data:

$$
\pi_N R_N = \pi_P R_P + E_{x \sim pu(x)} [\hat{y}] - E_{x \sim pu(x)} [y] = \pi_P R_P + E_{x \sim pu(x)} [\hat{y}] - \pi_P.
$$

(10)

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:

$$
R = 2\pi_P R_P + E_{x \sim pu(x)} [\hat{y}] - \pi_P.
$$

(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 $E_{x \sim p(x)} [\hat{y}]$ and $\pi_P$ 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 $E_{x \sim p(x)} [\hat{y}]$ is exactly $\pi_P$. Therefore, we propose a variant risk by putting an absolute function into $R$:

$$
R_{lab} = 2\pi_P R_P + \frac{E_{x \sim pu(x)} [\hat{y}] - \pi_P}{R_U}.
$$

(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 $f(x)$. Namely, $\hat{y} = \frac{1}{2} (\text{sgn}(f(x)) + 1)$. 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 $E_x [\hat{y}]$ and compute it by the conditional probability of $\hat{y}$ given $x$:

$$
E_x [\hat{y}] = E_x [\Pr (\hat{y} = 1|x) \cdot 1 + \Pr (\hat{y} = 0|x) \cdot 0].
$$

(13)

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

$$
s = \frac{1}{1 + \exp [-f(x)]}.
$$

(14)

In this manner, we could alternatively employ the differentiable $E_x [s]$ 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:

$$
\hat{R}_{lab} = 2\pi_P \frac{1}{n_L} \sum_{x \in X_L} s - 1 + \frac{1}{n_U} \sum_{x \in X_U} s - \pi_P,
$$

(15)

where $\hat{R}_L$ and $\hat{R}_U$ denote the empirical risk estimator of $R_P$ and $R_U$, respectively.
Proposition 1. For a class of $b$-uniformly bounded functions $F$ with the VC dimension $\mathcal{V}$, with a probability at least $1 - \delta$, it holds that:

$$
R \leq 2\hat{R}_{lab} + 8\pi_p \cdot C \sqrt{\frac{\mathcal{V}}{n_L}} + 12\pi_p \sqrt{\frac{\ln 4/\delta}{2n_L}} \\
+ 4C \sqrt{\frac{\mathcal{V}}{n_U}} + 6\sqrt{\frac{\ln 4/\delta}{2n_U}},
$$

(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 calculated as $\pi_p$ regardless 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:

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

(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 $x_1, x_2 \in X_{PU}$ convexly with the mixing proportion $\lambda$ chosen from a Beta distribution:

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

(18)

$$
\lambda' = \max(\lambda, 1 - \lambda),
$$

(19)

$$
x' = \lambda' x_1 + (1 - \lambda') x_2.
$$

(20)

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

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

(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 $x'$, while $s_1(s_2)$ refers to the corresponding soft label (i.e., predicted score) of $x_1(x_2)$.

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 $\lambda'$ 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:

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

(22)

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

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

(23)

where $\mu$ adjusts the importance of $L_{ent}$; $\nu$ and $\gamma$ control the strength of $L_{mix}$ and $L'_{ent}$, respectively.
### 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 GeForce 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 potential NaN error.

---

\(^1\)Dubey, S. Alzheimer’s Dataset. Available online: https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images
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 $5 \times 10^{-4}$; weight decay is set as $5 \times 10^{-3}$. Dist-PU first experiences a warm-up stage of several epochs without Mixup, then trains with Mixup for another 60 epochs, where $\mu$ is also with a cosine annealing scheduling to mitigate overfitting. Moreover, the values of hyperparameters $\mu, \nu, \gamma$ and $\alpha$ are searched within the range of $[0,0.1], [0,10], [0,0.3]$ and $[0.1,10]$, respectively.

### 4.2. Comparison 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.

### 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 regularization, 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/ VII. (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 w/ VI and V w/ 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-
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 $\mu$ for entropy minimization on $X_{\text{dir}}$.** Parameter $\mu$ 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 $\mu$ without Mixup. In Fig. 4, we can see that when $\mu$ locates between 0 and 0.1, all the metrics except Rec. are relatively stable. However, a $\mu$ larger than 0.5 heavily degenerates the model performance.

**Effectiveness of $\alpha$ for Beta distribution in Mixup.** Parameter $\alpha$ controls the shape of Beta distribution in Mixup. The higher $\alpha$ is, the narrower its curve will be and $\lambda$ will be more likely near 0.5. Fig. 5 shows that rationally larger $\alpha$ helps improve the performance. Understandably, a larger $\alpha$ means more chance to get a $\lambda$ 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 $\nu$ and $\gamma$ for Mixup.** Parameter $\nu$ controls the power of Mixup on alleviating confirmation bias.

In Fig. 6, we see that the larger $\nu$ is, the better confirmation bias is weakened and the higher ACC our algorithm could reach. It also shows that a moderately larger $\gamma$, 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


[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


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


[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.


