Positive Unlabeled Learning with Class-prior Approximation

Shizhen Chang, **Bo Du**[*] and **Liangpei Zhang**
School of Computer Science, State Key Lab of Information Engineering on Survey Mapping and Remote Sensing, Institute of Artificial Intelligence, National Engineering Research Center for Multimedia Software, Wuhan University
{szchang, dubo, zlp62}@whu.edu.cn

Abstract

The positive unlabeled (PU) learning aims to train a binary classifier from a set of positive labeled samples and other unlabeled samples. Much research has been done on this special branch of weakly supervised classification problems. Since only part of the positive class is labeled, the classical PU model trains the classifier assuming the class-prior is known. However, the true class prior is usually difficult to obtain and must be learned from the given data, and the traditional methods may not work. In this paper, we formulate a convex formulation to jointly solve the class-prior unknown problem and train an accurate classifier with no need of any class-prior assumptions or additional negative samples. The class prior is estimated by pursuing the optimal solution of gradient thresholding and the classifier is simultaneously trained by performing empirical unbiased risk. The detailed derivation and theoretical analysis of the proposed model are outlined, and a comparison of our experiments with other representative methods prove the superiority of our method.

1 Introduction

For traditional supervised classification problems, both positive and negative labels should be known before building a suitable binary classifier. However, in practical application, negative data labels are difficult to obtain, such as date sets, where only the relevant class is known, and the negative class is very large and dense [Kiryo *et al.*, 2017]. At this time, without the assistance of negative labels, analysis of positive and unlabeled (PU) data, which tries to learn a binary classifier by using only part of the labeled positive samples and other mixed samples, is used in practical applications. This special weakly supervised learning problem has been utilized in many real-world scenarios [Yu *et al.*, 2002; Li and Liu, 2003; Fang *et al.*, 2020a; Li *et al.*, 2010; Xu *et al.*, 2017; Wang *et al.*, 2018], such as web page classification, text classification, time series classification, multi-class classification, and remote-sensing images classification, etc.

For better understanding of the PU learning, we give one typical example, the personalized information pushing of webs. We may only know what the users' interests by their browsing records, but do not know what users' dislike. Therefore, to ensure matching between push information and users' demands, it is necessary to adopt an appropriate content filtering approach. Since only part of the preferences are provided, PU learning methods can push the text of users' needs while filtering out the irrelevant information.

Presently, mainstream PU data analysis methods can be classified into the following three categories. The first category is heuristic methods, which identify reliable negative examples from the unlabeled data and then train the classifier using the given positive samples as well as the learned reliable negative examples. The representative method of heuristic learning is the two-step approach, such as S-EM [Liu *et al.*, 2002], PEBL [Yu *et al.*, 2002], and Roc-SVM [Li and Liu, 2003]. The main disadvantage of most heuristic methods is that the classification accuracy is seriously affected by the selection of reliable negative examples. In the second category, all unlabeled data is treated as noisy negative examples, and the classifier is trained by introducing small weights to negative examples. This procedure makes unlabeled positive examples have a lower penalization, such that they are labeled as negatives. Lee and Liu [2003] explain that ideally, the known positive data is pure and reliable, and logistic regression is performed after weighting the noisy samples. In reality, however the positive data may also mislabeled, and Liu and Dai proposed a biased SVM (B-SVM) [2003] method that directly uses the asymmetric cost formulation of the SVM algorithm. Since these approaches need a training process to estimate the "bias", a volatile classification will result when the training set is limited or the positive samples are unreliable. The third category associates the classifier with the risk of classification and transfers PU learning into a cost-sensitive learning problem [Gong *et al.*, 2019b]. For example, Plessis [2014; 2015] proposed a convex formulation for PU learning and utilizes several different loss functions to maintain unbiased solutions. Further to the achievement of superior computational and memory performance, Sansone etc. [2018] proposed a scalable PU learning algorithm that converts the unbiased PU model into a sequence of quadratic programming (QP) subproblems. These methods

[*] Corresponding author.

usually employ a reweighing process to calibrate the data distribution because of the absence of negative samples; therefore, knowledge of the positive class prior is crucial. Additionally, there are also some setting-free PU methods, such as the label disambiguation-based methods [Zhang *et al.*, 2019; Gong *et al.*, 2019a], which enlarge the margin of potential positive examples and negative ones, as well as some booststrap sampling-based methods that create ensemble models for PU learning [Claesen *et al.*, 2015; Yang *et al.*, 2017].

Although a variety of methods have been proposed, the attempt to correctly label a data point will still interpreted by the absence of negative class. Fortunately, this defect can be rectified by explicitly acquiring the class prior, which is utilized as incorporated information for modeling or as a preprocessing step to assign weights to unlabeled data. Since the class prior is always unknown, in practical applications, research based on class-prior estimation has been active in past decades [Elkan and Noto, 2008; Christoffel *et al.*, 2016; Bekker and Davis, 2018].

To overcome the drawbacks of the aforementioned methods and to expand on the mixture proportion estimation theory [Ramaswamy *et al.*, 2016], we follow the third category and formulate a convex risk minimization joint class-prior approximation model for PU data analysis. In this model, we utilize an unbiased estimation of the classification risk of PU data, using double hinge loss to train the classifier, and pursue the class prior in a kernel embedding space with a mixture proportion approximation term. In particular, a proportion regularization term is added to balance the decision boundary and the prior estimation, as well as improve the accuracy.

The proposed PU learning model, which is named as “CAPU” in short, is effectively solved by the gradient thresholding algorithm. The theoretical analysis proves that the convergence of our model and the experimental results on extensive datasets demonstrate the superiority of the proposed algorithm compared to other state-of-the-art methods.

2 Preliminaries of PU Learning

Assume the d dimensional pattern $x \in \mathbb{R}^d$ and its class label $y \in \{1, -1\}$ follows the class-probability density, with $p(x, y)$. For the given i.i.d. positive dataset X_P and unlabeled dataset X_U, we have:

$$\begin{aligned} X_P &:= \{x_i\}_{i=1}^{n_p} \sim p(x|y=1), \\ X_U &:= \{x_i\}_{i=1}^{n_u} \sim p(x) \\ &= \pi p(x|y=1) + (1-\pi)p(x|y=-1), \end{aligned}$$

where $p(x)$ is the marginal density, and $\pi := p(y=1)$ is the positive class-prior probability in unlabeled data.

The goal is to learn an arbitrary decision function $g : \mathbb{R}^d \to \mathbb{R}$ to binary classify the PU data and utilize $\ell : \mathbb{R} \to \mathbb{R}$ as the loss function to quantify the values of $yg(x)$. Then, the risks of classifier g under the loss function ℓ are:

$$\begin{aligned} \mathcal{R}_P(g) &:= \mathbb{E}_{x\sim p(x|y=1)}[\ell(g(x))], \\ \mathcal{R}_U(g) &:= \mathbb{E}_{x\sim p(x)}[\ell(g(x))], \\ \mathcal{R}_{U,P}(g) &:= \mathbb{E}_{x\sim p(x|y=1)}[\ell(g(x))], \\ \mathcal{R}_{U,N}(g) &:= \mathbb{E}_{x\sim p(x|y=-1)}[\ell(-g(x))], \end{aligned}$$

where $\mathcal{R}_P(g)$, $\mathcal{R}_{U,P}$ and $\mathcal{R}_{U,N}(g)$ denote the classification risks of the positive and unlabeled samples, and $\mathbb{E}_{x\sim p(x,y)}[\cdot]$ denotes the expectation of x over the probability distribution $p(x, y)$.

The binary classification risk is a sum of the weighted positive class loss and negative class loss; therefore, the risk of our PU problem can be expressed as:

$$\begin{aligned} \mathcal{R}(g) &= \mathbb{E}_{p(x,y)}[\ell(yg(x))] \\ &= \pi\mathcal{R}_{U,P}(g) + (1-\pi)\mathcal{R}_{U,N}(g). \end{aligned} \tag{1}$$

The first term in Eq. (1) is equal to the risk of the positive samples, which means that $\mathcal{R}_{U,P}(g) = \mathcal{R}_P(g)$. Moreover, since no negative samples are labeled, naively training a classifier using only positive and unlabeled data may cause a bias. To overcome this problem, du Plessis *et al.* [2014; 2015] devised a risk deformation formula that is equivalent to traditional supervised classification risk. The second term in Eq. (1) is replaced by

$$\begin{aligned} &(1-\pi)\mathcal{R}_{U,N}(g) = (1-\pi)\mathbb{E}_{x\sim p(x|y=-1)}[\ell(-g(x)] \\ &= \mathbb{E}_{x\sim p(x)}[\ell(-g(x))] - \pi\mathbb{E}_{x\sim p(x|y=1)}[\ell(-g(x)] \\ &= \mathcal{R}_U(-g) - \pi\mathcal{R}_P(-g). \end{aligned} \tag{2}$$

Introducing a convex surrogate loss $\tilde{\ell}(m) = \ell(m) + \ell(-m) = -m$, the risk is yielded to the following convex optimization problem and solved efficiently:

$$\mathcal{R}(g) = \pi\tilde{\mathcal{R}}_P(g) + \mathcal{R}_U(-g) = \pi\mathbb{E}_1[-g] + \mathcal{R}_U(-g). \tag{3}$$

3 Our Method

In this section, we first establish our CAPU model, and then list brief theoretical analysis and optimization process of our model.

3.1 Model

Giving a dataset $X = \{X_P; X_U\} = \{(x_1, y_1), ..., (x_{n_p}, y_{n_p}); (x_{n_p+1}, y_{n_p+1}), ..., (x_n, y_n)\}$ with n_p positive samples and n_u unlabeled samples, where $n = n_p + n_u$. In practice, we use a linear-in-parameter function to denote g:

$$g(x) = \alpha^{\mathrm{T}}\phi(x) + b, \tag{4}$$

where ϕ is the set of kernel basis function which will be discussed later in this section. Then the empirical risk $\hat{\mathcal{R}}$ of Eq. (3) using n samples is:

$$\begin{aligned} \hat{\mathcal{R}}(g) = &-\frac{\pi}{n_p}\sum_{i=1}^{n_p}\alpha^{\mathrm{T}}\phi(x_i) - \pi b \\ &+ \frac{1}{n_u}\sum_{i=1}^{n_u}\ell(-\alpha^{\mathrm{T}}\phi(x_i) - b) + \frac{\lambda}{2}\alpha^{\mathrm{T}}\alpha \end{aligned} \tag{5}$$

where λ is the regularization parameter, and α is the parameter vector applied to minimize the ℓ_2-regularized empirical risk. To solve this optimization problem, a convex loss function [Ye *et al.*, 2018], double hinge loss, is introduced to Eq. (5). The formulation of DH loss is:

$$\ell_{\mathrm{DH}}(z) = \max(-z, \max(0, \frac{1}{2} - \frac{1}{2}z))$$

The PU model with a convex loss can be easily solved using the quadratic programming method. These risk estimators are based on the assumption that the class-prior probability π is known. However, in real-word applications, π is difficult to obtain and must be learned from data, which makes the risk evaluation model unsolvable. Therefore, combining a mixture proportion estimator, we proposed a **C**lass-prior **A**pproximation model for **PU** learning (CAPU), this optimization function can simultaneously pursue the class prior and optimize the parameters of the classifier.

If the distribution of X_U obeys i.i.d and can be approximately represented by a linear combination of positive and negative distributions:

$$\mathcal{P}_U = \pi^\star \mathcal{P}_{U,P} + (1-\pi^\star)\mathcal{P}_{U,N}. \tag{6}$$

where $\mathcal{P}_U$, $\mathcal{P}_{U,P}$ and $\mathcal{P}_{U,N}$ are the distributions of X_U, $X_{U,P}$ and $X_{U,N}$ in feature space $\mathcal{X}$, respectively, and $\pi^\star$ represents the true value of π,

For general semi-supervised classification learning that has both positive and negative labels, the labeled samples should have similar distributions to the unlabeled samples: $X_P, X_{U,P} \sim p(x|y=1)$ and $X_N, X_{U,N} \sim p(x|y=-1)$, and the mixture proportion estimation (MPE) methods [Ramaswamy *et al.*, 2016; Yu *et al.*, 2018] can solve this problem. It assumes there exists a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, where $\phi : \mathcal{X} \rightarrow \mathcal{H}$ represents the kernel mapping $x \rightarrow k(x,\cdot)$. The class proportion π is solved by minimizing the distance between the mixture and the given samples in the reproducing kernel Hilbert space (RKHS):

$$d(\pi) = \inf_{\pi} \|\phi(\mathcal{P}_U) - \pi\phi(\mathcal{P}_P) - (1-\pi)\phi(\mathcal{P}_N)\| \tag{7}$$

Intuitively, this equation equals the maximum mean discrepancy (MMD), where $d(\pi)$ is a reconstruction from $\mathcal{P}_P$ and $\mathcal{P}_N$ to the mixture $\mathcal{P}_U$. In practical applications, it must be replaced by empirical function:

$$\begin{aligned}\hat{d}(\pi) &= \min_{\pi} \|\phi(\hat{\mathcal{P}}_U) - \pi\phi(\hat{\mathcal{P}}_P) - (1-\pi)\phi(\hat{\mathcal{P}}_N)\| \\ &= \min_{\pi} \|\frac{1}{n_u}\sum_{i=1}^{n_u}\phi(x_i) - \frac{\pi}{n_p}\sum_{i=1}^{n_p}\phi(x_i) \\ &\quad - \frac{1-\pi}{n_n}\sum_{i=1}^{n_n}\phi(x_i)\|.\end{aligned} \tag{8}$$

which is a convex quadratic programming problem and can be solved using standard procedures. However, for practical PU analysis, the MPE model usually fails to separate the unlabeled negative samples from the mixture, since only positive samples are known. The third term of Eq. (8) is not identifiable; therefore, the optimal solution $\tilde{\pi}$ tends to be 1, which will cause serious misclassification.

To better suit PU data, we rewrite Eq. (6) in the following form:

$$\mathcal{P}_{U,N} = \theta^\star \mathcal{P}_U + (1-\theta^\star)\mathcal{P}_{U,P},$$

where $\theta^\star = \frac{1}{1-\pi^\star}$. So for the deformation from θ to π, there is a probability weighted vector $\nu \subseteq \mathbb{R}^n$ given by $\nu = \{\nu_i \in [0,1), \sum_i \nu_i = 1\}$, which makes the following empirical squared maximum mean discrepancy (MMD) lower bounded:

$$\inf_{\theta} \|\frac{\theta}{n_u}\sum_{i=1}^{n_u}\phi(x_i) + \frac{1-\theta}{n_p}\sum_{i=1}^{n_p}\phi(x_i) - \sum_{i=1}^{n}\nu_i\phi(x_i)\|^2. \tag{9}$$

With a combination of the empirical risk and squared MMD, the objective function of our CAPU model is formulated using the same RBF kernel, $\langle\phi(x_i), \phi(x_j)\rangle = \exp(-\frac{\|\mathrm{x_i}-\mathrm{x_j}\|^2}{2\sigma^2})$. We can tell the coefficient ν_i corresponds to the positive class turn out to be smaller, and that corresponds to negative class are larger, which just opposite to $\alpha^{\mathrm{T}}\phi(x_i)$. To further makes our objective function more robust, we assume the separability vector ν and the parameter α satisfy $\nu \approx -\frac{1}{n}\phi(x)^{\mathrm{T}}\alpha$. Therefore, the objective function of our CAPU model $\hat{\mathcal{F}}$ with the selected DH loss is written as:

$$\begin{aligned}\hat{\mathcal{F}}_{DH}(\alpha, b, \theta, \nu) &= \frac{\lambda}{2}\alpha^{\mathrm{T}}\alpha - \frac{\theta-1}{\theta n_p}\sum_{i=1}^{n_p}\alpha^{\mathrm{T}}\phi(x_i) \\ &- \frac{\theta-1}{\theta}b + \frac{1}{n_u}\sum_{i=1}^{n_u}\ell_{DH}(-\alpha^{\mathrm{T}}\phi(x_i) - b) + \\ &\|\frac{1-\theta}{n_p}\sum_{i=1}^{n_p}\phi(x_i) + \frac{\theta}{n_u}\sum_{i=1}^{n_u}\phi(x_i) - \sum_{i=1}^{n}\nu_i\phi(x_i)\|^2 \\ &+ \beta\|\nu + \frac{1}{n}\phi(x)^{\mathrm{T}}\alpha\|^2.\end{aligned} \tag{10}$$

Compared to state-of-the-art method, the proposed model expects to simultaneously approximate the class prior of unlabeled samples and learn a fit classifier. In particular, we introduce a slack variable $\xi \in \mathbb{R}^{n_u}$ to transform the double hinge loss function, and utilize $\mu_\theta^{\mathrm{T}} = [\frac{1-\theta}{n_p}\mathbf{1}_{n_p}^{\mathrm{T}}, \frac{\theta}{n_u}\mathbf{1}_{n_u}^{\mathrm{T}}]$, where $\mathbf{1}_k$ is a k-dimensional all ones vector, to transform Eq. (10), whose equivalent dual problem can be expressed as:

$$\begin{aligned}\min_{\gamma,\delta,\theta,\nu}\ & \frac{1}{2\lambda}\gamma^{\mathrm{T}}UKU^{\mathrm{T}}\gamma - \frac{\theta-1}{n_p\lambda\theta}\tilde{\mathbf{1}}^{\mathrm{T}}KU^{\mathrm{T}}\gamma - \frac{1}{2}\delta^{\mathrm{T}}\mathbf{1}_{n_u} \\ & + (\mu_\theta - \nu)^{\mathrm{T}}K(\mu_\theta - \nu) + \beta(\nu + \frac{\theta-1}{nn_p\lambda\theta}K\tilde{\mathbf{1}} \\ & - \frac{1}{n\lambda}KU^{\mathrm{T}}\gamma)^{\mathrm{T}}(\nu + \frac{\theta-1}{nn_p\lambda\theta}K\tilde{\mathbf{1}} - \frac{1}{n\lambda}KU^{\mathrm{T}}\gamma) \\ \text{s.t.}\ & \gamma + \frac{1}{2}\delta \preceq \frac{1}{n_u}\mathbf{1}_{n_u},\ \mathbf{0}_{n_u} \preceq \gamma - \frac{1}{2}\delta \preceq \frac{1}{n_u}\mathbf{1}_{n_u}, \\ & \mathbf{0}_{n_u} \preceq \delta \preceq \frac{1}{n_u}\mathbf{1}_{n_u},\ \mathbf{1}_{n_u}^{\mathrm{T}}\gamma = \frac{\theta-1}{\theta}, \\ & \nu \succeq \mathbf{0}_n,\ \mathbf{1}_n^{\mathrm{T}}\nu = 1.\end{aligned} \tag{11}$$

Where $\gamma, \delta \in \mathbb{R}^n$ are the Lagrange multipliers introduced to derive the dual formulation, $K \in \mathbb{R}^{n\times n}$ is the Gram matrix given by $K_{i,j} = \langle\phi(x_i), \phi(x_j)\rangle$, $\tilde{\mathbf{1}} = [1, ..., 1, 0, ..., 0]^{\mathrm{T}} \in \mathbb{R}^{n\times 1}$ has n_p ones elements, and $U \in \mathbb{R}^{n_u\times n}$ is a concatenation of a $n_u \times n_p$ null matrix and a n_u-size identity matrix. $\succeq$ represents the operation elements-wise. The proposed CAPU model can be efficiently computed via solving a standard quadratic programming using the gradient thresholding estimator.

3.2 Theoretical Analysis

To solve the parameters of the proposed CAPU model, we utilize the gradient thresholding estimator which first estimate $\tilde{\pi}$ (or equivalently $\tilde{\theta}$) and then using standard quadratic programming method to find $(\tilde{\gamma}, \tilde{\delta}, \tilde{\nu})$. $\tilde{(\cdot)}$ represents the optimal value of the current variable. Before solving this model, some theoretical analysis is listed for the proposed model.

First, our objective function can be viewed as a function related to parameter θ:

$$\begin{aligned}\mathcal{F}(\theta) = &- \frac{1}{\theta}\mathbb{E}_{x\sim(x|y=1)}[-g(x)] \\ &+ \|(1-\theta)\phi(\mathcal{P}_P) + \theta\phi(\mathcal{P}_U) - \nu\phi(\mathcal{P})\|^2 + \mathcal{C}\end{aligned}$$

Then the following Lemma holds,

Lemma 1. *Let the kernel k be such that $k(x,x) \leq 1$ for all $x \in \mathcal{X}$. Then with probability at least $1-\delta$, the following holds if $n_u \geq 2(\theta^\star)^2 \log(\frac{5}{\delta})$*

$$\mathbb{E}_1[-g] - \frac{1}{n_p}\sum_{i=1}^{n_p}(-g) \leq -2\sqrt{nC} + \frac{\sqrt{\log(\frac{5}{\delta})}}{\sqrt{2n_p}}$$

$$\|\phi(\mathcal{P}_U) - \phi(\hat{\mathcal{P}}_U)\| \leq \frac{2}{\sqrt{n_u}} + \frac{\sqrt{\log(\frac{5}{\delta})}}{\sqrt{n_u}}$$

$$\|\phi(\mathcal{P}_P) - \phi(\hat{\mathcal{P}}_P)\| \leq \frac{2}{\sqrt{n_p}} + \frac{\sqrt{\log(\frac{5}{\delta})}}{\sqrt{n_p}}$$

$$\|\phi(\mathcal{P}_N) - \phi(\hat{\mathcal{P}}_N)\| \leq \frac{2}{\sqrt{n_u/2\theta^\star}} + \frac{\sqrt{\log(\frac{5}{\delta})}}{\sqrt{n_u/2\theta^\star}}$$

where $C = \frac{2}{\lambda}|b| + \frac{1}{\lambda} + \frac{2}{\lambda}(\mu_\theta - \nu)^{\mathrm{T}}K(\mu_\theta - \nu) + \frac{2}{\lambda}\nu^{\mathrm{T}}\nu$. The first statement can be derived by deducing Rademacher complexity and the approximation of Eq. (10). The second and third statements are learned from Theorem 2 of Smola *et al.* [2007] and the last statement utilizes Hoeffding's inequality. Lemma 1 proves that $\hat{\mathcal{F}}(\theta)$ is upper bounded for $\theta \in [1, \theta^\star]$.

Lemma 2. *For all $\theta \in [1, \theta^\star]$ we have:*

$$\begin{aligned}\hat{\mathcal{F}}(\theta) \leq &\ (2\theta^2 - 2\theta + \frac{2\theta^2 - 2\theta}{\theta^\star} + \frac{\theta^2}{\theta^{\star 2}} + 2)\frac{9\log(\frac{5}{\delta})}{\min(n_p, n_u)} \\ &+ \frac{\sqrt{nC}}{\theta} - \frac{\sqrt{\log(\frac{5}{\delta})}}{\theta\sqrt{2n_p}},\end{aligned}$$

$$\begin{aligned}\hat{\mathcal{F}}(\theta) \geq &\ \mathcal{F}(\theta) - (2\theta^2 - 2\theta + 1)\frac{9\log(\frac{5}{\delta})}{\min(n_p, n_u)} \\ &- \frac{2\sqrt{nC}}{\theta} + \frac{\sqrt{\log(\frac{5}{\delta})}}{\theta\sqrt{2n_p}}).\end{aligned}$$

Theorem 1. *Assume the kernel k, the distributions $\mathcal{X}_P$ and $\mathcal{X}_N$ satisfy the separability condition with average level τ, the margin is $c > 0$, and the tolerance is d, then for arbitrary $\Delta > 0$, we have*

$$\mathcal{F}(\theta^\star + \Delta) \geq (c + \frac{\Delta}{\theta^\star}d)^2 - \frac{n(\tau - d)}{\theta^\star + \Delta} + \frac{b}{\theta^\star + \Delta}$$

Algorithm 1 The optimization process of the proposed model

Input: The given positive samples $x_P := \{x_i\}_{i=1}^{n_p}$ and the unlabeled samples $x_U := \{x_i\}_{i=1}^{n_u}$
Parameter: The width of Gaussian kernel σ, hyperparameters λ and β and threshold $\mu = 1/\min(n_p, n_u)$
Output: $\tilde{\theta}$,$\tilde{\gamma}$,$\tilde{\delta}$,$\tilde{\nu}$
Constants: $\epsilon = 0.04$, $\theta_{\max} = 10$

```
Let $\theta_l = 1$, $\theta_r = \theta_{\max}$
while $\theta_r - \theta_l \geq \epsilon$ do
    $\theta = \frac{\theta_l + \theta_r}{2}$
    $\theta_1 = \theta - \epsilon/4$
    $\mu_{\theta 1}^{\mathrm{T}} = [\frac{1-\theta_1}{n_p}\mathbf{1}_{n_p}^{\mathrm{T}}, \frac{\theta_1}{n_u}\mathbf{1}_{n_u}^{\mathrm{T}}]$
    $\mathcal{F}_1 = \hat{\mathcal{F}}(\theta_1)$
    $\theta_2 = \theta + \epsilon/4$
    $\mu_{\theta 2}^{\mathrm{T}} = [\frac{1-\theta_2}{n_p}\mathbf{1}_{n_p}^{\mathrm{T}}, \frac{\theta_2}{n_u}\mathbf{1}_{n_u}^{\mathrm{T}}]$
    $\mathcal{F}_2 = \hat{\mathcal{F}}(\theta_2)$
    $s = \frac{\sqrt{\mathcal{F}_2} - \sqrt{\mathcal{F}_1}}{\theta_2 - \theta_1}$
    if $s > \mu$ then
        $\theta_2 = \theta$.
    else
        $\theta_1 = \theta$.
    end if
end while
return $\theta \to \tilde{\theta}$ and $\tilde{\pi} = 1 - 1/\tilde{\theta}$
Optimize the objective function $\min_{\gamma,\delta,\nu} \hat{\mathcal{F}}(\tilde{\theta})$
return $\tilde{\gamma}, \tilde{\delta}, \tilde{\nu}$
```

Through the above analysis, we can finally derive that the slope of function $\mathcal{F}(\theta)$ satisfies:

Lemma 3. *Let the kernel k be such that $k(x,x) \leq 1$ for all $x \in \mathcal{X}$. Assume the kernel k, the distributions $\mathcal{P}_P$ and $\mathcal{P}_N$ satisfy the separability condition with average level τ, the margin is $c > 0$ and the tolerance is d. Then the gradient of $\hat{\mathcal{F}}(\theta)$ at some $\tilde{\theta}_\kappa < \theta^\star$ is also upper and lower bounded.*

Proof. (Sketch) This lemma can be derived from Lemma 2 and Theorem 1. Considering that $\hat{\mathcal{F}}(\theta)$ is a convex function of θ, the gradient of $\hat{\mathcal{F}}(\theta)$ at some $\tilde{\theta} < \theta^\star$ is also upper and lower bounded. □

3.3 Optimization

Since there is a reciprocal form of θ in our objective function, the general optimization method is difficult to apply. Through above analysis, we derived that our objective function $\mathcal{F}(\theta)$ is upper bounded and converges to $\theta^{\star 2}$ at a rate of $\mathcal{O}(n_p^{-1})$. Therefore, the gradient thresholding algorithm via binary search is designed to solve the proposed model. Detailed descriptions of the computation are given in Algorithm 1, where $\hat{\mathcal{F}}(\theta)$ is the first objective function and the setting of threshold μ is followed by Lemma 3.

This gradient thresholding estimator first gives the upper and lower bounds (θ_l and θ_r) to constrain the value of θ, and it conducts the binary search by minimizing the value of $\hat{\mathcal{F}}(\theta)$

at $\theta \pm \epsilon/4$, respectively. The current bounds of θ and the termination of Algorithm 1 is determined by the computed slope. The minimization of $\hat{\mathcal{F}}_\theta$ is a standard quadratic programming problem and can be optimized by the general purpose convex programming solver, such as CVXOPT, Gurobi or MATLAB's internal 'quadprog' function.

4 Experimental Procedure

In this section, we systematically evaluate the effectiveness of the proposed CAPU method compared with other state-of-the art PU methods in a synthetic dataset and real-world datasets taken from UCI Machine Learning Repository.

Synthetic Dataset. The parametric analysis and evaluations were implemented in the synthetic dataset. This data comprises two clusters generated from Gaussian distributions centered at $(0, 0)$ and $(2, 2)$, and the variance of both clusters is 1. In total, it contains 800 samples whereas 200 examples are labeled as positives. The size of the unlabeled samples is fixed at 600. Then, the proposed CAPU model was conducted with a class prior varies from [0.3, 0.5, 0.7].

Real-world Datasets. We utilize four real-world datasets downloaded from the UCI Machine Learning Repository to evaluate the performance of our proposed algorithm. These datasets include the *audit*, *ionosphere*, *diabetes* and *vertebral*, and their configurations are listed in Table 2. Based on the sizes of the different datasets, we set the number of positive samples as 100 and the unlabeled samples as 400 for the *audit* dataset. The sizes of n_p and n_u are 50 and 200, respectively, for the *diabetes* dataset. And for the *ionosphere* and *vertebral* datasets, we set $n_p = 25$ and $n_u = 100$. For each dataset, $\pi \in [30\%, 50\%, 70\%]$ samples of the unlabel to be positive and the rest samples are negative.

Comparable Methods. The proposed CAPU method is compared with related works such as EN[1] [Elkan and Noto, 2008], PE[2] [Du Plessis and Sugiyama, 2014], KM[3] [Ramaswamy *et al.*, 2016], and TIcE[4] [Bekker and Davis, 2018]. The aforementioned methods have made great efforts in estimating the true class prior. Considering that the KM and TIcE do not provide a classification process, we report the accuracies utilizing a benchmark PU learning method: the unbiased PU (UPU)[5] [Du Plessis *et al.*, 2015]. For further accuracy comparison, the UPU [Du Plessis *et al.*, 2015] and an improved method, USMO[6] [Sansone *et al.*, 2018], are also tested with the truth vale of π.

Method \ π	0.3	0.5	0.7
EN	0.588(0.288) 63.45±20.12✓	0.646(0.146) 68.34±25.90✓	0.690(0.010) 73.03±17.50✓
PE	0.403(0.103) 53.05±4.68✓	0.593(0.007) 55.59±4.14✓	0.774(0.074) 73.51±3.83✓
KM+UPU	**0.299(0.001)** 61.27±8.77✓	0.505(0.005) 54.05±33.05✓	**0.702(0.002)** 81.99±2.44 ✓
TIcE+UPU	0.505(0.205) 59.61±3.86✓	0.755(0.255) 66.19±3.25✓	0.903(0.203) 82.35±0✓
UPU	- 60.13±6.34✓	- 65.27±1.32✓	- 82.35±0✓
USMO	- 83.64±6.58✓	- 92.26±2.00	- 84.73±6.10✓
CAPU	0.379(0.079) **85.38±6.27**	**0.503(0.003)** **92.33±2.35**	0.715(0.015) **90.48±1.28**

Table 1: The comparative results of the various methods on the syntectic dataset when the class prior is set as $30\%, 50\%$ and 70% of the unlabeled data. The estimates/the absolute class prior error and the F-scores (%) over 20 trials are reported. The best record under each π is marked in **bold**. "✓" indicates that the proposed method is significantly better than the corresponding method via paired t-test.

4.1 Results

To create PU samples from each dataset, we derived three different settings of positive and unlabeled samples as follows. We first set a fraction of the positives as the labeled samples, then we select part of the remaining positive and negative instances as the unlabeled set. The class-prior in the unlabeled set varies in $[0.3, 0.5, 0.7]$. This procedure was repeated 20 times for each setting for each dataset, and the evaluation matrices applied for performance comparisons are the mean class-prior estimates $\tilde{\pi}$, the mean absolute errors $|\tilde{\pi} - \pi|$ and the F-scores [Fang *et al.*, 2020b] over 20 trials.

The results of all methods on the synthetic dataset are reported in Table 1. In this two-cluster Gaussian distributed dataset, KM has the nearest class-prior estimation when $\pi = 0.3$ and 0.5, and the proposed CAPU method generally has secondary performance. When applying the classification evaluation, our method achieves the best classification accuracy compared with other methods.

Table 2 illustrates the comparative result of the proposed CAPU method and other methods on four UCI datasets. We found that in most occasions, our method have the superior class-prior approximation, while in few cases, the KM and TIcE methods are better than CAPU. But for the classification accuracies, our model achieves the best in most occasions even the true class prior is given for USMO and UPU methods.

4.2 Parametric Analysis

There are three parameters included in our CAPU model: the width σ of the RBF kernel, and the trade off parameters λ and β. Therefore, this section examines the parametric sensitivity of our model on the synthetic dataset.

Figure 1 shows the mean absolute error $|\tilde{\pi} - \pi|$ and the F-scores related to σ on the three settings of the synthetic dataset. It is known that the smaller values of the error represents the better estimates, and the higher F-scores mean better classification accuracy. It is shown that in all three settings, the performance of our CAPU model is best when the kernel width $\sigma = 1$. To evaluate the accuracy associated with hy-

[1] The coding work of EN method can be found at https://github.com/aldro61/pu-learning

[2] The code for PE is taken from http://www.mcduplessis.com/index.php/class-prior-estimation-from-positive-and-unlabeled-data/

[3] The code for KM method is taken from http://web.eecs.umich.edu/~cscott/code/kernel_MPE.zip

[4] The code for TIcE method can be found at https://dtai.cs.kuleuven.be/software/tice

[5] The coding work of UPU method can be found at https://github.com/kiryor/nnPUlearning

[6] The code for USMO method is available at https://github.com/emsansone/USMO

Dataset	(N, d)	π	EN	PE	KM+UPU	TICE+UPU	UPU	USMO	CAPU
		0.3	0.638(0.338)	0.247(0.053)	0.171(0.129)	0.353(0.053)	-	-	**0.262(0.038)**
			60.96±2.34 ✓	63.76±3.50 ✓	83.74±5.12 ✓	87.32±4.99 ✓	91.05±1.34 ✓	82.35±0 ✓	**93.04±1.34**
audit	(774,17)	0.5	0.863(0.363)	0.423(0.077)	0.305(0.295)	0.464(0.036)	-	-	**0.520(0.020)**
			66.52±0.32 ✓	49.07±2.46 ✓	72.57±9.00 ✓	88.51±5.14 ✓	91.93±1.08 ✓	92.05±2.71 ✓	**96.38±2.29**
		0.7	0.899(0.199)	0.587(0.113)	0.511(0.189)	0.537(0.163)	-	-	**0.724(0.024)**
			82.34±0.21 ✓	68.80±2.63 ✓	81.27±6.14 ✓	82.58±6.14 ✓	90.56±1.34 ✓	84.66±7.90 ✓	**98.21±1.54**
		0.3	0.923(0.623)	0.260(0.040)	0.276(0.024)	0.239(0.061)	-	-	**0.280(0.020)**
			46.15±0 ✓	41.46±11.9 ✓	55.00±6.22 ✓	56.23±3.42 ✓	56.06±2.98 ✓	**78.18±5.23**	74.03±7.07
ionosphere	(351,33)	0.5	0.977(0.477)	0.350(0.150)	0.480(0.020)	0.321(0.079)	-	-	**0.506(0.006)**
			66.67±0 ✓	54.86±7.87 ✓	63.75±7.07 ✓	65.81±5.45 ✓	62.63±8.00 ✓	68.27±1.63 ✓	**71.21±10.47**
		0.7	1(0.3)	0.461(0.239)	0.411(0.289)	0.837(0.137)	-	-	**0.800(0.100)**
			62.35±0 ✓	69.54±8.80 ✓	69.57±22.94 ✓	69.54±8.80 ✓	59.20±2.00 ✓	75.75±7.66	**76.12±12.94**
		0.3	1(0.7)	0.556(0.256)	0.454(0.154)	0.597(0.297)	-	-	**0.326(0.026)**
			46.15±0 ✓	41.49±4.08 ✓	47.95±12.0 ✓	51.71±6.84 ✓	44.76±9.85 ✓	57.43±4.62 ✓	**65.38±7.90**
diabetes	(768,8)	0.5	1(0.5)	0.680(0.180)	0.339(0.161)	**0.626(0.126)**	-	-	0.753(0.253)
			66.67±0 ✓	60.64±6.23 ✓	61.15±8.86 ✓	64.41±4.37 ✓	61.33±6.50 ✓	70.68±8.80	**72.41±7.40**
		0.7	1(0.3)	0.748(0.048)	0.443(0.257)	0.537(0.163)	-	-	**0.741(0.041)**
			62.34±0 ✓	76.62±7.92	69.72±10.8 ✓	72.68±9.13 ✓	74.19±3.93	71.35±6.78 ✓	**77.90±8.16**
		0.3	0.758(0.458)	0.567(0.267)	0.410(0.110)	**0.237(0.063)**	-	-	0.627(0.327)
			44.38±26.36 ✓	42.12±11.52 ✓	54.09±11.38 ✓	54.64±9.92 ✓	53.57±8.33 ✓	67.37±10.53 ✓	**70.31±12.79**
vertebral	(310,6)	0.5	0.828(0.328)	0.671(0.171)	**0.505(0.005)**	0.597(0.097)	-	-	0.740(0.240)
			67.31±3.57 ✓	60.99±7.60 ✓	57.97±34.78 ✓	64.42±4.73 ✓	61.39±6.00 ✓	73.85±13.88	**74.65±10.53**
		0.7	0.884(0.184)	0.818(0.118)	0.306(0.394)	0.661(0.039)	-	-	**0.733(0.033)**
			72.26±0.49 ✓	78.97±6.33 ✓	67.14±11.49 ✓	73.24±8.64 ✓	80.56±1.34	68.79±11.67 ✓	**81.90±10.17**

Table 2: The comparative results of various methods on real-world datasets when the class prior is set as $30\%, 50\%$ and 70% of the unlabeled data. The estimates/the absolute class-prior error and the F-scores $(\%)$ over 20 trials are reported. The best record under each π is marked in **bold**. "✓" indicates that the proposed method is significantly better than the corresponding method via paired t-test.

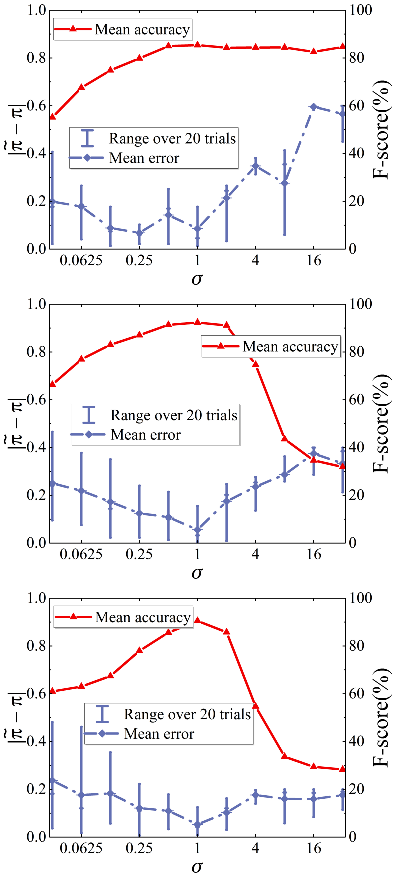

Figure 1: The Gaussian kernel width evaluations of the proposed CAPU model on the synthetic dataset with $\pi = 0.3, 0.5, 0.7$ over 20 trials. The left axis reports the absolute class-prior error $|\tilde{\pi} - \pi|$ and the right axis reports the F-score $(\%)$ measurement.

perparameters λ and β, Figure 2 shows a 3-D cube where the z-axis show the mean F-score values of CAPU over 20 times' repeated tests. It can be seen that the proposed CAPU model is not much sensitive to β, but to λ.

4.3 Discussion

Through out previous analysis, we found that when estimating the class prior, our method has similar performances to other methods. But for classification accuracy, our method gets the best F-score in most cases. In this section, we discuss the reasons for this result.

Firstly, traditional PU learning models based on the risk minimization treat misclassification errors as proportions of labeled positives and unlabeled negatives, which determines that it is always crucial to know the class prior, such as in UPU and USMO methods. However, the necessary knowledge of π is obviously inconsistent with reality. Therefore, some research based on class-prior estimation has been proposed, such as the KM and TIcE methods. These methods provide additional information for later analysis, but they only provide an estimation of the positive proportions. In addition, there are also some methods based on class-prior correction and classification, such as the benchmark EN and PE methods, which can estimate class prior as well as classify data. We found that these methods have unstable performance with huge estimation errors, which then easily leads to poor classification results.

Compared with the aforementioned research, our method is undoubtedly more suitable for PU data analysis. This model overcomes the lack of class-prior information for the general PU method, and it can classify the dataset with superior accuracy compared to other methods.

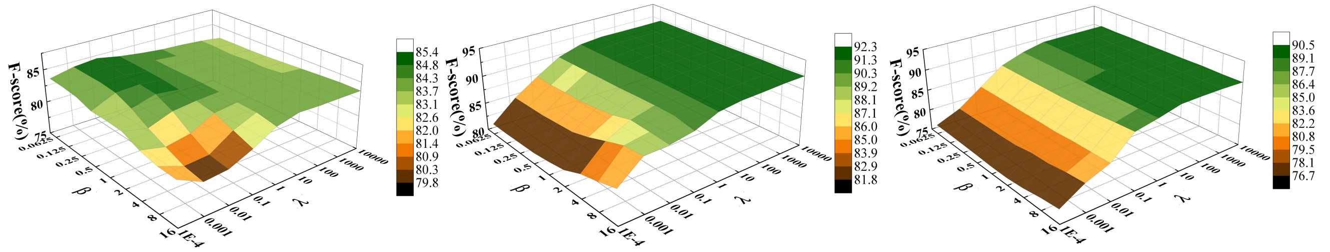

Figure 2: The hyperparameters λ and β evaluations of the proposed CAPU model on the synthetic dataset with $\pi = 0.3, 0.5, 0.7$ over 20 trials.

5 Conclusion

This paper proposed a novel PU learning method with class prior approximation. Different from previous analyses, we convert PU learning to a direct class-prior estimation and classification problem by introducing the mixture proportion estimation to the loss minimization function. Furthermore, an additional regularization term balances the classifying vector and the proportion estimation result. A gradient thresholding algorithm is utilized based on rigorous theoretical analysis. Experimental results on both synthetic and real-world datasets clearly show that CAPU is superior to other state-of-the-art methods.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61822113, 62041105, the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170, the Natural Science Foundation of Hubei Province under Grants 2018CFA050, the Fundamental Research Funds for the Central Universities under Grant 413000092 and 413000082. And the numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

References

[Bekker and Davis, 2018] Jessa Bekker and Jesse Davis. Estimating the class prior in positive and unlabeled data through decision tree induction. In *AAAI*, 2018.

[Christoffel *et al.*, 2016] Marthinus Christoffel, Gang Niu, and Masashi Sugiyama. Class-prior estimation for learning from positive and unlabeled data. In *ACML*, pages 221–236, 2016.

[Claesen *et al.*, 2015] Marc Claesen, Frank De Smet, Johan AK Suykens, and Bart De Moor. A robust ensemble approach to learn from positive and unlabeled data using svm base models. *Neurocomputing*, 160:73–84, 2015.

[Du Plessis and Sugiyama, 2014] Marthinus Christoffel Du Plessis and Masashi Sugiyama. Class prior estimation from positive and unlabeled data. *IEICE Transactions on Information and Systems*, 97(5):1358–1362, 2014.

[Du Plessis *et al.*, 2014] Marthinus C 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 Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *ICML*, pages 1386–1394, 2015.

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

[Fang *et al.*, 2020a] Yixiang Fang, Xin Huang, Lu Qin, Ying Zhang, Wenjie Zhang, Reynold Cheng, and Xuemin Lin. A survey of community search over big graphs. *The VLDB Journal*, 29(1):353–392, 2020.

[Fang *et al.*, 2020b] Yixiang Fang, Yixing Yang, Wenjie Zhang, Xuemin Lin, and Xin Cao. Effective and efficient community search over large heterogeneous information networks. *VLDB Endowment*, 13(6):854–867, 2020.

[Gong *et al.*, 2019a] Chen Gong, Tongliang Liu, Jian Yang, and Dacheng Tao. Large-margin label-calibrated support vector machines for positive and unlabeled learning. *IEEE T-NNLS*, 2019.

[Gong *et al.*, 2019b] Chen Gong, Hong Shi, Tongliang Liu, Chuang Zhang, Jian Yang, and Dacheng Tao. Loss decomposition and centroid estimation for positive and unlabeled learning. *IEEE T-PAMI*, 2019.

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

[Lee and Liu, 2003] Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *ICML*, volume 3, pages 448–455, 2003.

[Li and Liu, 2003] Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *IJCAI*, volume 3, pages 587–592, 2003.

[Li *et al.*, 2010] Wenkai Li, Qinghua Guo, and Charles Elkan. A positive and unlabeled learning algorithm for one-class classification of remote-sensing data. *IEEE T-GRS*, 49(2):717–725, 2010.

[Liu *et al.*, 2002] Bing Liu, Wee Sun Lee, Philip S Yu, and Xiaoli Li. Partially supervised classification of text documents. In *ICML*, volume 2, pages 387–394, 2002.

[Liu *et al.*, 2003] Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and S Yu Philip. Building text classifiers using positive and unlabeled examples. In *ICDM*, volume 3, pages 179–188, 2003.

[Ramaswamy *et al.*, 2016] Harish Ramaswamy, Clayton Scott, and Ambuj Tewari. Mixture proportion estimation via kernel embeddings of distributions. In *ICML*, pages 2052–2060, 2016.

[Sansone *et al.*, 2018] Emanuele Sansone, Francesco GB De Natale, and Zhi-Hua Zhou. Efficient training for positive unlabeled learning. *IEEE T-PAMI*, 2018.

[Smola *et al.*, 2007] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory*, pages 13–31, 2007.

[Wang *et al.*, 2018] Zheng Wang, Xiang Bai, Mang Ye, and Shin'ichi Satoh. Incremental deep hidden attribute learning. In *ACM Multimedia*, pages 72–80, 2018.

[Xu *et al.*, 2017] Yixing Xu, Chang Xu, Chao Xu, and Dacheng Tao. Multi-positive and unlabeled learning. In *IJCAI*, pages 3182–3188, 2017.

[Yang *et al.*, 2017] Pengyi Yang, Wei Liu, and Jean Yang. Positive unlabeled learning via wrapper-based adaptive sampling. In *IJCAI*, pages 3273–3279, 2017.

[Ye *et al.*, 2018] Mang Ye, Zheng Wang, Xiangyuan Lan, and Pong C Yuen. Visible thermal person re-identification via dual-constrained top-ranking. In *IJCAI*, volume 1, page 2, 2018.

[Yu *et al.*, 2002] Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. Pebl: positive example based learning for web page classification using svm. In *Proc. SIGKDD*, pages 239–248, 2002.

[Yu *et al.*, 2018] Xiyu Yu, Tongliang Liu, Mingming Gong, Kayhan Batmanghelich, and Dacheng Tao. An efficient and provable approach for mixture proportion estimation using linear independence assumption. In *CVPR*, pages 4480–4489, 2018.

[Zhang *et al.*, 2019] Chuang Zhang, Dexin Ren, Tongliang Liu, Jian Yang, and Chen Gong. Positive and unlabeled learning with label disambiguation. In *IJCAI*, pages 4250–4256, 2019.