Margin Based PU Learning

Tieliang Gong,[1] **Guangtao Wang,**[2] **Jieping Ye,**[2] **Zongben Xu,**[1] **Ming Lin**[2]
[1]School of Mathematics and Statistics,
Xi'an Jiaotong University, Xi'an 710049, Shaanxi, P. R. China
[2]Department of Computational Medicine and Bioinformatics,
University of Michigan, Ann Arbor, Michigan 48109, USA

Abstract

The PU learning problem concerns about learning from positive and unlabeled data. A popular heuristic is to iteratively enlarge training set based on some margin-based criterion. However, little theoretical analysis has been conducted to support the success of these heuristic methods. In this work, we show that not all margin-based heuristic rules are able to improve the learned classifiers iteratively. We find that a so-called large positive margin oracle is necessary to guarantee the success of PU learning. Under this oracle, a provable positive-margin based PU learning algorithm is proposed for linear regression and classification under the truncated Gaussian distributions. The proposed algorithm is able to reduce the recovering error geometrically proportional to the positive margin. Extensive experiments on real-world datasets verify our theory and the state-of-the-art performance of the proposed PU learning algorithm.

Introduction

As an important branch of classification problems, learning a classifier from positive and unlabeled data, also known as the PU learning, has attracted a great deal of attention in machine learning (Letouzey, Denis, and Gilleron 2000; Scott and Blanchard 2009; Plessis, Niu, and Sugiyama 2015; Blanchard, Lee, and Scott 2010) and data mining communities (Liu et al. 2003; Fung et al. 2006; Elkan and Noto 2008). Different from supervised learning and semi-supervised learning, the training set in PU learning consists of a set of positive instances and a large number of unlabeled instances. The main goal of PU learning is to make full use of the unlabeled data together with the limited positive data to learn a reliable predictive model. To this end, a lot of attempts have been made on designing efficient PU learning algorithms. These algorithms can be roughly divided into two categories, characterized by two different ways of exploring unlabeled data. One category can be boiled down to two-stage methods (Liu et al. 2002; Yu, Han, and Chang 2002; Li and Liu 2003; Yu 2005) which first select high confidence negative instances from the unlabeled data and then merge them with available positive instance to train the classifier. The other category is the one-stage methods (Elkan and Noto 2008; Blanchard, Lee, and Scott 2010) which regard all unlabeled data as negative instances in training.

Recently, the iterative multi-stage training strategy enjoys increasing popularity in curriculum learning (CL) (Bengio et al. 2009) and self-paced learning (SPL) (Kumar, Packer, and Koller 2010). The basic idea is to start the learning task with a small set of easy instances. After the initial training stage, more difficult instances are gradually appended to the training set. Many real-world applications verify the effectiveness of this training strategy (Pentina, Sharmanska, and Lampert 2015; Supancic and Ramanan 2013; Jiang et al. 2014; Zhao et al. 2015). The main reason for its success comes down to the easy-to-hard information revealed in the consecutive training stages.

Inspired by this idea, a natural question has raised: can PU learning benefit from a similar strategy? To answer this question, we develop a general framework to study the PU learning problem. We find that the iterative multi-stage training cannot guarantee to improve the classifier under agnostic PU learning setting. Therefore additional assumptions must be implicitly or explicitly made if one expects success. To this end, we introduce a large positive margin oracle which claims that the positive instances are located far away from the decision boundary. Different from the conventional margin definition, the positive margin oracle only relates with positive instances while the conventional margin is defined on both positive and negative instances (Vapnik 1998; Schapire et al. 1998). This oracle is critical to the success of PU learning. It allows us to design an efficient iterative multi-stage algorithm, named Positive Margin-based PU (PMPU) learning algorithm, to solve the PU regression and classification problems. The main idea of PMPU is to estimate the labels of unlabeled data in each iteration according to the positive margin shrinkage and then retrain the classifier based on random sampling. The detailed algorithm is given in Section 3.

From theoretical side, there are several early works on studying the error bounds of PU learning. Denis et al. (Denis 1998) established the PAC bound for PU learning. Plessis et al. (Plessis, Niu, and Sugiyama 2014) proved that the error of PU classification is no worse than $2\sqrt{2}$ times of the fully supervised one when the numbers of labeled and unlabeled examples are equal. Niu et al. (Niu et al. 2016) pointed out that PU learning sometimes outperforms supervised learning if the class-prior probability is known exactly. Albeit with more or less success in explaining the efficiency of PU learning, these existing results cannot explain why iterative training will improve the classifier in PU learning. This paper addresses this issue with the large-positive margin oracle. Particularly, we prove that under the truncated Gaussian distribution with large positive margin, the estimator sequence generated by PMPU converges to the optimal classifier. Both the recovery error and misclassification error decay geometrically with respect to the positive margin parameter.

The main contribution of this paper is summarized as follows:

- Propose the large positive margin oracle, base on which we design a positive margin-based PU learning (PMPU) algorithm.
- We prove that the recovery error of PMPU decays on order of $\mathcal{O}(\exp(-\tau^2 t))$where τ is the positive margin parameter and t is the number of PU iteration. To the best of our knowledge, this is the first theoretical result on the related assimilation.
- Comprehensive experiments on large scale datasets to demonstrate the efficiency and efficacy of PMPU.

The rest of this paper is organized as follows: Section 2 sets the notation and problem statement. Section 3 introduces the proposed PMPU algorithm. Section 4 establishes the corresponding recovery error bound for PMPU. Sections 5 presents numerical studies on large-scale datasets. Section 6 concludes our work.

Background and Notation

Suppose that the feature-label pairs of training instances $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ are drawn independently and identically from a fixed unknown distribution $P(\mathbf{x}, y)$, where $\mathcal{X}$ denotes the feature space and $\mathcal{Y}$ is the label space. We use m, d to denote the number of samples and the feature dimension respectively. Let $X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \mathbb{R}^{d\times N}$ be the feature matrix, where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector of the i-th instance. The label of the i-th instance is $y_i \in \{-1, 1\}$ and the label vector $\mathbf{y} = [y_1, y_2, \cdots, y_N]^\top$.

In linear regression, the response is generated via a linear mapping

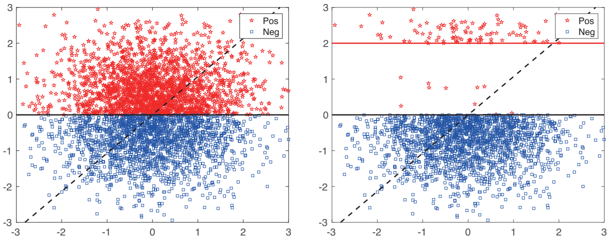
$$\mathbf{z} = X^\top \mathbf{w}^*$$

where $\mathbf{z} = [z_1, z_2, \cdots, z_N]$ is the response vector and $\mathbf{w}^*$ is the target weight vector. On the other hand, in binary classification we cannot observe the response value $\mathbf{z}$ directly but its sign

$$\mathbf{y} = \operatorname{sign}(X^\top \mathbf{w}^*)$$

where $\operatorname{sign}(\cdot)$ is element-wise applied.

In this paper, we are interested in linear classifiers of the form $f(\mathbf{x}) = \operatorname{sign}(X^\top \mathbf{w})$, where $\mathbf{w}$ is the weight vector we need to learn from training data. Denote $(X_P, \mathbf{y}_P)$ as the positive sample set, and X_U as the unlabeled sample set. The goal of PU learning is to utilize X_U together with $(X_P, \mathbf{y}_P)$ to train a reliable classifier $\mathbf{w}$ close to $\mathbf{w}^*$. To this end, we develop a positive margin-based PU learning (PMPU) algorithm based on the large positive margin oracle.



(a)

(b)

Figure 1: (a) Gaussian distribution; (b) Truncated Gaussian distribution with positive margin $\tau = 2$.

Positive Margin Based PU Learning

In this section, we present the large positive margin oracle and the working scheme of PMPU. We will focus on the high-level intuition of our algorithm. The rigorous theoretical analysis is postponed to the next section.

Before we introduce the positive large margin oracle, we give an agnostic example where iterative re-training fails in PU learning. Consider a two dimensional classification problem given in Figure 1 (a). Suppose that the ground-truth distribution follows the standard Gaussian, where the red star denotes the positive instances and the blue square denotes the unlabeled instances. Following the PU learning setting, only a small portion of the positive instances are labeled in the training set. The initial classifier, denoted by the dash line, is fitted based on the randomly sampled training instances (one positive instance from positive sample set and one negative instance from unlabeled sample set). The optimal classifier is represented as the horizontal real line. The initial classifier tells us that the instances above the dash line are considered to be positive and the instances below the dash line are negative. We then perform re-training by randomly sampling on this new artificially labeled dataset to learn a new classifier. Since the artificial labels deviate from the ground-truth badly, the new classifier is hardly better than the initial one.

However, this situation can be avoided by the large positive margin oracle which claims that all positive instances are located far away from the optimal decision boundary. It emphasizes the role of positive margin while the conventional definition of margin in SVM and Boosting depends on both the positive and negative instances. This oracle makes the iterative re-training strategy feasible in PU learning. As illustrated in Figure 1 (b), we truncated the density function of the Gaussian distribution between $y = 0$ and $y = 2$ (the red real line), i.e., the large positive margin oracle $\tau = 2$. Note that we allow noise instances in the truncated region. Comparing to Figure 1 (a), the dash line in (b) is more likely rotated clockwise because the initial classifier provides higher accuracy, which guarantees more correct artificial labels in next estimation. Hence the large positive margin assumption is critical to the success of iterative PU learning.

Algorithm 1 Positive Margin Based PU Learning (PMPU)

```
1: Require: positive margin parameter $\tau$, unlabeled sample set $X_U$, positive sample set $X_P$.
2: Randomly sample $m_+ = |X_P|$ instances from $X_U$ to generate negative set $\mathbb{S}_-^{(0)}$. Denote the sampled training set by $X^{(0)} := X_P \bigcup \mathbb{S}_-^{(0)}$.
3: Train an initial classifier $\mathbf{w}^{(0)}$ on $X^{(0)}$.
4: for $t = 1, 2, \cdots, T$ do
5:     Re-sample $m_t$ instances from $X_U$ as training set $X^{(t)}$.
6:     For any $\mathbf{x} \in X^{(t)}$, if $\mathbf{x}^\top \mathbf{w}^{(t)} \geq \eta_t$ then labeled as positive otherwise negative.
7:     $\hat{\mathbf{w}}^{(t)} \leftarrow \mathbf{w}^{(t-1)} - \frac{1}{\lambda_\tau m_t} X^{(t)} \Delta \mathbf{y}_t$.     (1)
8:     $\mathbf{w}^{(t)} = \hat{\mathbf{w}}^{(t)} / \|\hat{\mathbf{w}}^{(t)}\|_2$ .
9: end for
10: Output: Final estimator $\mathbf{w}^{(T)}$.
```

Inspired by the above toy examples, we develop a Positive Margin Based PU learning algorithm (PMPU) summarized in Algorithm 1. The large positive margin oracle τ plays a key role in PMPU. It is an intrinsic parameter of the data distribution. A large τ will lead to high accuracy and small sampling complexity. The initial classifier is fitted by the instances from positive set and random samples from the unlabeled set X_U. All instances sampled from X_U are regarded as negative instances at initialization step.

The key step of PMPU is to construct an iteration sequence $\{\mathbf{w}^{(t)}\}_{t=1}^T$ to approximate $\mathbf{w}^*$. To this end, denote $X^{(t)}$ as the feature matrix of instances by re-sampling and $\mathbf{y}^{(t)}$ as the predicted label of $X^{(t)}$ at step t and

$$\hat{\mathbf{y}}^{(t)} = \mathcal{S}_\tau(X^\top \mathbf{w}^{(t)}),$$

where for any vector $\mathbf{z}$

$$\{\mathcal{S}_\tau(\mathbf{z})\}_i = \begin{cases} 1 & \text{if } \mathbf{z}_i \geq \tau \\ -1 & \text{otherwise.} \end{cases}$$

We call the above step as the positive margin shrinkage. Let $\Delta \mathbf{y}_t = \mathbf{y}^{(t)} - \hat{\mathbf{y}}^{(t)}$. Then we can construct the iterative sequence by least square gradient descent as in Eq. (1) . As only the direction of $\mathbf{w}^*$ matters, we enforce $\mathbf{w}^{(t)}$ normalized to unit length on Line 8. The positive margin shrinkage requires a parameter τ. In practice τ should be tuned via cross-validation and a good range of τ can be estimated from the margin distribution of the initial classifier.

The value of m_t should be increased with t. As m_t instances are sampled from the unlabeled set, it could be as many as we want. In our proof we only require m_t to be larger than $\mathcal{O}(1/\|\mathbf{w}^{(t)} - \mathbf{w}^*\|_2)$. In practice we keep m_t to be a small number in order to minimize the computation cost and increase m_t if the validation error stop decreasing.

Our theoretical analysis shows that $\|\mathbf{w}^{(t)} - \mathbf{w}^*\| \to 0$ with high probability. The recovery error decays geometrically with respect to the large positive oracle τ and the PU iteration step t. For one step PMPU iteration, the sampling complexity is with order $\mathcal{O}[d \log d \exp^2(-\tau^2)/\epsilon^2]$. This implies that we need fewer positive instances when the positive margin is large.

Theoretical results

In this section we study the theoretical properties of PMPU. Please refer to appendix for the detailed proof of these lemmas and the main theorems.

Our analysis is built on the matrix Bernstein's inequality whose proof could be found in many textbooks (Tropp 2015). To abbreviate our concentration bound, we frequently use C_δ to denote a factor consisting of logarithm terms in δ and any other necessary variables that do not affect the order of our bound.

Lemma 1. *(Matrix Bernstein's inequality) Consider a finite sequence $\{S_i\}$ of independent random matrices of dimension $d_1 \times d_2$. Assume that each matrix has uniformly bounded deviation from its mean:*

$$\|S_i - \mathbb{E}S_i\| \leq L \quad \textit{for each index } i.$$

Introduce the random matrix $Z = \sum_i S_i$ and let $\nu(Z)$ be the matrix variance of Z where

$$\begin{aligned} \nu(Z) =& \max\Big\{\|\mathbb{E}(Z - \mathbb{E}Z)(Z - \mathbb{E}Z)^\top\|, \\ & \|\mathbb{E}(Z - \mathbb{E}Z)^\top (Z - \mathbb{E}Z)\|\Big\} \\ =& \max\Big\{\|\sum_i \mathbb{E}(S_i - \mathbb{E}S_i)(S_i - \mathbb{E}S_i)^\top)\|, \\ & \|\sum_i \mathbb{E}(S_i - \mathbb{E}S_i)^\top (S_i - \mathbb{E}S_i)\|\Big\}. \end{aligned}$$

Then

$$\mathbb{E}\|Z - \mathbb{E}Z\| \leq \sqrt{2\nu(Z)\log(d_1 + d_2)} + \frac{1}{3} L \log(d_1 + d_2).$$

Furthermore, for all $t > 0$,

$$\mathbb{P}\{\|Z - \mathbb{E}Z\| \geq t\} \leq (d_1 + d_2) \exp\left\{-\frac{t^2/2}{\nu(Z) + Lt/3}\right\}.$$

The next lemma shows that on the truncated Gaussian distribution, the gradient of least square loss is still an isometric mapping as in the Gaussian distribution.

Lemma 2. *Let $X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$. The first dimension of each $\mathbf{x}_i$ is a truncated Gaussian random variable, and the remaining $d - 1$ dimensions are i.i.d. copies of $\mathcal{N}(0, 1)$. Then for any $\mathbf{w} \in \mathbb{R}^d$ with $\|\mathbf{w}\| = 1$, we have*

$$\mathbb{E}\mathrm{sign}(\langle \mathbf{x}, \mathbf{w} \rangle)\mathbf{x} = \lambda_\tau \mathbf{w},$$

where $\lambda_\tau = \sqrt{\frac{2}{\pi} + \frac{\exp(-\frac{\tau^2}{2}) - 1}{2}}$.

Different from the standard calculation of correlation parameter λ (Plan and Vershynin 2013) which claims that $\lambda = \sqrt{2/\pi}$, in our case the correlation parameter λ_τ is a function of τ due to the large positive margin oracle. Clearly $\lambda_\tau \to \lambda$ as $\tau \to 0$.

We now present the main theorem which establishes an upper bound between the unnormalized output $\hat{\mathbf{w}}^{(t)}$ and the theoretical optimal $\mathbf{w}^*$.

Theorem 1. *In Algorithm 1, suppose $\mathbf{x}_i$'s are independently sampled from the truncated Gaussian distribution with positive margin τ. Then with probability at least $1-\delta$,*

$$\|\hat{\mathbf{w}}^{(t)} - \mathbf{w}^*\|_2 \leq \gamma \|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|$$

where $\gamma := C_\delta \exp(-c_2\tau^2)$ for some constant C_δ and c_2.

Theorem 1 shows that $\|\hat{\mathbf{w}}^{(t)} - \mathbf{w}^*\|_2$ will decay linearly at least. However we have to normalize $\mathbf{w}^{(t)}$ as we only care about the direction in classification. We must show that after normalization the recovery error is also decreased.

Proposition 1. *When τ is larger than a universal constant,*

$$\|\mathbf{w}^{(t)} - \mathbf{w}^*\|_2 \leq 3\gamma \|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|.$$

Proposition 1 can be derived by a few steps of linear algebra thus the proof is omitted here. Finally, we prove that the overall convergence rate of our PMPU algorithm is

$$\|\hat{\mathbf{w}}^{(t)} - \mathbf{w}^*\|_2 \leq (3\gamma)^t \|\mathbf{w}^{(0)} - \mathbf{w}^*\|. \tag{2}$$

Compared with the previous theoretical research on PU learning (Niu et al. 2016; Plessis, Niu, and Sugiyama 2014), Eq. (2) provides the recovery error bound based on iterative training strategy without knowing class-prior probability. We prove that $\mathbf{w}^{(t)}$ will globally converge to $\mathbf{w}^*$ and the recovery error only related with the large positive margin oracle.

Let's focus on the main information Eq. (2) conveys. The recovery error of PMPU depends on the initial difference $\|\mathbf{w}^{(0)} - \mathbf{w}^*\|_2$ and the large positive margin parameter τ. Note that the initial difference can be regarded as a constant because it can be directly computed after the first training. The convergence rate of PMPU is dominated by γ, which is on order of $\mathcal{O}(\exp(-\tau^2))$. It shows that the recovery error decays geometrically with large positive margin parameter τ and iteration number t. Observe that for large τ, the recovery error would be small. Specifically, $\|\mathbf{w}^{(t)} - \mathbf{w}^*\|_2 \to 0$ as $\tau \to \infty$. It is worth noting that the conventional margin theory claims that the larger the margin over the training sample, the better the generalization performance (Vapnik 1998; Schapire et al. 1998; Breiman 1999). Our analysis supports this viewpoint.

The following Theorem demonstrates the RIP (Restricted Isometry Property)-type condition for our problem.

Theorem 2. *With the same condition of Theorem 1. For two normalized vectors $\mathbf{w}, \mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}\|_2 = 1$, $\|\mathbf{w}^*\|_2 = 1$, with probability at least $1-\delta$,*

$$\begin{aligned}&\|\frac{1}{m_t\lambda_\tau}\big(X\mathrm{sign}(X^\top\mathbf{w}) - X\mathrm{sign}(X^\top\mathbf{w}^*)\big) - (\mathbf{w}-\mathbf{w}^*)\|_2 \\ &\leq \epsilon\max(\|\mathbf{w}-\mathbf{w}^*\|_2, \|\mathbf{w}-\mathbf{w}^*\|_2^{\frac{1}{2}})\end{aligned}$$

provided $m_t \geq C_\delta d\exp(\frac{-\tau^2}{2})\big(\frac{4+\tau}{\sqrt{2\pi}} + \frac{1}{2}\big)/\epsilon^2$.

Proof. (Sketch) Without loss of generality, let $\mathbf{w}^* = (1, 0, \cdots, 0)$ and define the correlation coefficient $\lambda_\tau = \mathbb{E}\mathbf{y}_i(\langle\mathbf{x}_i, \mathbf{w}\rangle) = \mathbb{E}\mathrm{sign}(\langle\mathbf{x}_i, \mathbf{w}^*\rangle)\langle\mathbf{x}_i, \mathbf{w}\rangle$. Lemma 2 gives the result. Next, define random variable

$$B_i = \frac{1}{\lambda_\tau}[\mathbf{x}_i\mathrm{sign}(\langle\mathbf{x}_i, \mathbf{w}\rangle) - \mathbf{x}_i\mathrm{sign}(\langle\mathbf{x}_i, \mathbf{w}'\rangle)]$$

by Lemma 2, we have

$$\mathbb{E}B_i = \mathbf{w} - \mathbf{w}'.$$

Further we set

$$Z_i = B_i - \mathbb{E}B_i,$$

then bound the terms $\max_i \|Z_i\|_2$, $\|\mathbb{E}Z_i^\top Z_i\|$ and $\|\mathbb{E}Z_iZ_i^\top\|_2$ respectively. By applying Lemma 1, we obtain the final result. □

Table 1: Statistics of real-world datasets

Data	Categories	# Train	# Test	# Features
WAVEFORM	3	2500	2500	21
COVERTYPE	7	11340	565892	54
CIFAR-10	10	50000	10000	4096
RCV-1	53	15564	518571	47236
MNIST	10	60000	10000	784
CIFAR-100	100	50000	10000	4096

We can draw similar conclusions made in Theorem 1. The convergence rate is controlled by large positive margin oracle τ. It decreases geometrically w.r.t. τ. Besides, the larger τ is, the higher the accuracy we can obtain. The sample complexity is on order of $\mathcal{O}(d\exp(-\tau^2)/\epsilon^2)$. It can be observed that we can use a small number of instances to achieve the guaranteed error when large τ is specified.

Experiments

To demonstrate the performance of PMPU learning algorithm in practice, we apply our algorithm to several real-world datasets.

Datasets

We evaluated the proposed PMPU on 6 real-world classification datasets, including *WAVEFROM, COVERTYPE , MNIST, RCV-1*[1], *CIFAR-10, CIFAR-100*[2]. These datasets cover a range of application domains such as text, hand digits and images. Table 1 summarizes the dataset statistical. The numbers of training of the 6 datasets vary from 2500 to 60000, the numbers of testing vary from 2500 to 565892 and the feature dimensions vary from 21 to 47236.

The training set and testing set are prespecified for all datasets. Note that the COVERTYPE and RCV-1 contain plenty of test examples but a small number of training examples. The features of *CIFAR-10* and *CIFAR-100* are extracted by VGG net (Simonyan and Zisserman 2014). We use the HOG feature for *MNIST*, and raw data for the other datasets.

Experimental Setting

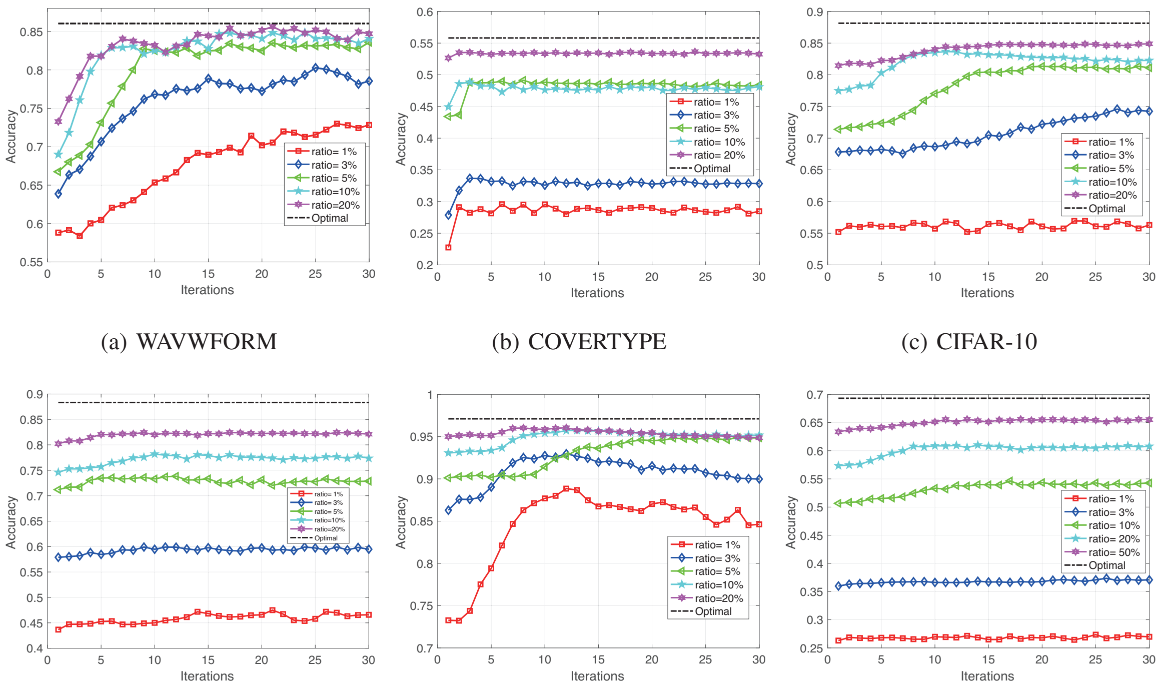
All these datasets are multi-class classification tasks, we use linear SVM classifier with one-versus-all strategy for all experiments and implemented by LIBSVM (Chang and Lin

[1] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/multiclass.html

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

Table 2: Accuracy ($\%$) with different sampling ratio

Data	Lowest/Highest accuracy(sampling ratio)					Optimal
WAVEFORM	58.38/72.99(1%)	63.87/80.27(3%)	66.75/83.59 (5%)	68.99/84.87 (10%)	73.27/85.63(20%)	86.04
COVERTYPE	22.76/29.58(1%)	27.86/33.67(3%)	43.42/48.85 (5%)	44.93/49.14 (10%)	51.32/53.65(20%)	55.81
CIFAR-10	55.19/56.94(1%)	67.55/74.59(3%)	71.39/81.34(5%)	77.48/83.76(10%)	81.45/84.89(20%)	88.14
RCV-1	43.67/47.51(1%)	57.89/59.93(3%)	71.21/73.83(5%)	74.60/78.19(10%)	80.22/82.44(20%)	88.33
MNIST	73.22/88.85(1%)	86.29/92.98(3%)	90.12/94.92(5%)	93.07/95.74(10%)	94.87/96.05(20%)	97.11
CIFAR-100	26.31/27.35(1%)	35.98/37.35(3%)	50.66/54.64(10%)	57.34/61.01(20%)	63.36/65.61(50%)	70.67



(a) WAVWFORM

(b) COVERTYPE

(c) CIFAR-10

(d) RCV-1

(e) MNIST

(f) CIFAR-100

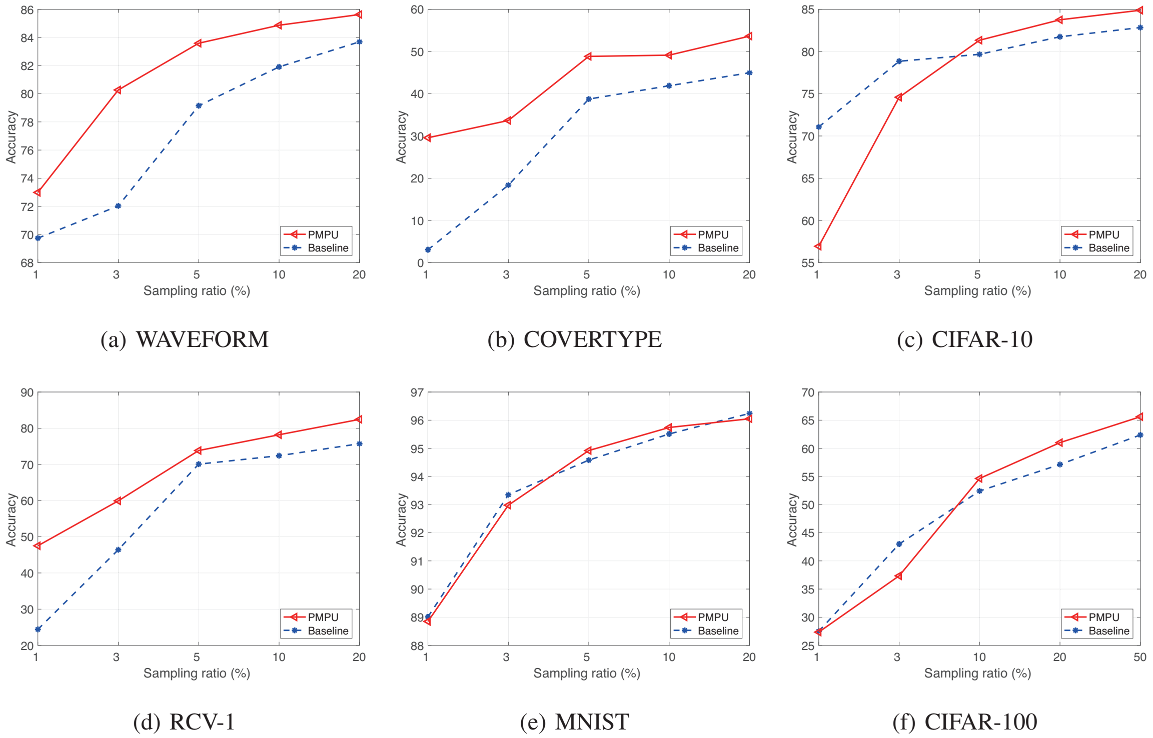
Figure 2: Test accuracy of MPU with different sampling ratio.

2011). The regularization parameter C for each experiment is selected from the set $\{0\} \cup \{10^{-6}, 10^{-5}, \cdots, 10^5, 10^6\}$ by 5-fold cross validation. The optimal classifier is obtained by batch training on all training samples.

Let $(X, \mathbf{y}) = (X_P, \mathbf{y}_P) \cup X_U$, where $(X_P, \mathbf{y}_P)$ denotes the positive sample set and X_U the unlabeled sample set. The main goal of PMPU is to make full use of $(X_P, \mathbf{y}_P)$ together with X_U to construct a reliable classifier. In our experiments, the training process is done as follows:

- Given the training set, we perform random sampling on each class according to the pre-specified sampling ratio r to generate $(X_P, \mathbf{y}_P)$, and keep the remaining as X_U.
- Train an initial model based on the sampled instances, and then make predictions on X_U to generate $(X_U, \mathbf{y}_U)$.
- Perform random sampling again on $(X_U, \mathbf{y}_U)$, and get $(X_Q, \mathbf{y}_Q)$ named as the queried sample set, then retrain the model based on $(X_P, \mathbf{y}_P) \bigcup (X_Q, \mathbf{y}_Q)$.

Repeat the above procedure until the terminate condition is satisfied. In our experiments, the large positive margin oracle τ is determined according to the distribution of decision value generated by initial model. We set τ to be the 75% quantile of positive decision values predicted by the initial model for all datasets. We also set $|X_Q| = 3/4|X_U|$, where $|X_Q|$ denotes the number of re-sampled instances and $|X_U|$ the number of unlabeled instances. At the same time, the number of PU iterations is set to 30 for all experiments. We report both the lowest and highest accuracies of PMPU under different sampling ratio in Table 2. It should be mentioned that the *CIFAR-100* dataset extends *CIFAR-10* by increasing the number of categories to 100, while remains the same number of training and testing examples. Therefore, it is considered to be a more difficult classification task than *CIFAR-10*. In our experiments, the highest sampling ratio of *CIFAR-100* is set to be 50%. We illustrate the iteration pro-



(a) WAVEFORM

(b) COVERTYPE

(c) CIFAR-10

(d) RCV-1

(e) MNIST

(f) CIFAR-100

Figure 3: Comparisons on accuracy with different sampling ratio.

cess of PMPU on 6 datasets in Figure 2.

Results

Table 2 records both the lowest and highest accuracies of PMPU on 6 datasets under different sampling ratio. The optimal accuracy for each dataset is reported in the last column of Table 2. It can be observed that the accuracy of PMPU is close to the optimal one, even if sampling ratio r remains a small value (we set the largest sample ratio to 20% for most datasets).

Figure 2 plots the iteration process of PMPU on all 6 datasets. Note that the recovery error decays with an exponential rate, so it is expected that the PMPU should converge fast. As observed in the reported figures, the accuracy of PMPU indeed converges quickly to a stable level. Another observation can be made from these figures is that the improvement on accuracy is not prominent if $r \leq 1\%$, but remarkable when r keeps increasing. The main reason behind this phenomenon is that the low sampling ratio will result in an inaccurate estimation on initial model, and this initial model can not provide enough useful information for next iteration. However, higher sampling ratio will lead to a more reliable initial model which can offer more valuable information for the next iteration. It's worth noting that although PMPU cannot improve the accuracy remarkably at very low sampling ratio for some datasets (such as 1% for *CIFAR-10*, *CIFAR-100* and *RCV-1*), it doesn't degrade the classification performance. The iteration process tends to stable.

We also compared the proposed PMPU to the Label Random PU learning (LRPU) in (Elkan and Noto 2008), one of the state-of-the-art PU learning methods, which suggests a PU-learner predicts probabilities that differ by only a constant factor from the true conditional probabilities of being positive. This method adopts probability output as the classification criterion in binary classification. We follow the same criterion by extending it to the multi-class case. Figure 3 reports the comparisons on 6 datasets under different sampling ratio in terms of test accuracy. It can be observed that the test accuracies of both methods increase along the rising sampling ratio, and PMPU outperforms the LRPU for almost all datasets. Generally speaking, PMPU gets higher accuracy when sampling ratio is greater than 5%. The success comes down to the retraining strategy based on the large positive margin oracle of PMPU. We also notice that PMPU performs worse than LRPU on *CIFAR-10* and *CIFAR-100* when sampling ratio keeps very low (less than 5%). In this case, low sampling ratio will result in an inaccurate initial classifier which cannot provide enough information for next prediction.

In order to further verify the effectiveness of PMPU, we apply it to three classification algorithms: Random Forest (RF), Logistic Regression (LR) and Gradient Boosting Tree (GBT). For RF and GBT, the number of trees is fixed to 50 for all experiments. Table 3 records the accuracies of the three methods under different sampling ratios (1%, 5%, 10%), in which the results obtained by PMPU are based on 5-round re-training. It can be observed that PMPU always improve the baseline.

Table 3: Accuracy with different sampling ratio (%)

Data	Methods	Baseline accuracy (sampling ratio)			PMPU accuracy (sampling ratio)		
WAVEFORM	RF	0.6202(1%)	0.7587(5%)	0.8115(10%)	0.6527(1%)	0.7825(5%)	0.8291(10%)
	LR	0.5749 (1%)	0.6612(5%)	0.6845(10%)	0.6132(1%)	0.8033(5%)	0.8327(10%)
	GBT	0.6471(1%)	0.7863(5%)	0.8127(10%)	0.6671(1%)	0.8119(5%)	0.8355(10%)
COVERTYPE	RF	0.2975(1%)	0.3433(5%)	0.4067(10%)	0.3405(1%)	0.3728(5%)	0.4493(10%)
	LR	0.2343(1%)	0.3752(5%)	0.4238(10%)	0.2937(1%)	0.4121(5%)	0.4782(10%)
	GBT	0.2092(1%)	0.357(5%)	0.4494(10%)	0.2488(1%)	0.4065(5%)	0.4899(10%)
CIFAR-10	RF	0.57(1%)	0.721(5%)	0.7434(10%)	0.5892(1%)	0.7321(5%)	0.7596(10%)
	LR	0.5532(1%)	0.7134(5%)	0.7752(10%)	0.5753(1%)	0.7356(5%)	0.7934(10%)
	GBT	0.5558(1%)	0.6872(5%)	0.6841(10%)	0.5872(1%)	0.7257(5%)	0.7462(10%)
RCV-1	RF	0.2926(1%)	0.4531(5%)	0.5018(10%)	0.3125(1%)	0.4882(5%)	0.5324(10%)
	LR	0.4286(1%)	0.7087(5%)	0.7392(10%)	0.4674(1%)	0.8032(5%)	0.7784(10%)
	GBT	0.3116(1%)	0.4303(5%)	0.5039(10%)	0.3427(1%)	0.4684(5%)	0.5417(10%)
MNIST	RF	0.7878(1%)	0.9185(5%)	0.9378(10%)	0.83(1%)	0.9257(5%)	0.9456(10%)
	LR	0.7386(1%)	0.9037(5%)	0.9324(10%)	0.8455(1%)	0.9249(5%)	0.9489(10%)
	GBT	0.8328(1%)	0.8924(5%)	0.9268(10%)	0.8571(1%)	0.9245(5%)	0.9413(10%)
CIFAR-100	RF	0.1578(1%)	0.3549(5%)	0.4325(10%)	0.1624(1%)	0.3682(5%)	0.4476(10%)
	LR	0.2581(1%)	0.5066(5%)	0.5648(10%)	0.2612(1%)	0.5341(5%)	0.5927(10%)
	GBT	0.1497(1%)	0.3484(5%)	0.4013(10%)	0.1776(1%)	0.3599(5%)	0.4259(10%)

Let's make a further discussion of PMPU. First, the choice of large positive margin parameter τ is critical to PMPU. The theoretical analysis claims that the misclassification error rate decays geometrically with τ. A large τ will lead to a fast convergence rate. However, τ cannot be arbitrarily large in practical applications. It could be specified by cross-validation. Secondly, $|X_Q|$ plays an important role in PMPU. As demonstrated in Algorithm 1, PMPU needs to perform random sampling in each iteration after obtaining the initial model. The prediction on $\mathbf{y}_U$ determines the quality of the sampling set for the next iteration. A small $|X_Q|$ will decrease the classification accuracy. To guarantee the performance of PMPU, $|X_Q|$ can not be too small. Thirdly, the convergence rate of PMPU is closely related with the margin parameter. A large shrinkage parameter will lead to a fast convergence rate of PMPU.

Conclusion

In this paper, we propose a large positive margin oracle for the PU learning problem and design a provable efficient Positive Margin based PU (PMPU) learning algorithm. We analyze the performance of PMPU in terms of the recovery error and the misclassification error. The theoretical results show that the estimator generated by PMPU converges to the global optimal. The recovery error decays with an exponential rate w.r.t. the positive margin oracle τ. Experiments on large scale datasets demonstrate the effectiveness and efficacy of PMPU. Future work includes how to extend this analysis framework to more general distribution, e.g., subgaussian distribution, and how to relax linear classifier assumption to non-linear case. All these problems deserve further research.

Acknowledgement

This research was supported in part by NIH (RF1AG051710), NSF (III-1539991 and III-1539722), National 973 Programming (2013CB329404), and National Natural Science Foundation of China (No. 11671161 and 11131006).

References

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *International Conference on Machine Learning, ICML*, 41–48.

Blanchard, G.; Lee, G.; and Scott, C. 2010. Semi-supervised novelty detection. *Journal of Machine Learning Research* 11:2973–3009.

Breiman, L. 1999. Prediction games and arcing algorithms. *Neural Computation* 11(7):1493–1517.

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27.

Denis, F. 1998. PAC learning from positive statistical queries. In *Algorithmic Learning Theory, 9th International Conference, ALT*, 112–126.

Elkan, C., and Noto, K. 2008. Learning classifiers from only positive and unlabeled data. In *International Conference on Knowledge Discovery and Data Mining, SIGKDD*, 213–220.

Fung, G. P. C.; Yu, J. X.; Lu, H.; and Yu, P. S. 2006. Text classification without negative examples revisit. *IEEE Trans. Knowl. Data Eng.* 18(1):6–20.

Jiang, L.; Meng, D.; Yu, S.; Lan, Z.; Shan, S.; and Hauptmann, A. G. 2014. Self-paced learning with diversity. In *Advances in Nerual Information Processing Systems,NIPS*, 2078–2086.

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-paced learning for latent variable models. In *Advances in Nerual Information Processing Systems,NIPS*, 1189–1197.

Letouzey, F.; Denis, F.; and Gilleron, R. 2000. Learning from positive and unlabeled examples. In *Algorithmic Learning Theory, ALT*, 71–85.

Li, X., and Liu, B. 2003. Learning to classify texts using positive and unlabeled data. In *International Joint Conference on Artificial Intelligence,IJCAI*, 587–594.

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

Liu, B.; Dai, Y.; Li, X.; Lee, W. S.; and Yu, P. S. 2003. Building text classifiers using positive and unlabeled examples. In *International Conference on Data Mining, ICDM*, 179–188.

Niu, G.; Plessis, M.; Sakai, T.; Ma, Y.; and Sugiyama, M. 2016. Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In *Advances in Nerual Information Processing Systems,NIPS*, 1199–1207.

Pentina, A.; Sharmanska, V.; and Lampert, C. H. 2015. Curriculum learning of multiple tasks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 5492–5500.

Plan, Y., and Vershynin, R. 2013. Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach. *IEEE Trans. Information Theory* 59(1):482–494.

Plessis, M.; Niu, G.; and Sugiyama, M. 2014. Analysis of learning from positive and unlabeled data. In *Advances in Nerual Information Processing Systems (NIPS)*, 703–711.

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

Schapire, R. E.; Freund, Y.; Bartlett, P.; Lee, W. S.; et al. 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics* 26(5):1651–1686.

Scott, C., and Blanchard, G. 2009. Novelty detection: Unlabeled data definitely help. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS*, 464–471.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Supancic, J. S., and Ramanan, D. 2013. Self-paced learning for long-term tracking. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2379–2386.

Tropp, J. A. 2015. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning* 8(1-2):1–230.

Vapnik, V. 1998. *Statistical Learning Theory*. Wiley-Interscience.

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

Yu, H. 2005. Single-class classification with mapping convergence. *Machine Learning* 61(1-3):49–69.

Zhao, Q.; Meng, D.; Jiang, L.; Xie, Q.; Xu, Z.; and Hauptmann, A. G. 2015. Self-paced learning for matrix factorization. In *Proceedings of the Conference on Artificial Intelligence AAAI*, 3196–3202.