Positive and Unlabeled Learning with Label Disambiguation

Chuang Zhang[1] , **Dexin Ren**[1] , **Tongliang Liu**[3] , **Jian Yang**[1,2] and **Chen Gong**[1]
[1]PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2]Jiangsu Key Lab of Image and Video Understanding for Social Security
[3]UBTECH Sydney AI Centre, SCS, FEIT, The University of Sydney, Australia
chen.gong@njust.edu.cn

Abstract

Positive and Unlabeled (PU) learning aims to learn a binary classifier from only positive and unlabeled training data. The state-of-the-art methods usually formulate PU learning as a cost-sensitive learning problem, in which every unlabeled example is simultaneously treated as positive and negative with different class weights. However, the ground-truth label of an unlabeled example should be unique, so the existing models inadvertently introduce the label noise which may lead to the biased classifier and deteriorated performance. To solve this problem, this paper proposes a novel algorithm dubbed as "**P**ositive and **U**nlabeled learning with **L**abel **D**isambiguation" (PULD). We first regard all the unlabeled examples in PU learning as ambiguously labeled as positive and negative, and then employ the margin-based label disambiguation strategy, which enlarges the margin of classifier response between the most likely label and the less likely one, to find the unique ground-truth label of each unlabeled example. Theoretically, we derive the generalization error bound of the proposed method by analyzing its Rademacher complexity. Experimentally, we conduct intensive experiments on both benchmark and real-world datasets, and the results clearly demonstrate the superiority of the proposed PULD to the existing PU learning approaches.

1 Introduction

Collecting a large amount of labeled data, especially labeled negative data, is a critical bottleneck in many real-world machine learning applications due to the laborious manual annotation or the difficulty of collecting negative data. In contrast, positive and unlabeled data can often be collected easily. This has led to the development of *Positive and Unlabeled* (PU) learning [Denis *et al.*, 2005], which aims at learning a binary classifier only from positive and unlabeled data without the assistance of negative data. Recently, PU learning has gained lots of popularity in tackling many real-world scenarios such as software clone detection [Wei and Li, 2018], protein function prediction [Youngs *et al.*, 2014; Fu *et al.*, 2016], and remote-sensed hyperspectral image classification [Li *et al.*, 2011], etc.

Given its broad applicability as mentioned above, PU learning has attracted a great deal of research attention in recent years. Effective algorithms have been developed which can be roughly divided into three categories. The first category [Liu *et al.*, 2002; Li and Liu, 2003] firstly identifies some reliable negative examples from the unlabeled set, and then employs the reliable negative as well as the original positive examples to train a traditional supervised classifier. The second category [Lee and Liu, 2003; Shi *et al.*, 2018] takes PU leaning problem as a one-side label noise learning problem, which directly treats the unlabeled examples as negative ones. Particularly, the undiscovered positive examples in the unlabeled set are regarded as mislabeled, while no negative examples are mislabeled as positive. The last category [Elkan and Noto, 2008; Youngs *et al.*, 2015; Du Plessis *et al.*, 2014; Du Plessis *et al.*, 2015] imposes different weights on the loss values incurred by simultaneously treating unlabeled examples as positive and negative, and thus transferring PU learning into a cost-sensitive learning problem. However, the ground-truth label of each unlabeled example is definite and unique, so simultaneously evaluating every unlabeled example as positive and negative under the framework of empirical risk minimization will inevitably introduce the label noise, leading to the imperfect performance.

To address this drawback, this paper proposes a novel PU leaning algorithm dubbed "**P**ositive and **U**nlabeled learning with **L**abel **D**isambiguation" (PULD). To be specific, we first treat PU learning problem as a *Partial Label Learning* (PLL) [Jin and Ghahramani, 2003] problem, where we regard the unlabeled examples as ambiguously labeled as positive and negative, and then employ the disambiguation technique, which is widely used in PLL, to robustly determine the real label of every unlabeled example. PLL aims to learn from ambiguous labeling information where each training example is associated with a set of candidate labels, among which only one label is valid [Cour *et al.*, 2011; Gong *et al.*, 2018; Feng and An, 2018; Chen *et al.*, 2018]. Recent successful PLL methods have devised various disambiguation regularizers to identify the single ground-truth label from the candidate label set associated with each training example [Nguyen and Caruana, 2008; Wu and Zhang, 2018], which play an important role in boosting the PLL performance.

In our PU learning case, we take each unlabeled example as a partially labeled example with the candidate label set $\{1, 2\}$, and then utilize the margin based disambiguation strategy to enlarge the margin between the most likely label and the less likely one. As a result, the ground-truth label in the candidate label set can be effectively highlighted. Besides, a manifold regularizer with geometric knowledge of the original data is applied [Zhou *et al.*, 2004; Belkin *et al.*, 2006; Gong *et al.*, 2019b], so that the similar examples in the feature space are assigned similar labels. Furthermore, we theoretically derive the generalization error bound of the proposed PULD by analyzing its Rademacher complexity, and intensive experiments on both benchmark and real-world datasets clearly demonstrate the superiority of the proposed method to the state-of-the-art PU learning approaches.

2 Our Method

In this section, we first establish our proposed PULD model, and then explain its iterative optimization process.

2.1 Model

Assume that there are n training examples $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_p, y_p), (\mathbf{x}_{p+1}, y_{p+1}), \ldots, (\mathbf{x}_n, y_n); n = p + u\}$ identically and independently drawn from some distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \in \mathbb{R}^d$ and $\mathcal{Y} = \{1, 2\}$ denote the feature space and label space, respectively. For the convenience of the subsequent derivations, we follow the setting of multi-class classification, where negative examples are labeled as 1 and positive examples are labeled as 2. In $\mathcal{S}$, the first p elements with the label $\{y_i\}_{i=1}^{p} = 2$ constitute the positive set $\mathcal{P}$, and the rest u elements form the unlabeled set $\mathcal{U}$ in which the label of each example can be either 1 or 2 but not known.

In the training stage, our target is to find a real-valued scoring function $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ based on the training set $\mathcal{S} = \mathcal{P} \cup \mathcal{U}$, and the classification is performed according to the score, i.e. $\arg\max_{y \in \mathcal{Y}} h(\mathbf{x}, y)$. Firstly, we build a K-NN graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ over the training set $\mathcal{S}$ where $\mathcal{V}$ is the vertex set consisted of all $\mathbf{x}_i \in \mathcal{S}$, and $\mathcal{E}$ is the edge set describing the similarity between these nodes. Further, $\mathbf{A}$ is the adjacency matrix of $\mathcal{G}$, and its (i, j)-th element $\mathbf{A}_{ij}$ encodes the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, which is computed by $\mathbf{A}_{ij} = \exp\big(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\delta^2)\big)$ with δ being the Gaussian kernel width if $\mathbf{x}_i$ and $\mathbf{x}_j$ are linked, and $\mathbf{A}_{ij} = 0$ otherwise. Therefore, the normalized adjacency matrix adopted by us is $\bar{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ with $\mathbf{D}$ being the diagonal degree matrix of which the (i, i)-th element is $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. Moreover, we define the mapping $\Phi(\mathbf{x}, y) : \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{F}$ which projects the example-label pair $(\mathbf{x}, y)$ to $\Phi(\mathbf{x}, y) \in \mathcal{F}$, namely

$$\Phi(\mathbf{x}, y) = \begin{pmatrix} \mathbf{x} \cdot \mathbb{1}(y = 1) \\ \mathbf{x} \cdot \mathbb{1}(y = 2) \end{pmatrix} \in \mathbb{R}^{2d}, \tag{1}$$

with d being the dimensionality of the input feature $\mathbf{x}$, and $\mathbb{1}(a)$ being an indicator function which returns 1 if a is true and 0 otherwise. Let $\mathbf{y} = (y_1, \ldots, y_p, y_{p+1}, \ldots, y_n)$ be the ground-truth labels of training examples. Then given the classifier as $h(\mathbf{x}, y) = \boldsymbol{\omega}^\top \Phi(\mathbf{x}, y)$, and $\boldsymbol{\xi} = \{\xi_i\}_{i=1}^{p}$ and $\boldsymbol{\eta} = \{\eta_i\}_{i=p+1}^{p+u}$ as slack variables, our model is formulated as

$$\min_{\boldsymbol{\omega}, \mathbf{y}, \boldsymbol{\xi}, \boldsymbol{\eta}} \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{\alpha}{p}\sum_{i=1}^{p}\xi_i + \frac{\beta}{u}\sum_{i=p+1}^{p+u}\eta_i + \gamma\sum_{i,j=1}^{n}\bar{\mathbf{A}}_{ij}(y_i - y_j)^2 \tag{2}$$

$$s.t. \quad \text{For } \mathbf{x}_i \in \mathcal{P}:$$

$$\boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, 2) - \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, 1) \geq 1 - \xi_i \tag{3}$$

$$\xi_i \geq 0, \quad i = 1, \cdots, p$$

$$\text{For } \mathbf{x}_i \in \mathcal{U}:$$

$$\max_{y_i' \in \mathcal{Y}} \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i') - \max_{y_i'' \neq y_i'} \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i'') \geq 1 - \eta_i \tag{4}$$

$$\eta_i \geq 0, \quad i = p + 1, \cdots, n$$

$$\sum\nolimits_{i=p+1}^{p+u} \mathbb{1}(y_i = 2) = uc, \tag{5}$$

where α, β, and γ are nonnegative trade-off parameters, and c is the proportion of positive examples in unlabeled set $\mathcal{U}$.

In (2), the first term is used to present overfitting. The second term along with the constraint (3) imposed on $\mathcal{P}$ requires that for every $\mathbf{x}_i \in \mathcal{P}$, the score $\boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, 2)$ with the correct label $y_i = 2$ should be larger than the score $\boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, 1)$ associated with the incorrect label $y_i = 1$, by at least $1 - \xi_i$. The third term as well as the constraint (4) enlarges the margin between the most likely label and the less likely one for each unlabeled example $\mathbf{x}_i \in \mathcal{U}$, in which the score associated with the most likely label y' (i.e. $\max_{y'} \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i')$) should be larger than the score associated with the less likely one y'' (i.e. $\max_{y_i'' \neq y_i'} \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i'')$), by at least $1 - \eta_i$. The last term ensures that the similar examples in the feature space should obtain similar labels. Constraint (5) requires the number of positive examples in $\mathcal{U}$ to be identical to uc, where the estimation of c is deferrd to Section 4.1.

Since we regard the unlabeled examples in $\mathcal{U}$ as ambiguously labeled as positive and negative, and the candidate label set of unlabeled examples can be denoted as $\{1, 2\}$, so we follow multi-class SVM [Crammer and Singer, 2001] and partial label learning [Yu and Zhang, 2017] to equip our model with good label discriminability. Therefore, the unique ground-truth label of every unlabeled example can be precisely highlighted. After we have learned the model parameter $\boldsymbol{\omega}$, a test example $\mathbf{x}_i$ is classified as $y_i = \arg\max_{y_i \in \mathcal{Y}} \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i)$.

Note that once the ground-truth labels $\{y_i\}_{i=p+1}^{p+u}$ of the unlabeled examples in $\mathcal{U}$ are determined, our algorithm proceeds to maximize the canonical multi-class margin over each $\mathbf{x}_i \in \mathcal{P} \cup \mathcal{U}$, i.e. $\boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i) - \max_{y_i' \neq y_i} \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i')$. Therefore, by introducing the slack variables $\boldsymbol{\nu} = \{\nu_i\}_{i=1}^{n}$, the previous model can be transformed to

$$\min_{\boldsymbol{\omega}, \boldsymbol{\nu}, \mathbf{y}} \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{\mu}{n}\sum\nolimits_{i=1}^{n}\nu_i + \gamma\sum\nolimits_{i,j=1}^{n}\bar{\mathbf{A}}_{ij}(y_i - y_j)^2 \tag{6}$$

$$s.t. \ \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i) - \max_{y_i' \neq y_i} \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i') \geq 1 - \nu_i \tag{7}$$

$$\nu_i \geq 0, \quad i = 1, \cdots, n$$

$$y_i \in \{1, 2\}, \quad i = 1, \cdots, n \tag{8}$$

$$\sum\nolimits_{i=1}^{n} \mathbb{1}(y_i = 2) = uc + p. \tag{9}$$

In above model, the constraint (7) enforces the maximum margin between different label responses for each training example. In addition, the constraint (8) enforces that the ground-truth label y_i should take a value in the candidate label set $\{1, 2\}$.

Note that our model is different from the LapSVM [Belkin *et al.*, 2006] for semi-supervised learning, as LapSVM does not contain the label disambiguation regularizer as our PULD. Furthermore, if LapSVM is employed for PU learning, it will incorrectly classify all unlabeled examples to positive due to the absence of negative training data.

2.2 Optimization

Since our model (6)-(9) involves mixed-type variables (i.e. integer variables $\mathbf{y}$, and continuous variables $\boldsymbol{\omega}$ and $\boldsymbol{\nu}$), so its optimization is non-trivial. In this section, an alternating optimization procedure is devised to find the optimal solution.

Note that the variable $\mathbf{y}$ appears in both the second term and the third term of (6), so it is difficult to be updated all at once. Therefore, we adopt the variable splitting technique and introduce an auxiliary variable $\mathbf{z} = \{z_1, \ldots, z_n\}$. As a result, our model (6)-(9) is transformed to

$$\min_{\boldsymbol{\omega},\boldsymbol{\nu},\mathbf{y},\mathbf{z}} \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{\mu}{n}\sum_{i=1}^{n}\nu_i + \gamma\sum_{i,j=1}^{n}\bar{\mathbf{A}}_{ij}(z_i - z_j)^2 + \lambda\sum_{i=1}^{n}\mathbb{1}(y_i \neq z_i) \tag{10}$$

$$\begin{aligned} s.t.\ \ & \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i) - \max\nolimits_{y_i' \neq y_i}\boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i') \geq 1 - \nu_i \\ & \nu_i \geq 0, \quad i = 1, \cdots, n \\ & y_i \in \{1, 2\}, \quad i = 1, \cdots, n \\ & \sum\nolimits_{i=1}^{n}\mathbb{1}(y_i = 2) = uc + p, \end{aligned}$$

where the last term in (10) is to keep $\mathbf{y}$ and $\mathbf{z}$ to be consistent, and λ is a trade-off parameter. Now, we are going to solve the three subproblems associated with $\boldsymbol{\omega}$, $\mathbf{y}$, and $\mathbf{z}$, respectively.

Update $\boldsymbol{\omega}$. By fixing $\mathbf{y}$ and $\mathbf{z}$, the subproblem related to $\boldsymbol{\omega}$ is

$$\begin{aligned} \min_{\boldsymbol{\omega},\boldsymbol{\nu}} \ & \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{\mu}{n}\sum\nolimits_{i=1}^{n}\nu_i \\ s.t.\ \ & \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i) - \max_{y_i' \neq y_i}\boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i') \geq 1 - \nu_i \\ & \nu_i \geq 0 \quad i = 1, \cdots, n. \end{aligned} \tag{11}$$

Note that (11) coincides with the well-studied multi-class maximum margin formulation [Crammer and Singer, 2001]. Therefore, (11) can be readily solved by utilizing any off-the-shelf implementation for multi-class SVM [Fan *et al.*, 2008].

Update $\mathbf{y}$. By dropping the unrelated terms to $\mathbf{y}$ in (10), the subproblem regarding $\mathbf{y}$ is

$$\begin{aligned} \min_{\mathbf{y},\boldsymbol{\nu}} \ & \frac{\mu}{n}\sum\nolimits_{i=1}^{n}\nu_i + \lambda\sum\nolimits_{i=1}^{n}\mathbb{1}(y_i \neq z_i) \\ s.t.\ \ & \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i) - \max\nolimits_{y_i' \neq y_i}\boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i') \geq 1 - \nu_i \\ & \nu_i \geq 0, \quad i = 1, \cdots, n \\ & y_i \in \{1, 2\}, \quad i = 1, \cdots, n \\ & \sum\nolimits_{i=1}^{n}\mathbb{1}(y_i = 2) = uc + p. \end{aligned} \tag{12}$$

Let $\zeta_i^{y_i} = \boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i) - \max_{y_i' \neq y_i}\boldsymbol{\omega}^\top\Phi(\mathbf{x}_i, y_i')$, then according to the first two constraints of (12), we have $\nu_i = \max(0, 1 - \zeta_i^{y_i})$, so the first term of the objective function in (12) can be written as

$$\min_{\mathbf{y}} \frac{\mu}{n}\sum\nolimits_{i=1}^{n}\max(0, 1 - \zeta_i^{y_i}). \tag{13}$$

Let $\mathbf{Y} = [Y_{ij}]_{n\times 2}$ be the binary-valued presentation of $\mathbf{y}$, where $Y_{ij} = 1$ indicates that the label of example $\mathbf{x}_i$ is j, and 0 otherwise. Similarly, we may define $\mathbf{Z} = [Z_{ij}]_{n\times 2}$. By further defining the i-th row of an $n \times 2$ coefficient matrix $\mathbf{C}$ as:

$$\mathbf{C}_{i\cdot} = \begin{cases} \big(M, \max(0, 1 - \zeta_i^2)\big) & \text{if } \mathbf{x}_i \in \mathcal{P}, \\ \big(\max(0, 1 - \zeta_i^1), \max(0, 1 - \zeta_i^2)\big) & \text{if } \mathbf{x}_i \in \mathcal{U}, \end{cases} \tag{14}$$

where M is a user-specfied large number to keep the label of each positive example to be constant, we know that the formulation (13) can be transformed to

$$\min_{\mathbf{Y}} \frac{\mu}{n}\sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{2} Y_{ij} \cdot C_{ij}. \tag{15}$$

Since $\mathbf{Y}$ and $\mathbf{Z}$ are both binary-valued, the second term of the objective function in (12) can be written as

$$\min_{\mathbf{y}} \lambda\sum_{i=1}^{n}\mathbb{1}(y_i \neq z_i) \Leftrightarrow -\min_{\mathbf{Y}} \lambda\sum_{i=1}^{n}\sum_{j=1}^{2} Y_{ij} \cdot Z_{ij}. \tag{16}$$

By Combining (12), (13), (15), and (16), the subproblem related to $\mathbf{Y}$ (i.e. the binary-valued representation of $\mathbf{y}$) can be finally transformed to

$$\begin{aligned} \min_{\mathbf{Y}} \ & \sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{2} Y_{ij} \cdot \big(\frac{\mu}{n}C_{ij} - \lambda Z_{ij}\big) \\ s.t.\ \ & \sum\nolimits_{j=1}^{2} Y_{ij} = 1, \quad i = 1, \cdots, n \\ & \sum\nolimits_{i=1}^{n} Y_{ij} = uc + p, \quad j \in \{1, 2\} \\ & Y_{ij} \in \{0, 1\}, \end{aligned} \tag{17}$$

where the first constraint $\sum_{j=1}^{2} Y_{ij} = 1$ ensures that each training example has a unique ground-truth label and the second constraint $\sum_{i=1}^{n} Y_{ij} = uc + p$ controls the amount of classified positive examples.

Note that (17) is a binary integer programming (BIP) problem. By relaxing $Y_{ij} \in \{0, 1\}$ as $Y_{ij} \in [0, 1]$, it can be efficiently solved by employing standard Linear Programming (LP) solvers such as the simplex algorithm or the interior point algorithm [Boyd and Vandenberghe, 2004], where $\mathbf{y}$ can be computed based on $y_i = \arg\max_j Y_{ij}$.

Update $\mathbf{z}$. The subproblem regarding $\mathbf{z}$ is

$$\min_{\mathbf{z}} \gamma\sum\nolimits_{i,j=1}^{n}\bar{\mathbf{A}}_{ij}(z_i - z_j)^2 + \lambda\sum\nolimits_{i=1}^{n}\mathbb{1}(y_i \neq z_i). \tag{18}$$

Considering that the last term in (18) is usually difficult to optimize, here we replace the indicator function with a surrogate ℓ_2 norm term, therefore (18) can be converted to

$$\min_{\mathbf{Z}} \gamma\sum\nolimits_{i,j=1}^{n}\bar{\mathbf{A}}_{ij}\|\mathbf{Z}_i - \mathbf{Z}_j\|^2 + \lambda\sum\nolimits_{i=1}^{n}\|\mathbf{Y}_i - \mathbf{Z}_i\|^2. \tag{19}$$

According to [Zhou *et al.*, 2004], the closed form solution of (19) can be expressed as

$$\mathbf{Z} = \frac{\lambda/\gamma}{1 + \lambda/\gamma}\big(\mathbf{I} - \frac{1}{1 + \lambda/\gamma}\bar{\mathbf{A}}\big)^{-1}\mathbf{Y}. \tag{20}$$

Algorithm 1 The algorithm for solving PULD

Input: positive set $\mathcal{P}$, unlabeled set $\mathcal{U}$; trade-off parameters μ, γ, and λ; stopping criteria ϵ, $iter\_max$;
Initialize $\boldsymbol{\omega} = \mathbf{0}, \mathbf{z} = \mathbf{y} = \mathbf{0}$;
Construct K-NN graph and compute $\bar{\mathbf{A}}$;
Initialize the coefficient matrix $\mathbf{C}$, where $\{\mathbf{C}_{i\cdot}\}_{i=1}^{p} = (0, 1)$ and $\{\mathbf{C}_{i\cdot}\}_{i=p+1}^{u+p} = (1/2, 1/2)$;
Initialize the ground-truth label $\mathbf{y}$ via solving (17);
while not converge **do**
  $Obj_{old} = Obj$;
  Update $\boldsymbol{\omega}$ via solving (11);
  Update $\boldsymbol{C}$ via (14);
  Update $\mathbf{y}$ via solving (17);
  Update $\mathbf{z}$ via solving (19);
  Compute the objective function value Obj via (10);
  $iter := iter + 1$;
  Check the convergence conditions: $Obj - Obj_{old} < \epsilon$ or $iter > iter\_max$;
end while
Output: optimized classifier parameter $\boldsymbol{\omega}$.

Therefore, we can get $\mathbf{z}$ according to $z_i = \arg\max_j Z_{ij}$.

The entire optimization procedure of our PULD is summarized in Algorithm 1, in which the alternating procedure is guaranteed to converge according to [Hong and Luo, 2017].

3 Theoretical Analysis

In this section, we study the generalizability of the proposed PULD algorithm. In our PU learning case, the hypothesis is defined on a scoring function $h \in \mathcal{H} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ with $\mathcal{H}$ denoting the hypothesis space. The label associated with the input $\mathbf{x}$ is the one resulting in the largest score $h(\mathbf{x}, y)$ when $y \in \mathcal{Y}$, which defines the mapping

$$\mathbf{x} \mapsto \arg\max_{y\in\mathcal{Y}} h(\mathbf{x}, y), \tag{21}$$

where $h(\mathbf{x}, y) = \boldsymbol{\omega}^\top \Phi(\mathbf{x}, y)$. We may define the margin loss on positive examples $\rho_p(\mathbf{x}, y)$ and unlabeled examples $\rho_u(\mathbf{x}, y)$ as

$$\rho_p(\mathbf{x}, y) = h(\mathbf{x}, y) - \max_{y'\neq y} h(\mathbf{x}, y') \tag{22}$$

and

$$\rho_u(\mathbf{x}, y) = \max_{y'\in\mathcal{Y}} h(\mathbf{x}, y') - \max_{y''\neq y'} h(\mathbf{x}, y''). \tag{23}$$

To get the empirical margin loss for our PU learning case, we first introduce the following margin loss function [Gong *et al.*, 2019a]:

$$\ell^\rho = \begin{cases} 0 & \text{if } x > \rho, \\ 1 - x/\rho & \text{if } 0 \le x \le \rho, \\ 1 & \text{otherwise}, \end{cases} \tag{24}$$

where ρ is the margin. Let $\rho_{(p)} > 0$ and $\rho_{(u)} > 0$ denote the margins for positive and unlabeled examples, respectively, then we may define the empirical margin loss as

$$\hat{R}_\rho(h) = \frac{1}{p}\sum_{i=1}^{p} \ell^{\rho_{(p)}}(\rho_p(\mathbf{x}_i, y_i)) + \frac{1}{u}\sum_{i=p+1}^{p+u} \ell^{\rho_{(u)}}(\rho_u(\mathbf{x}_i, y_i)). \tag{25}$$

Let $\hat{h}$ be any learned classifier output by PULD and $R(\hat{h})$ be the corresponding expected loss, so our target is to upper bound the generalization error $R(\hat{h}) - \hat{R}_\rho(\hat{h})$.

Theorem 1. *Assume that $\forall \mathbf{x} \in \mathcal{X}$, $\|\mathbf{x}\| \le b$. Let p and u be the sizes of positive set and unlabeled set respectively, α, β be the trade-off parameters in* (2)*, and $\hat{h}$ be the scoring function learned by the proposed algorithm. For any $\delta \ge 0$, with probability at least $1 - \delta$, we have*

$$R(\hat{h}) - \hat{R}_\rho(\hat{h}) \le \frac{8b\sqrt{2(\alpha+\beta)}}{\rho_{(p)}\sqrt{p}} + \frac{8b\sqrt{2(\alpha+\beta)}}{\rho_{(u)}\sqrt{u}} + \sqrt{\frac{\ln(1/\delta)}{2n}}. \tag{26}$$

Before proving this theorem, we first present some useful definitions and lemmas.

Definition 2. ***(Rademacher complexity, [Bartlett and Mendelson, 2002])*** *Let $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_n\}$ be a set of independent Rademacher variables which are uniformly sampled from $\{-1, 1\}$. Let $v_1, \ldots, v_n$ be an independent distributed sample set and $\mathcal{F}$ be a function class. The Rademacher complexity is defined as:*

$$\mathfrak{R}_n(\mathcal{F}) = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(v_i)\right]. \tag{27}$$

Lemma 3. ***(Generalization bound, [Bartlett and Mendelson, 2002])*** *Let $\mathcal{F}$ be a $[0, 1]$-valued function class on $\mathcal{X}$ and $f \in \mathcal{F}$. Given $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ are i.i.d. variables, then for any $\delta > 0$, with probability at least $1 - \delta$, we have*

$$\sup_{f\in\mathcal{F}}\left(\mathbb{E}f(\mathbf{x}) - \frac{1}{n}\sum\nolimits_{i=1}^{n} f(\mathbf{x}_i)\right) \le 2\mathfrak{R}_n(\mathcal{F}) + \sqrt{\frac{\ln(1/\delta)}{2n}}. \tag{28}$$

The Rademacher complexity appeared in (28) can be bounded by the following Lemma, which is

Lemma 4. ***(Talagrand contraction Lemma, [Bartlett and Mendelson, 2002])*** *If $\mathcal{L} : \mathbb{R} \to \mathbb{R}$ is Lipschitz continuous with constant L and satisfies $\mathcal{L}(0) = 0$, then*

$$\mathfrak{R}_n(\mathcal{L} \circ \mathcal{F}) \le L\mathfrak{R}_n(\mathcal{F}), \tag{29}$$

where "$\circ$" represents the composition of two functions.

Now we present the formal proof for Theorem 1:

Proof. Let $\mathcal{H}_1$ be the family of hypothesis mapping $\mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}$ defined by $\mathcal{H}_1 = \{(\mathbf{x}, y) \mapsto \rho^h(\mathbf{x}, y) : h \in \mathcal{H}\}$. By Lemma 3, with probability at least $1 - \delta$, for all $h \in \mathcal{H}$, we have

$$\begin{aligned}\mathbb{E}\left[\ell\left(\rho^h(\mathbf{x}, y)\right)\right] \le \hat{R}_\rho(h) + 2\mathfrak{R}_p(\ell^{\rho_{(p)}} \circ \mathcal{H}_1) \\ + 2\mathfrak{R}_u(\ell^{\rho_{(u)}} \circ \mathcal{H}_1) + \sqrt{\frac{\ln(1/\delta)}{2n}}.\end{aligned} \tag{30}$$

Since $1_{t\le 0} \le \ell(t)$ for all $t \in \mathbb{R}$, the expected error $R(h)$ is the lower bound of $\mathbb{E}\left[\ell\left(\rho^h(\mathbf{x}, y)\right)\right]$, namely $R(h) = \mathbb{E}[1_{[h(\mathbf{x},y)-h(\mathbf{x},y')]\le 0}] \le \mathbb{E}\left[\ell\left(\rho^h(\mathbf{x}, y)\right)\right]$, and thus we have

$$\begin{aligned}R(h) \le \hat{R}_\rho(h) + 2\mathfrak{R}_p(\ell^{\rho_{(p)}} \circ \mathcal{H}_1) \\ + 2\mathfrak{R}_u(\ell^{\rho_{(u)}} \circ \mathcal{H}_1) + \sqrt{\frac{\ln(1/\delta)}{2n}},\end{aligned} \tag{31}$$

where

$$\mathfrak{R}_p(\ell^{\rho_{(p)}} \circ \mathcal{H}_1) = \mathbb{E}\left[\sup_{h\in\mathcal{H}} \frac{1}{p}\sum_{i=1}^{p} \sigma_i \ell^{\rho_{(p)}}\left(\rho_p\left(\mathbf{x}_i, y_i\right)\right)\right] \quad (32)$$

and

$$\mathfrak{R}_u(\ell^{\rho_{(u)}} \circ \mathcal{H}_1) = \mathbb{E}\left[\sup_{h\in\mathcal{H}} \frac{1}{u}\sum_{i=p+1}^{p+u} \sigma_i \ell^{\rho_{(u)}}\left(\rho_u\left(\mathbf{x}_i, y_i\right)\right)\right]. \quad (33)$$

To upper bound the Rademacher complexities in (32) and (33), we first derive an upper bound for the classifiers in $\mathcal{H}$. Due to the optimality of any classifier parameters $\boldsymbol{\omega}$, we have

$$\begin{aligned}&\frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{\alpha}{p}\sum\nolimits_{i=1}^{p}\left(1-\left(\boldsymbol{\omega}^\top\Phi\left(\mathbf{x}_i,2\right)-\boldsymbol{\omega}^\top\Phi\left(\mathbf{x}_i,1\right)\right)\right)_+\\ &+\frac{\beta}{u}\sum_{i=p+1}^{p+u}\left(1-\left(\max_{y_i'}\boldsymbol{\omega}^\top\Phi\left(\mathbf{x}_i,y_i'\right)-\max_{y_i''\neq y_i'}\boldsymbol{\omega}^\top\Phi\left(\mathbf{x}_i,y_i''\right)\right)\right)_+\\ &+\gamma\sum\nolimits_{i,j=1}^{n}\bar{\mathbf{A}}_{ij}\left(y_i-y_j\right)^2 \le \alpha+\beta, \end{aligned} \quad (34)$$

when $\boldsymbol{\omega}$ is set to $\mathbf{0}$. Since each term in (34) is nonnegative, we can get an upper bound for $\|\boldsymbol{\omega}\|^2$, which implys that $\|\boldsymbol{\omega}\|^2 \le 2\alpha + 2\beta$.

Now we are going to bound $\mathfrak{R}_p(\ell^{\rho_{(p)}} \circ \mathcal{H}_1)$ and $\mathfrak{R}_u(\ell^{\rho_{(u)}} \circ \mathcal{H}_1)$. Specially, since the function $\ell(x)$ is $1/\rho$-Lipschitiz, by using Lemma 4, we have $\mathfrak{R}_p(\ell^{\rho_{(p)}} \circ \mathcal{H}_1) \le \mathfrak{R}_p(\mathcal{H}_1)/\rho_{(p)}$ and $\mathfrak{R}_u(\ell^{\rho_{(u)}} \circ \mathcal{H}_1) \le \mathfrak{R}_u(\mathcal{H}_1)/\rho_{(u)}$. For any family of hypothesis mapping $h : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$, we define $\Pi_1(\mathcal{H}) = \{\mathbf{x} \mapsto h(\mathbf{x}, y) : y \in \mathcal{Y}, h \in \mathcal{H}\}$. Based on the theorem 8.1 in [Mohri *et al.*, 2018], we can directly get the upper bounds for $\mathfrak{R}_p(\mathcal{H}_1)$ and $\mathfrak{R}_u(\mathcal{H}_1)$, which are $\mathfrak{R}_p(\mathcal{H}_1) \le 4\mathfrak{R}_p(\Pi_1(\mathcal{H}))$ and $\mathfrak{R}_u(\mathcal{H}_1) \le 4\mathfrak{R}_u(\Pi_1(\mathcal{H}))$. Next we are going to upper bound $\mathfrak{R}_p(\Pi_1(\mathcal{H}))$ and $\mathfrak{R}_u(\Pi_1(\mathcal{H}))$. By noting that $\mathbb{E}[\sigma_i\sigma_j] = 0$ for $i \neq j$, we have

$$\begin{aligned}\mathfrak{R}_p(\Pi_1(\mathcal{H})) &= \mathbb{E}\left[\sup_{y\in\mathcal{Y}}\frac{1}{p}\sum\nolimits_{i=1}^{p}\sigma_i\boldsymbol{\omega}^\top\Phi\left(\mathbf{x}_i,y\right)\right]\\ &\overset{1}{\le}\frac{1}{p}\|\boldsymbol{\omega}\|\mathbb{E}\sqrt{\sum\nolimits_{i=1}^{p}\mathbb{E}\left[\|\mathbf{x}_i\|^2\right]} \le \frac{b\sqrt{2\alpha+2\beta}}{p},\end{aligned} \quad (35)$$

where the Inequality 1 holds because of the Jensen's Inequality and the concave property of the square-root function.

Similarly, we have

$$\mathfrak{R}_u(\Pi_1(\mathcal{H})) \le \frac{b\sqrt{2\alpha+2\beta}}{u}. \quad (36)$$

By combining the formulation (31) and the upper bound for (32) and (33), it is easy to get that for any learned classifier $\hat{h}$, with probability at least $1-\delta$, we have

$$R(\hat{h}) - \hat{R}_\rho(\hat{h}) \le \frac{8b\sqrt{2(\alpha+\beta)}}{\rho_{(p)}\sqrt{p}} + \frac{8b\sqrt{2(\alpha+\beta)}}{\rho_{(u)}\sqrt{u}} + \sqrt{\frac{\ln(1/\delta)}{2n}}, \quad (37)$$

which concludes the proof of Theorem 1. □

Theorem 1 shows that by either increasing the sample size of positive or unlabeled data, the generalization error bound of PULD decreases, which justifies the usefulness of positive and unlabeled data in PU learning. This also guarantees the generalization ability of the proposed learning algorithm.

4 Experiments

To demonstrate the superiority of our proposed PULD to the existing PU methods, we perform intensive experiments on both benchmark and real-world datasets in this section.

4.1 Benchmark Datasets

In this section, we compare our PULD with state-of-the-art PU learning algorithms such as Weighted SVM (WSVM) [Elkan and Noto, 2008], Unbiased PU risk [Du Plessis *et al.*, 2015], Non-Negative PU risk (NNPU) [Kiryo *et al.*, 2017], and Loss Decomposition and Centroid Estimation (LDCE) [Shi *et al.*, 2018] on OpenML[1] benchmark datasets. Specifically, five binary datasets are adopted for algorithm evaluation including *vote*, *diabetes*, *wdbc*, *fri*, and *phishing*, and their configurations are listed in Table 1.

For each dataset, we randomly choose $r = 20\%$, 30%, and 40% positive examples as well as all negative examples as unlabeled and leave the rest positive examples as labeled. Under each r, we conduct five-fold cross validation on every compared method and report the average accuracy over the five independent implementations. Note that all data features have been normalized in advance, and the selected positive examples and the dataset splits are kept identical for all compared methods. In our experiments, the proportion of positive examples in unlabeled set c is assumed to be known during training for all the compared methods, which is also assumed by the existing PU learning works such as [Du Plessis *et al.*, 2015; Kiryo *et al.*, 2017; Shi *et al.*, 2018]. Practically, it can be efficiently estimated by the methods such as [Liu and Tao, 2016; Christoffel *et al.*, 2016]. The parameters of every algorithm have been carefully tuned on the validation set to achieve the best performance. To be specific, the parameter β of NNPU is tuned to the default value 0 on all benchmark datasets. For LDCE, we choose regularization parameter λ from $\{2^{-4}, \ldots, 2^4\}$ and β from $\{0.1, 0.2, \ldots, 0.9\}$ according to [Shi *et al.*, 2018]. Moreover, for the proposed PULD, K is chosen from the candidate set $\{6, 8, 10, 12, 14\}$, δ is chosen from $\{10^{-2}, \ldots, 10^1\}$, and the trade-off parameters μ, γ, and λ in (10) are turned by searching the grid $\{10^{-4}, \ldots, 10^1\}$. Furthermore, we also adopt the t-test with significance level 0.05 to investigate whether our PULD is significantly better than other baselines.

The test accuracies of all methods on the five benchmark datasets are reported in Table 1. We can find that PULD generally achieves the best classification accuracy compared with the baselines. Besides, the accuracies obtained by PULD on the five datasets are all above 75%, which suggests that PULD generates very impressive classification results although it is trained without negative data.

4.2 Real-world Datasets

Here we investigate the performance of WSVM, UPU, NNPU, LDCE, and PULD on the practical image classification task, and *CIFAR-10* [Krizhevsky and Hinton, 2009] and *SVHN* [Netzer *et al.*, 2011] datasets are chosen to test their performance. *CIFAR-10* consists of 60000 32×32 natural im-

[1] https://www.openml.org/

Dataset	(n, d)	r	WSVM	UPU	NNPU	LDCE	PULD
vote	(435,16)	0.2	**0.961±0.017**	0.943±0.014	0.868±0.032 √	0.902±0.013 √	0.959±0.019
		0.3	0.941±0.020	0.905±0.022 √	0.900±0.009 √	0.892±0.010 √	**0.955±0.014**
		0.4	0.918±0.044	0.911±0.012 √	0.891±0.033 √	0.907±0.032 √	**0.948±0.023**
wdbc	(569,30)	0.2	0.943±0.020	0.899±0.023 √	0.877±0.026 √	0.954±0.015	**0.963±0.021**
		0.3	0.939±0.018 √	0.911±0.036 √	0.933±0.016 √	0.950±0.014 √	**0.974±0.006**
		0.4	0.911±0.010 √	0.911±0.019 √	0.939±0.010 √	0.925±0.005 √	**0.967±0.011**
diabetes	(768,8)	0.2	0.751±0.040	0.651±0.003 √	0.730±0.014 √	0.749±0.020 √	**0.787±0.039**
		0.3	0.714±0.017 √	0.686±0.031 √	0.714±0.036 √	0.751±0.017 √	**0.791±0.027**
		0.4	0.730±0.035 √	0.697±0.037 √	0.651±0.003 √	0.734±0.019 √	**0.781±0.040**
fri	(1000,25)	0.2	0.675±0.015 √	0.563±0.002 √	0.732±0.013 √	0.521±0.008 √	**0.779±0.006**
		0.3	0.657±0.026 √	0.579±0.021 √	0.720±0.011 √	0.498±0.034 √	**0.773±0.012**
		0.4	0.630±0.025 √	0.571±0.020 √	0.720±0.011 √	0.485±0.025 √	**0.757±0.029**
phishing	(11055,68)	0.2	0.928±0.002 √	0.872±0.004 √	0.904±0.005 √	0.901±0.005 √	**0.939±0.005**
		0.3	0.928±0.006 √	0.897±0.007 √	0.921±0.004 √	0.792±0.006 √	**0.941±0.006**
		0.4	0.919±0.004 √	0.907±0.010 √	0.927±0.008	0.796±0.011 √	**0.931±0.007**

Table 1: The accuracies of various methods on five OpenML benchmark datasets when $r = 20\%$, 30%, and 40%. The best record under each r is marked in **bold**. "√" indicates that PULD is significantly better than the corresponding methods via paired t-test.

Dataset	r	WSVM	UPU	NNPU	LDCE	PULD
CIFAR-10	0.2	0.829	0.749	0.752	0.772	**0.834**
	0.3	0.820	0.810	0.771	0.761	**0.861**
	0.4	0.787	0.836	0.748	0.701	**0.860**
SVHN	0.2	0.794	0.728	0.779	0.785	**0.851**
	0.3	0.786	0.769	0.810	0.776	**0.852**
	0.4	0.776	0.790	0.828	0.748	**0.850**

Table 2: The test accuracies on the adopted real-world datasets and the best record under each r is marked in **bold**.

ages in 10 classes with each class containing 6000 images. We choose the images of 'airplane', 'auto mobile', 'ship', and 'truck' as negative, and regard the images of 'bird', 'cat', 'deer', 'dog', 'frog', and 'horse' as positive. Therefore, there are 24000 positive examples and 36000 negative examples. *SVHN* contains 99289 32×32 digit images belonging to 10 classes, i.e. the digits '0'-'9', where the negative set is formed by the digit images '1'-'5', and the rest digit images compose the positive set. As a result, we get 34699 positive examples and 64590 negative examples.

In our experiment, we extract the 512-dimensional GIST features for each image. Similar to the above experiments, for each image dataset, the situations of $r = 20\%$, 30%, and 40% are particularly studied. Note that the training set and the test set are split in advance with 50000 training examples and 10000 test examples for *CIFAR-10*, and 73257 training examples and 26032 test examples for *SVHN*. The test accuracies achieved by the compared methods on the two datasets are presented in Table 2, where we can clearly see that our PULD achieves the highest classification accuracy among all comparators on both *CIFAR-10* and *SVHN*. Therefore, the proposed PULD is effective in handling real-world data.

4.3 Parametric Sensitivity

The model (10) of our PULD contains three trade-off parameters μ, γ, and λ. Therefore, this section examines the parametric sensitivity of our model to them. The two real-world datasets *CIFAR-10* and *SVHN* are adopted here. Figure 1 shows the test accuracies of our method on these two datasets when $r = 20\%$. We change μ from 10^{-3} to 10^{0}, and γ and λ from 10^{-4} to 10^{-1}. From Figure 1, we observe that the three parameters have very slight effect on the accuracy, therefore they can be easily tuned for practical implementations.

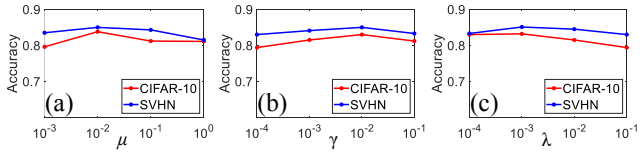

Figure 1: The parametric sensitivity of (a) μ, (b) γ, and (c) λ of our PULD on *CIFAR-10* and *SVHN* datasets.

5 Conclusion

This paper proposed a novel PU learning algorithm named "Positive and Unlabeled learning with Label Disambiguation" (PULD). Specifically, we convert PU learning to a partial label learning problem, and use the disambiguation technique to efficiently identify the ground-truth labels. The proposed model can be easily solved via an alternative optimization process, and its generalization bound has also been theoretically proved. The experimental results on both benchmark and real-world datasets clearly show that PULD is superior to the state-of-the-art PU methods.

Acknowledgments

This research is supported by NSF of China (Nos: 61602246, U1713208), NSF of Jiangsu Province (No: BK20171430), the Fundamental Research Funds for the Central Universities (No: 30918011319), the Summit of the Six Top Talents Program (No: DZXX-027), the Innovative and Entrepreneurial Doctor Program of Jiangsu Province, the Young Elite Scientists Sponsorship Program by Jiangsu Province, the Young Elite Scientists Sponsorship Program by CAST (No: 2018QNRC001), the Australian Research Council Projects (Nos: DP-180103424, DE-1901014738), and Program for Changjiang Scholars.

References

[Bartlett and Mendelson, 2002] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR*, 3(Nov):463–482, 2002.

[Belkin *et al.*, 2006] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7(Nov):2399–2434, 2006.

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Chen *et al.*, 2018] Ching-Hui Chen, Vishal M Patel, and Rama Chellappa. Learning from ambiguously labeled face images. *IEEE T-PAMI*, 40(7):1653–1667, 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.

[Cour *et al.*, 2011] Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *JMLR*, 12(May):1501–1536, 2011.

[Crammer and Singer, 2001] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2(Dec):265–292, 2001.

[Denis *et al.*, 2005] François Denis, Rémi Gilleron, and Fabien Letouzey. Learning from positive and unlabeled examples. *Theoretical Computer Science*, 348(1):70–83, 2005.

[Du Plessis *et al.*, 2014] Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *NeurIPS*, 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 *SIGKDD*, pages 213–220. ACM, 2008.

[Fan *et al.*, 2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *JMLR*, 9(Aug):1871–1874, 2008.

[Feng and An, 2018] Lei Feng and Bo An. Leveraging latent label distributions for partial label learning. In *IJCAI*, pages 2107–2113, 2018.

[Fu *et al.*, 2016] Guangyuan Fu, Jun Wang, Bo Yang, and Guoxian Yu. Neggoa: negative go annotations selection using ontology structure. *Bioinformatics*, 32(19):2996–3004, 2016.

[Gong *et al.*, 2018] Chen Gong, Tongliang Liu, Yuanyan Tang, Jian Yang, Jie Yang, and Dacheng Tao. A regularization approach for instance-based superset label learning. *IEEE T-CYB*, 48(3):967–978, 2018.

[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, Jie Yang, and Jian Yanga. Multi-manifold positive and unlabeled learning for visual analysis. *IEEE T-CSVT*, 2019.

[Hong and Luo, 2017] Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162(1-2):165–199, 2017.

[Jin and Ghahramani, 2003] Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *NeurIPS*, pages 921–928, 2003.

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

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[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.*, 2011] 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, 2011.

[Liu and Tao, 2016] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE T-PAMI*, 38(3):447–461, 2016.

[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. Citeseer, 2002.

[Mohri *et al.*, 2018] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[Nguyen and Caruana, 2008] Nam Nguyen and Rich Caruana. Classification with partial labels. In *SIGKDD*, pages 551–559. ACM, 2008.

[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.

[Wei and Li, 2018] Huihui Wei and Ming Li. Positive and unlabeled learning for detecting software functional clones with adversarial training. In *IJCAI*, pages 2840–2846, 2018.

[Wu and Zhang, 2018] Xuan Wu and Min-Ling Zhang. Towards enabling binary decomposition for partial label learning. In *IJCAI*, pages 2868–2874, 2018.

[Youngs *et al.*, 2014] Noah Youngs, Duncan Penfold-Brown, Richard Bonneau, and Dennis Shasha. Negative example selection for protein function prediction: the nogo database. *PLoS computational biology*, 10(6):e1003644, 2014.

[Youngs *et al.*, 2015] Noah Youngs, Dennis Shasha, and Richard Bonneau. Positive-unlabeled learning in the face of labeling bias. In *ICDMW*, pages 639–645. IEEE, 2015.

[Yu and Zhang, 2017] Fei Yu and Min-Ling Zhang. Maximum margin partial label learning. *Machine Learning*, 106(4):573–593, 2017.

[Zhou *et al.*, 2004] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NeurIPS*, pages 321–328, 2004.