Positive-Unlabeled Learning from Imbalanced Data

Guangxin Su[1], **Weitong Chen**[1], **Miao Xu**[1,2*]
[1]The University of Queensland, Brisbane QLD4072, Australia
[2]RIKEN AIP, Tokyo 103-0027, Japan
guangxinsu6@gmail.com, {weitong.chen, miao.xu}@uq.edu.au

Abstract

Positive-unlabeled (PU) learning deals with the binary classification problem when only positive (P) and unlabeled (U) data are available, without negative (N) data. Existing PU methods perform well on the balanced dataset. However, in real applications such as financial fraud detection or medical diagnosis, data are always imbalanced. It remains unclear whether existing PU methods can perform well on imbalanced data. In this paper, we explore this problem and propose a general learning objective for PU learning targeting specially at imbalanced data. By this general learning objective, state-of-the-art PU methods based on optimizing a consistent risk estimator can be adapted to conquer the imbalance. We theoretically show that in expectation, optimizing our learning objective is equivalent to learning a classifier on the oversampled balanced data with both P and N data available, and further provide an estimation error bound. Finally, experimental results validate the effectiveness of our proposal compared to state-of-the-art PU methods.

1 Introduction

In recent years, mobile wallets are set to become more and more common [JuniperResearch, 2018]. Due to its popularity, the security of mobile wallets is highly concerned. To protect users' money, the digital payment platform needs to detect risky accounts and give warnings before any fraud happens. In practice, the public security office usually provides a list of illegal accounts as positive (P) data with which a classifier can be trained. However, because it is not certain whether any account outside the list is trustable or not, treating them as negative (N) may bring unnecessary noise into the system. To get rid of noise, the classifier needs to be trained on only P data and unlabeled (U) data.

Such kind of problems is formed into a positive-unlabeled (PU) learning problem [Denis, 1998] in which P and U data are available and *no* negative (N) data is provided. PU learning has applicability in the fields of not only financial fraud detection, but also Alzheimer's disease diagnosis [Chen *et al.*, 2020], information retrieval [Dupret and Piwowarski, 2008] and link prediction [Hsieh *et al.*, 2015]. Recently, many efforts [du Plessis *et al.*, 2015; Kiryo *et al.*, 2017; Shi *et al.*, 2018; Chen *et al.*, 2020; Chen *et al.*, 2021] have been devoted to case-control PU learning [Menon *et al.*, 2015] and efficient algorithms based on deep neural networks are proposed [Kiryo *et al.*, 2017; Chen *et al.*, 2020].

Although existing PU methods have been shown to be successful in benchmark datasets [Kiryo *et al.*, 2017; Chen *et al.*, 2020], they may not perform well on tasks such as fraud detection or medical diagnosis. The reason is that in these tasks, different from the benchmark datasets, the data is highly imbalanced [Chawla, 2010; He and Garcia, 2009], i.e., if the data are i.i.d. sampled from the underlying data distribution, the number of P data is much smaller than the number of N data. For example, among all the mobile wallet accounts, only a small amount of accounts are illegal; in all medical check-ups, only a few patients have the disease. However, most of the current PU methods do not consider special techniques to handle the imbalance. Even worse, some of the PU methods weigh the risk incurred on P data by the small class prior, further enlarging the impact of imbalance. There are a few works touching the imbalanced PU learning problem. [Xie and Li, 2018; Sakai *et al.*, 2018] optimize directly the AUC in PU learning. However, as F1 is counted as one of the metrics suitable for imbalanced learning, a good AUC does not necessarily mean a good F1, as it cares about the relative order of real outputs, instead of the classification result. [Chen *et al.*, 2021] dealt with cost-sensitive PU learning, which required the cost to be known, while in our study such information is not available.

In normal classification when both P and N data are available, the imbalanced learning problem has been widely investigated. Related methods based on a single model can be divided into four categories. One category is sampling. Either oversampling [Chawla *et al.*, 2002; Yan *et al.*, 2019; Guo *et al.*, 2019] is used to increase the number of minority data, or undersampling [Peng *et al.*, 2019] is used to decrease the number of majority data. Such methods cannot be easily adapted to improve PU learning. Undersampling cannot be used due to that no N data is available. For oversampling, due to that state-of-the-art PU methods weigh the risks of P data by the small class prior [du Plessis *et al.*, 2014;

*Contact Author

du Plessis *et al.*, 2015; Kiryo *et al.*, 2017; Shi *et al.*, 2018; Chen *et al.*, 2020], no matter how many data points are oversampled their effect on learning is dramatically reduced due to the weighting.

Another category of methods is based on cost-sensitive learning [Wang *et al.*, 2017; Khan *et al.*, 2018; Huang *et al.*, 2020; Byrd and Lipton, 2019], i.e., assigning different costs to majority class and minority class. Such kind of methods can also be effective when the cost is appropriately set up but most of the time, setting up the cost accurately is not possible due to lacking of domain knowledge. Recently, imbalanced learning methods based on metric-learning and semi-supervised learning are also proposed. For methods based on metric-learning [Wang *et al.*, 2018; Viola *et al.*, 2020], N data is required to learn an appropriate distance metric, which is unavailable in PU learning. For semi-supervised learning methods, their performance depends on learning a good enough initial classifier to label U data [Kim *et al.*, 2020; Yang and Xu, 2020]; however, as we have discussed, the current PU method may not learn such a good enough classifier on imbalanced data. Besides these methods on a single model, ensemble methods [Liu *et al.*, 2020] are also proposed to combine the results of several base methods, such as combining oversampling and cost-sensitive methods. Despite the good performance of them on PN data, they inherited the single model methods' disadvantages to handle PU data.

In this paper, we propose a general re-weighting strategy for imbalanced PU learning. We assume that oversampling will work well to tackle the imbalance problem if both P and N data are available. Based on such an assumption, we carefully design the weights for the risks on P data and U data and show theoretically that in expectation, the risk of the balanced PN data can be perfectly estimated through the available PU data using our proposed re-weighting strategy. We further give an empirical error bound on the classifier learned empirically. Experimental results have verified that using our general re-weighting strategy can enhance the F1 performance of PU methods on handling imbalanced data.

Note that designing a re-weighting strategy in PU learning is not as easy as in PN learning. In PN learning, a straightforward strategy is to give a large weight for P data and a small weight for N data. In PU learning, due to the unavailability of N data, the risk of P data being treated as negative is also calculated and deducted from the overall optimization objective [du Plessis *et al.*, 2015; Kiryo *et al.*, 2017]. Additionally, a risk on U data is incorporated. Due to the existence of these different risks, the reweighting for PU learning becomes much more complex than PN learning. Although it looks straightforward to improve the weights on P data and keep the weights on U data, our analysis in Sec. 2 shows that reweighting in this way will distort the target data distribution and fail to guarantee a statistical consistency as our proposal.

Our contribution are summarized as follows

- We propose a general learning objective for PU learning with imbalanced data. A reweighting strategy is designed in this general learning objective. As far as we know, this is the first work specially dealing with such a practical problem.
- We theoretically verify that in expectation, optimizing such a learning objective on available PU data can enable learning a classifier on balanced PN data which is not available. We also give an estimation error bound to guarantee the performance.
- We show empirically that when the proposed learning objective is used, existing PU methods can be adapted to better handle imbalanced data: their performance on imbalanced data is dramatically improved.

2 Methodology

2.1 Formulation and Background

Assume there is an underlying distribution $P(X, Y)$, where $X \in \mathbb{R}^d$ is the input and $Y \in \{-1, +1\}$ is the output random variables. In case-control PU learning [Menon *et al.*, 2015], P data of size n_p are sampled from $P(x|Y = +1)$ and U data of size n_u are sampled from $P(x)$. $\pi = P(Y = 1)$ represents the class prior of positive label. In most cases, it is assumed to be known. It can also be estimated from the data if it is unknown [Elkan and Noto, 2008; du Plessis *et al.*, 2017]. Based on the given P and U data, our objective is to learn a classifier $f : \mathbb{R}^d \to \{-1, +1\}$ which can successfully classify an instance x. In practice, we often learn a function $g : \mathbb{R}^d \to [0, 1]$, whose output value can represent the posterior probability of $P(Y|x)$.

In PU learning, the following risk is used as the learning objective [du Plessis *et al.*, 2015]

$$\begin{aligned}\mathcal{R}_{\text{pu}}(g) = \pi\mathbb{E}_{P(x|Y=+1)}[\ell(g(x), +1)] + \\ (\mathbb{E}_{P(x)}[\ell(g(x), -1)] - \pi\mathbb{E}_{P(x|Y=+1)}[\ell(g(x), -1)]),\end{aligned} \tag{1}$$

where $\ell(\cdot, \cdot)$ is any trainable surrogate loss function of zero-one loss [du Plessis *et al.*, 2015], such as the sigmoid loss

$$\ell_{\text{sig}}(g(x), y) = \frac{1}{1 + \exp(yg(x))} \tag{2}$$

We can see that the loss on P data $\mathbb{E}_{P(x|Y=+1)}[\ell(g(x), +1)]$ is weighted by π, which is very small in imbalanced data. Additionally, the loss treating P data as negative, i.e., $\mathbb{E}_{P(x|Y=+1)}[\ell(g(x), -1)])$, is additionally measured and deduced from the learning objective. Such an item never exists in PN learning.

The following estimator is then optimized based on the given PU data

$$\begin{aligned}\widehat{\mathcal{R}}_{\text{pu}}(g) = \frac{\pi}{n_p}\sum_{x_i \in P}\ell(g(x_i), +1) + \\ \left(\frac{1}{n_u}\sum_{x_i \in U}\ell(g(x), -1) - \frac{\pi}{n_p}\sum_{x_i \in P}\ell(g(x_i), -1)\right)\end{aligned} \tag{3}$$

named as uPU (unbiased PU) [du Plessis *et al.*, 2015].

In practice, it is found that due to the strong fit ability of deep neural networks, the second term of Eq. (3) can go much lower than zero. However, theoretically, this term is used to estimate $(1 - \pi)\mathbb{E}_{P(x|Y=-1)}[\ell(g(x), -1)])$, which should always be non-negative. In this way, Kiryo *et al.* proposed to optimize the following non-negative risk

$$\widehat{\mathcal{R}}_{\text{nnpu}}(g) = \frac{\pi}{n_p}\sum_{x_i\in P}\ell(g(x_i),+1) + \max\Bigg(0, \frac{1}{n_u}\sum_{x_i\in U}\ell(g(x),-1) - \frac{\pi}{n_p}\sum_{x_i\in P}\ell(g(x_i),-1)\Bigg)$$

and gave the nnPU (non-negative PU) method [du Plessis *et al.*, 2017]. From then on, nnPU has become the state-of-the-art method for PU learning using deep neural networks. And many algorithms for PU learning are proposed based upon nnPU [Xu *et al.*, 2019; Hsieh *et al.*, 2019; Chen *et al.*, 2020].

2.2 Algorithm

In this part, we assume that oversampling can help combat imbalanced data in PN learning. Based on such an assmuption, we employ both the data generation process of case-control PU learning and oversampling to solve the imbalanced PU learning problem.

The data generation process is depicted in Figure 1. Imagine we have an imbalanced PN dataset $\mathcal{D}_{\text{PN}}$ based on which the available PU dataset is generated. The PN dataset is sampled from the underlying distribution $P(X,Y)$, which contains $\hat{n}_p$ positive data and $\hat{n}_n$ negative data. If both $\hat{n}_p$ and $\hat{n}_n$ are large enough, sampling from this dataset can approximate sampling according to the original data distribution. In this way, we have $\hat{n}_p/\hat{n}_n = \pi/(1-\pi)$. If we oversample the P data in $\mathcal{D}_{\text{PN}}$ according to $P(x|Y=1)$, we will have a balanced dataset $\mathcal{D}_{\text{balancedPN}}$ with distribution $P_{\text{balanced}}(X,Y)$, which contains m_p positive data and m_n negative data. $m_p \gg \hat{n}_p$, $m_n = \hat{n}_n$, $m_p/m_n = \pi'/(1-\pi')$, and $\pi' = P_{\text{balanced}}(Y=1)$. π' is around 0.5 such that the newly generated PN data is balanced. We assume that the learned classifier on $\mathcal{D}_{\text{balancedPN}}$ has a good performance on metrics suitable for imbalanced learning such as F1 score. In this way, what we want to do is to learn a classifier on the balanced PN data to tackle the imbalanced problem. The risk we want to optimize for $\mathcal{D}_{\text{balancedPN}}$ is

$$\mathcal{R}_{\text{balancePN}}(g) = \mathbb{E}_{P_{\text{balanced}}(x,y)}\ell(g(x),y). \tag{4}$$

Note that we have oversampled from $\mathcal{D}_{\text{PN}}$ the P data only. It means that although the joint distribution $P_{\text{balanced}}(X,Y)$ is different from the original joint probability $P(X,Y)$, the class conditional probability remains unchanged, i.e., $P(x|Y=1) = P_{\text{balanced}}(x|Y=1)$. We also have $P(x|Y=-1) = P_{\text{balanced}}(x|Y=-1)$ due to that we do nothing to the N data. In this way, we have

$$\begin{aligned}\mathcal{R}_{\text{balancePN}}(g) &= \mathbb{E}_{P_{\text{balanced}}(x,y)}\ell(g(x),y) \\ &= \pi'\mathbb{E}_{P_{\text{balanced}}(x|Y=1)}\ell(g(x),+1) + (1-\pi')\mathbb{E}_{P_{\text{balanced}}(x|Y=-1)}\ell(g(x),-1) \\ &= \pi'\mathbb{E}_{P(x|Y=1)}\ell(g(x),+1) + (1-\pi')\mathbb{E}_{P(x|Y=-1)}\ell(g(x),-1).\end{aligned} \tag{5}$$

Since

$$\begin{aligned}&(1-\pi)\mathbb{E}_{P(x|Y=-1)}\ell(g(x),-1) \\ =\ &\mathbb{E}_{P(x)}\ell(g(x),-1) - \pi\mathbb{E}_{P(x|Y=+1)}\ell(g(x),-1),\end{aligned} \tag{6}$$

we can have

Theorem 1. *For a joint distribution $P_{\text{balanced}}(X,Y)$, the objective risk is defined in Eq. (4). If there is another distribution $P(x,y)$ which has different class prior $P(Y)$ with $P_{\text{balanced}}(X,Y)$ but the same class conditional probability $P(x|Y)$, we have*

$$\mathcal{R}_{\text{balancePN}}(g) = \pi'\mathbb{E}_{P(x|Y=+1)}\ell(g(x),+1) + \frac{1-\pi'}{1-\pi}\left[\mathbb{E}_{P(x)}\ell(g(x),-1) - \pi\mathbb{E}_{P(x|Y=+1)}\ell(g(x),-1)\right] \tag{7}$$

in which $\pi = P(y=1)$ and $\pi' = P_{\text{balanced}}(y=1)$.

Proof. The proof can be derived by combining the above Eqs. (5) and (6) . □

Theorem 1 gives us a guide on how to learn a classifier (in expectation) for the balanced PN data but the only availability is imbalanced PU data. In practice, we need to optimize an empirical estimation of $\mathcal{R}_{\text{balancePN}}$, which is

$$\widehat{\mathcal{R}}_{\text{balancePN}}(P,U) = \frac{\pi'}{n_p}\sum_{x_i\in P}\ell(g(x_i),+1) + \frac{1-\pi'}{n_u(1-\pi)}\sum_{x_i\in U}\ell(g(x),-1) - \frac{(1-\pi')\pi}{n_p(1-\pi)}\sum_{x_i\in P}\ell(g(x_i),-1). \tag{8}$$

Since we also want to make use of the deep neural networks as our base learner, we may face the same problem as nnPU [Kiryo *et al.*, 2017]. In this way, we will optimize a similar non-negative loss as nnPU, which is,

$$\widehat{\mathcal{R}}_{\text{nnBalancePN}}(P,U) = \frac{\pi'}{n_p}\sum_{x_i\in P}\ell(g(x_i),+1) + \max\left(0, \frac{1-\pi'}{n_u(1-\pi)}\sum_{x_i\in U}\ell(g(x),-1) - \frac{(1-\pi')\pi}{n_p(1-\pi)}\sum_{x_i\in P}\ell(g(x_i),-1)\right). \tag{9}$$

We want to minimize the above risk to get a classifier $\widehat{g}(x;\theta)$. Practically, $\widehat{\mathcal{R}}_{\text{nnBalancePN}}$ is optimized through a gradient ascend strategy employed also in [Kiryo *et al.*, 2017; Han *et al.*, 2020; Ishida *et al.*, 2020]. We call our proposed method **ImbalancednnPU** and give the procedures in Algorithm 1. In this strategy, we define

$$\widehat{\mathcal{R}}_N(P,U) = \frac{1-\pi'}{n_u(1-\pi)}\sum_{x_i\in U}\ell(g(x),-1) - \frac{(1-\pi')\pi}{n_p(1-\pi)}\sum_{x_i\in P}\ell(g(x_i),-1).$$

Then when $\widehat{\mathcal{R}}_N$ is larger than zero, we do normal gradient descend (Line 6); if $\widehat{\mathcal{R}}_N$ is smaller than zero, we do gradient ascent instead (Line 8). This is the same strategy for nnPU.

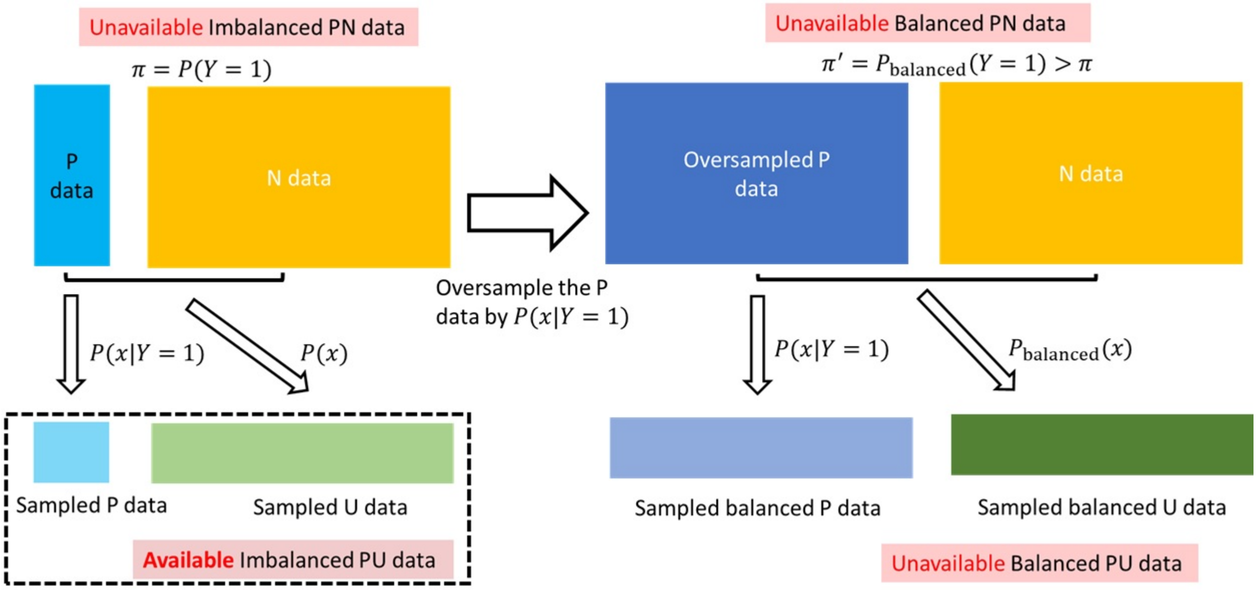


Figure 1: The illustration of the original imbalanced PN data, the generated imbalanced PU data, the oversampled balanced PN data, the generated balanced PU data and how they are generated. Note that the only available data are the imbalanced PU data.

Algorithm 1 ImbalancednnPU

Input: Training data P and U
Parameter: class prior π and π', MAX_E
Output: classifier $\widehat{g}(x;\theta)$

Let $\mathcal{A}$ be an SGD-like optimizer such as Adam [Kingma and Ba, 2015] and $t = 1$
while $t <$ MAX_E **do**
  Shuffle P and U into b mini-batches, each is represented as P_i and U_i respectively
  for $i = 1$ to b **do**
    if $\widehat{\mathcal{R}}_N(\mathrm{P}_i,\mathrm{U}_i) \geq 0$ **then**
      Update θ by $\mathcal{A}$ with the gradient $\nabla_\theta \widehat{\mathcal{R}}_{\text{nnBalancePU}}(\mathrm{P}_i,\mathrm{U}_i)$
    else
      Update θ by $\mathcal{A}$ with the gradient $-\nabla_\theta \widehat{\mathcal{R}}_N(\mathrm{P}_i,\mathrm{U}_i)$
    end if
  end for
end while
return θ and the corresponding classifier $\widehat{g}(x;\theta)$

A naive way: reweighting P data by π'. Note that a straightforward strategy is to reweight the P data by π' but keeping the weights on U data unchanged. Note that in this strategy, there is an implicity assumption that the $P_{\text{balanced}}(x)$ is equal to $P(x)$ in Figure 1. However, since we have

$$P_{\text{balanced}}(x) = \pi' P_{\text{balanced}}(x|Y=1) + (1-\pi')P_{\text{balanced}}(x|Y=-1),$$

$P_{\text{balanced}}(x)$ cannot be equal to $P(x)$ unless $\pi = \pi'$, i.e., no oversampling has ever happened. In another word, the naive strategy to reweigh only P data by π' resembles learning the classifier on data sampled from an unknown distribution which cannot be generated from the original data set by oversampling method.

PU methods beyond nnPU. There are many methods proposed recently for PU learning , and some of them based on the original nnPU have achieved state-of-the-art performance. For these methods, additional tricks such as self-learning, meta-learning, or knowledge distillation is added beyond nnPU [Chen *et al.*, 2020]. However, at their basis, an nnPU method should be run. In this way, the optimization objective in them is also updated to Eq. 9 and their performance is expected to be improved on imbalanced data.

2.3 Theoretical Properties

Note that in Section 2.2, we first derive the expected risk Eq. (7). Then we have an estimation of it based on the given data in Eq. (8). So if we optimize Eq. (8), how would the achieved classifier be different from the one optimizing Eq. (7)? In this section, we answer this question by giving an estimation error bound.

To have an estimation error bound, we first need to make assumption on the surrogate loss function $\ell(\cdot,\cdot)$. We assume ℓ is Lipschitz continuous with respect to its first argument, and the Lipschitz constant is L_ℓ. We further assume ℓ is symmetric, i.e.,

$$\ell(t,+1) + \ell(t,-1) = 1.$$

Note that these two assumptions are satisfied by the commonly used surrogate loss function sigmoid loss in Eq. (2). We will also use sigmoid loss in our experiments.

In learning, suppose we have a function class $\mathcal{G}$. Let

$$g^* = \arg\min_{g\in\mathcal{G}} \mathcal{R}_{\text{balancePN}}(g)$$

and

$$\widehat{g}_{\mathrm{PU}} = \arg\min_{g\in\mathcal{G}} \widehat{\mathcal{R}}_{\mathrm{balancePN}}(g).$$

We denote the Rademacher complexity [Shalev-Shwartz and Ben-David, 2014] of $\mathcal{G}$ for the sampling of size n sampled with probability P as $\mathfrak{R}_{n,P}(\mathcal{G})$. We then have the following theoretical results

Theorem 2. *Assume ℓ is symmetric and Lipschitz continuous with respect to its first argument, and the Lipschitz constant is L_ℓ. For any $\delta > 0$ with probability at least $1-\delta$, we have*

$$\begin{aligned}&\mathcal{R}_{\mathrm{balancePN}}(\widehat{g}_{\mathrm{PU}}) - \mathcal{R}_{\mathrm{balancePN}}(g^*) \le \\ &\quad \frac{4(\pi'+\pi-2\pi'\pi)}{1-\pi} L_\ell \mathfrak{R}_{n_p,P(x|Y=1)}(\mathcal{G}) + \\ &\quad\quad \frac{2(1-\pi')}{1-\pi} L_\ell \mathfrak{R}_{n_u,P(x)}(\mathcal{G}) + \\ &\frac{2(\pi'+\pi-2\pi'\pi)}{1-\pi}\sqrt{\frac{\ln(4/\delta)}{2n_p}} + \frac{2(1-\pi')}{1-\pi}\sqrt{\frac{\ln(4/\delta)}{2n_u}}.\end{aligned}$$

We put the detailed proof of Theorem 2 into the supplementary files.

When $\mathfrak{R}$ is upper bounded for $\mathcal{G}$, we have that

$$\mathcal{R}_{\mathrm{balancePN}}(\widehat{g}_{\mathrm{PU}}) - \mathcal{R}_{\mathrm{balancePN}}(g^*) \to 0$$

in $\mathcal{O}(1/\sqrt{n_p} + 1/\sqrt{n_u})$, i.e., theoretically we still need enough P data to make the classifier satisfied. In practice, we observed that our proposal performs better than classical PU method such as nnPU [Kiryo *et al.*, 2017]. We plan to explore this interesting direction on more tight theoretical results in the future work.

Note that when $\pi' = 0.5$ and ℓ is the zero-one loss, our learning objective Eq. (4) can be seen as the arithmetic mean of true positive rate and true negative rate. In this way, [Menon *et al.*, 2013] provides a regret bound if we can estimate $P(Y|x)$ in a precise way. Motivated by this work, we will set $\pi' = 0.5$ and the final prediction is made by $\mathrm{sgn}(g(x) - 0.5)$.

3 Experiments

In this section, we compare the performance of our proposed ImbalancednnPU with state-of-the-art PU methods on imbalanced datasets. We will show that state-of-the-art PU methods, although have been shown to be effective on balanced PU data, fails to be superior on imbalanced PU data. We also adapt several imbalanced learning methods for normal classification into PU learning, and compare with them. All the codes are implemented in Python 3 and Pytorch 1.7, and running on a GPU server with CUDA 11.1.

Dataset. In previous PU work, datasets such as CIFAR10 [1] have been widely used [Kiryo *et al.*, 2017]. These multi-class datasets are processed into balanced binary classification data by picking five categories out of all ten categories as P, with π from 0.40 to 0.50. In our tasks, following existing works to test the scalability of our proposal, we also use the CIFAR10 data. Different from [Kiryo *et al.*, 2017], which divided the data into animal and non-animal, we pick only one category from all the ten categories as P, and treat all other data as N. In this way, we have 10 different datasets, with π approximately equaling 0.1. In each dataset, there are $50,000$ training data and $10,000$ test data as provided by the original CIFAR10. To make the training data into a PU learning problem, we follow [Kiryo *et al.*, 2017] to sample $1,000$ positive instances and treat them as P; all the training data are used as U, i.e., $n_u = 50,000$.

Methods. We compare our proposed ImbalancednnPU and other algorithms,

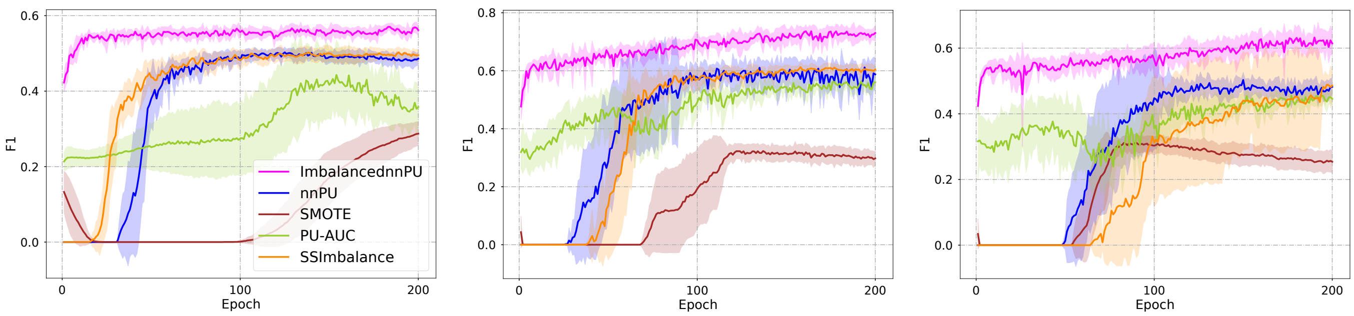
- Our method. We will show the empirical results for two versions of our proposed method, depending whether they use additional labeled data to do meta-learning. One of the method is the **ImbalancednnPU** we proposed. Note that our proposed learning objective is general such that any method optimizing a risk similar to nnPU can be enhanced to handle imbalanced data by our proposal. In this way, we further enhance self-PU [Chen *et al.*, 2020], the meta-learning method to handle imbalanced data, and call the method **ImbalancedSelfPU**.

We will compare with

- PU learning method *without* meta-learning. For PU learning method, we compare with **nnPU** [Kiryo *et al.*, 2017] which is the state-of-the-art method in PU learning. For nnPU, we use the same network structure and the recommended parameter tuning strategy as in the original paper.
- PU learning method *with* meta-learning. Method **Self-PU** [Chen *et al.*, 2020] is a meta-learning method for PU learning based on nnPU. In such a method, additional data with groundtruth label is used for meta-learning.
- Oversampling method. We compare with classical oversampling method **SMOTE** [Chawla *et al.*, 2002]. Since SMOTE requires to do kNN first, we set $k = 5$ as suggested for SMOTE. For SMOTE, the number of P data is oversampled to be $50,000$, the same as the number of U data.
- Semi-supervised imbalanced learning method. We use the strategy proposed in [Yang and Xu, 2020], which first trains a classifier using nnPU. After initial training for 100 epochs, U data is labeled by this classifier and the training starts again by optimizing a combination of PU risk and PN risk. The parameter to weight the loss on labeled data and U data is set as recommended in the original paper. The method is called **SSImbalance**.
- PU-AUC. **PU-AUC** directly optimizes AUC. We include it into comparison.

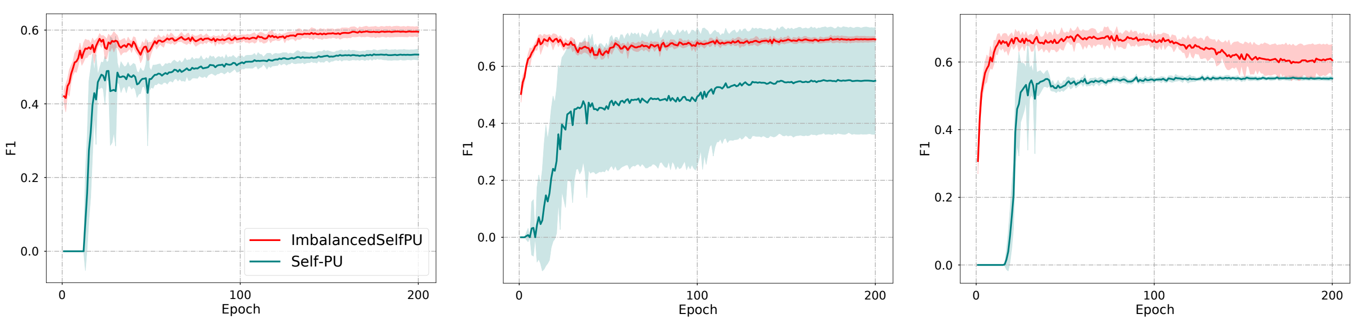
Settings. For our proposed ImbalancednnPU, we set $\pi' = 0.5$ and $\pi = 0.1$. We use the same network structure as [Kiryo *et al.*, 2017], i.e., a 13-layer CNN with ReLU and Adam as the optimizer. We tune the hyper-parameters step size and weight decay by a grid select from $\{10^{-10}, 10^{-9}, \ldots, 10^{0}\}$ for all methods based on neural networks. All the other hyperparameters in the network are set as default.

[1] https://www.cs.toronto.edu/ kriz/cifar.html



(a) airplane (b) automobile (c) truck

Figure 2: F1 score on CIFAR10 dataset without meta-learning when treating airplane, automobile or truck as P label. The dense line shows the average of 10 trials, and the shadow area show the standard deviation.



(d) airplane (e) automobile (f) truck

Figure 3: F1 score on CIFAR10 dataset with meta-learning when treating airplane, automobile or truck as P label. The dense line shows the average of 10 trials, and the shadow area show the standard deviation.

Evaluation. The same as [Kiryo *et al.*, 2017], we set the number of epochs to be 200. For PU data induced by each label, 10 random PU datasets are generated and we show the average results of the 10 trials, as well as the standard deviation. On the imbalanced test set, we will show the F1 score. We additionaly show the performance on AUC in our supplementary files due to space limitation.

Results without meta-learning. The experimental results of methods without meta-learning on F1 score are shown in Figure 2. We only show the results on treating three labels, “aeroplane”, “automobile” and “truck”, as P and put other seven result in the supplementary. From these experimental results, we can see that our porposed ImbalancednnPU achieved the best results among all compared methods on most datasets. Among the compared methods, the semi-supervised imbalanced method [Yang and Xu, 2020] performs the best among all baselines; however, its performance strongly relies on a satisfiable base classifier.

Results with meta-learning. The experimental results of methods with meta-learning on F1 score are shown in Figure 3. We can see that our proposal improves the performance of self-PU, and sometimes, the variance can also be reduced.

4 Conclusion

In this paper, we propose a novel reweighting strategy for PU learning from imbalanced data. In this method, we oversample the implicit PN data to balance, and then use risk on the available PU data to mimic the risk on the balanced PN data. We prove the equality of these two risks in expectation, and also give the estimation error bound. Based on the strategy, we propose ImbalancednnPU and further ImbalancedSelfPU. Experimental results verify the effectiveness.

There are many directions worth investigating in the future. One interesting problem is the theoretical studies. In our paper, although we have given an estimation error bound, it did not show much the merit of our proposed method comparing against state-of-the-art PU method such as nnPU [Kiryo *et al.*, 2017]. In this way, we may need a new theoretical results sensitive to the difference between π and π'.

References

[Byrd and Lipton, 2019] Jonathon Byrd and Zachary Chase Lipton. What is the effect of importance weighting in deep learning? In *ICML*, pages 872–881, 2019.

[Chawla *et al.*, 2002] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *JAIR*, 16:321–357, 2002.

[Chawla, 2010] Nitesh V. Chawla. Data mining for imbalanced datasets: An overview. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook, 2nd ed*, pages 875–886. Springer, 2010.

[Chen *et al.*, 2020] 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 *ICML*, pages 1510–1519, 2020.

[Chen *et al.*, 2021] Xiuhua Chen, Chen Gong, and Jian Yang. Cost-sensitive positive and unlabeled learning. *Information Sciences*, 558:229–245, 2021.

[Denis, 1998] François Denis. PAC learning from positive statistical queries. In *ALT*, pages 112–126, 1998.

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

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

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

[Dupret and Piwowarski, 2008] Georges Dupret and Benjamin Piwowarski. A user browsing model to predict search engine click data from past observations. In *IJCAI*, pages 331–338, 2008.

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

[Guo *et al.*, 2019] Ting Guo, Xingquan Zhu, Yang Wang, and Fang Chen. Discriminative sample generation for deep imbalanced learning. In *IJCAI*, pages 2406–2412, 2019.

[Han *et al.*, 2020] Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor W. Tsang, and Masashi Sugiyama. SIGUA: forgetting may make learning with noisy labels more robust. In *ICML*, pages 4006–4016, 2020.

[He and Garcia, 2009] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *TKDE*, 21(9):1263–1284, 2009.

[Hsieh *et al.*, 2015] Cho-Jui Hsieh, Nagarajan Natarajan, and Inderjit S. Dhillon. PU learning for matrix completion. In Francis R. Bach and David M. Blei, editors, *ICML*, pages 2445–2453, 2015.

[Hsieh *et al.*, 2019] Yu-Guan Hsieh, Gang Niu, and Masashi Sugiyama. Classification from positive, unlabeled and biased negative data. In *ICML*, pages 2820–2829, 2019.

[Huang *et al.*, 2020] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Deep imbalanced learning for face recognition and attribute prediction. *TPAMI*, 42(11):2781–2794, 2020.

[Ishida *et al.*, 2020] Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. Do we need zero training loss after achieving zero training error? In *ICML*, pages 4604–4614, 2020.

[JuniperResearch, 2018] JuniperResearch. Digital wallets transforming the way we pay? *juniperresearch.com*, 2018.

[Khan *et al.*, 2018] Salman H. Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous Ahmed Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *TNNLS*, 29(8):3573–3587, 2018.

[Kim *et al.*, 2020] Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *NeurIPS*, 2020.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

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

[Liu *et al.*, 2020] Zhining Liu, Pengfei Wei, Jing Jiang, Wei Cao, Jiang Bian, and Yi Chang. MESA: boost ensemble imbalanced learning with meta-sampler. In *NeurIPS*, 2020.

[Menon *et al.*, 2013] Aditya Krishna Menon, Harikrishna Narasimhan, Shivani Agarwal, and Sanjay Chawla. On the statistical consistency of algorithms for binary classification under class imbalance. In *ICML*, pages 603–611, 2013.

[Menon *et al.*, 2015] Aditya Krishna Menon, Brendan van Rooyen, Cheng Soon Ong, and Bob Williamson. Learning from corrupted binary labels via class-probability estimation. In *ICML*, pages 125–134, 2015.

[Peng *et al.*, 2019] Minlong Peng, Qi Zhang, Xiaoyu Xing, Tao Gui, Xuanjing Huang, Yu-Gang Jiang, Keyu Ding, and Zhigang Chen. Trainable undersampling for class-imbalance learning. In *AAAI*, pages 4707–4714, 2019.

[Sakai *et al.*, 2018] Tomoya Sakai, Gang Niu, and Masashi Sugiyama. Semi-supervised AUC optimization based on positive-unlabeled learning. *MLJ*, 107(4):767–794, 2018.

[Shalev-Shwartz and Ben-David, 2014] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014.

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

[Viola *et al.*, 2020] Rémi Viola, Rémi Emonet, Amaury Habrard, Guillaume Metzler, and Marc Sebban. Learning from few positives: a provably accurate metric learning algorithm to deal with imbalanced data. In *IJCAI*, pages 2155–2161, 2020.

[Wang *et al.*, 2017] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NIPS*, pages 7029–7039, 2017.

[Wang *et al.*, 2018] Nan Wang, Xibin Zhao, Yu Jiang, and Yue Gao. Iterative metric learning for imbalance data classification. In *IJCAI*, pages 2805–2811, 2018.

[Xie and Li, 2018] Zheng Xie and Ming Li. Semi-supervised AUC optimization without guessing labels of unlabeled data. In *AAAI*, pages 4310–4317, 2018.

[Xu *et al.*, 2019] Miao Xu, Bingcong Li, Gang Niu, Bo Han, and Masashi Sugiyama. Revisiting sample selection approach to positive-unlabeled learning: Turning unlabeled data into positive rather than negative. *arXiv*, 1901.10155, 2019.

[Yan *et al.*, 2019] Yuguang Yan, Mingkui Tan, Yanwu Xu, Jiezhang Cao, Michael K. Ng, Huaqing Min, and Qingyao Wu. Oversampling for imbalanced data via optimal transport. In *AAAI*, pages 5605–5612, 2019.

[Yang and Xu, 2020] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In *NeurIPS*, 2020.