Large-Margin Label-Calibrated Support Vector Machines for Positive and Unlabeled Learning

Chen Gong, *Member, IEEE*, Tongliang Liu, Jian Yang, *Member, IEEE*, and Dacheng Tao, *Fellow, IEEE*

***Abstract*— Positive and unlabeled learning (PU learning) aims to train a binary classifier based on only PU data. Existing methods usually cast PU learning as a label noise learning problem or a cost-sensitive learning problem. However, none of them fully take the data distribution information into consideration when designing the model, which hinders them from acquiring more encouraging performance. In this paper, we argue that the clusters formed by positive examples and potential negative examples in the feature space should be critically utilized to establish the PU learning model, especially when the negative data are not explicitly available. To this end, we introduce a hat loss to discover the margin between data clusters, a label calibration regularizer to amend the biased decision boundary to the potentially correct one, and propose a novel discriminative PU classifier termed " Large-margin Label-calibrated Support Vector Machines" (LLSVM). Our LLSVM classifier can work properly in the absence of negative training examples and effectively achieve the max-margin effect between positive and negative classes. Theoretically, we derived the generalization error bound of LLSVM which reveals that the introduction of PU data does help to enhance the algorithm performance. Empirically, we compared LLSVM with state-of-the-art PU methods on various synthetic and practical data sets, and the results confirm that the proposed LLSVM is more effective than other compared methods on dealing with PU learning tasks.**

***Index Terms*— Label calibration, large margin, positive and unlabeled learning (PU Learning), support vector machines (SVMs).**

Manuscript received May 30, 2018; revised September 25, 2018 and December 26, 2018; accepted December 29, 2018. Date of publication February 6, 2019; date of current version October 29, 2019. This work was supported in part by NSF of China under Grant 61602246 and Grant U1713208, in part by NSF of Jiangsu Province under Grant BK20171430, in part by the Fundamental Research Funds for the Central Universities under Grant 30918011319, in part by the State Key Laboratory of Integrated Services Networks, Xidian University, through the Open Project under Grant ISN19-03, in part by the "111" Program under Grant AH92005, in part by the "Summit of the Six Top Talents" Program under Grant DZXX-027, in part by the Program for Changjiang Scholars, in part by the "Lift Program for Young Talents" of Jiangsu Province, in part by the "CAST Lift Program for Young Talents," and in part by the Australian Research Council Projects under Grant FL-170100117, Grant DE-1901014738, and Grant DP-180103424. *(Corresponding authors: Chen Gong; Jian Yang.)*

C. Gong is with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, P.R. China, and is also with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, 710071, P.R. China. (e-mail: chen.gong@njust.edu.cn).

T. Liu and D. Tao are with the UBTECH Sydney Artificial Intelligence Centre and the School of Information Technologies, Faculty of Engineering and Information Technologies, The University of Sydney, Darlington, NSW 2008, Australia (email: tongliang.liu@sydney.edu.au, dacheng.tao@sydney.edu.au).

J. Yang is with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Laboratory of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, P.R. China. (e-mail: csjyang@njust.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2892403

I. Introduction

POSITIVE and unlabeled learning (PU learning) has attracted intensive research efforts in recent years, of which the target is to train an accurate binary classifier based on only PU examples. PU learning is very useful when the negative training data are absent or too diverse. For example, the fake review detection system for online shopping website can only identify some definite fake reviews (i.e., positive data), but cannot explicitly return the genuine reviews (i.e., negative data) [1]. In other words, only a fraction of positive data are at hand and the remaining unlabeled reviews can be either genuine or fake; therefore, PU learning can be utilized to build a more precise detector for discriminating fake reviews from genuine reviews. In addition, in remote sensing, we are often interested in identifying a specific land type (e.g., "water") from a hyperspectral image [2]. In this case, it is easy to annotate some water regions (i.e., positive data), but is difficult to sufficiently and representatively collect diverse non-water regions (i.e., negative data), as "nonwater region" is an open notion that may contain unlimited land types.

Since PU learning is demanded in many practical applications as mentioned above, it receives a great deal of attention in academic research. Suppose we are given a set of d-dimensional examples $\mathcal{T} = \{\mathbf{x}_i \in \mathcal{X} \in \mathbb{R}^d, i = 1, 2, \ldots, p, p+1, \ldots, n; n = p + u\}$ where n is the amount of examples and $\mathcal{X}$ denotes the feature space. In $\mathcal{T}$, the first p elements with the label $\{y_i\}_{i=1}^{p} = +1$ constitute the positive set $\mathcal{P}$, and the rest u elements form the unlabeled set $\mathcal{U}$ in which the label of every example can be either positive or negative. Therefore, by respectively denoting $\boldsymbol{\omega}$ and b as the coefficient vector and bias term, a PU learning algorithm aims to train a suitable binary classifier $f(\mathbf{x}) = \boldsymbol{\omega}^\top \mathbf{x} + b$ on $\mathcal{T}$ so that it can correctly decide the label $y_t = sgn(f(\mathbf{x}_t))$ from the label space $\mathcal{Y} = \{\pm 1\}$ for an unseen test example $\mathbf{x}_t$.

To conduct PU learning, current algorithms can be mainly divided into three categories according to how the potential negative examples are handled. The first category deploys the two-step strategy which initially identifies some reliable

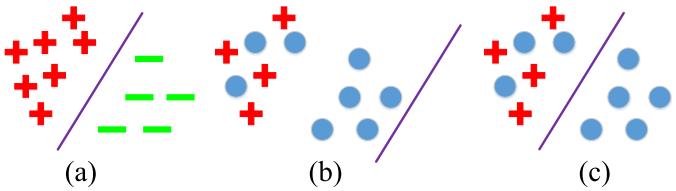

Fig. 1. Motivation of our proposed LLSVM. The red positive sign, green negative sign, and blue circle represent the positive example, negative example, and unlabeled example, respectively. (a) Ideal decision boundary can be found based on the large-margin theory if we know the ground truth labels of training data. (b) Due to the unknown negative data in PU learning, the direct deployment of large-margin theory will lead to the biased decision boundary that classifies all examples into positive. (c) Label calibration is conducted to push the biased decision boundary in (b) to the correct one.

negative data from the unlabeled set and then employs a traditional supervised classifier to perform the ordinary learning under positive and negative examples. Such a two-step framework is mostly followed by some early staged methods such as [3] and [4], of which the drawback lies in that the algorithm performance is dominated by the identified negative training data. If the detection of negative data is inaccurate, the final output could be disastrous. To address this defect, the methods belonging to the second category directly treat all unlabeled examples as negative, among which the original positive examples are regarded as mislabeled. As a consequence, the PU learning problem is converted to a label noise learning problem [5]–[7] and some specific models can be built via label centroid estimation [8] or other probabilistic way [9]. The recent state-of-the-art PU learning approaches usually adopt the third category, namely, designing various unbiased loss functions and casting PU learning as a cost-sensitive learning problem. By imposing different weights on the losses incurred by the unlabeled data regarding positive class and negative class, these methods try to achieve the unbiased estimation of the loss value under traditional binary supervised case. The representative loss functions include double hinge loss [10], ramp loss [11], and nonnegative risk estimator [12].

Although the above methods have obtained good performance to some extent, they did not fully exploit the data distribution information which is actually very important for PU learning. To be specific, although the unlabeled examples do not have explicit labels, they form different clusters corresponding to positive class and underlying negative class, which can be deployed to determine the negative examples and amend the label bias caused by the missing of negative examples. Taking Fig. 1(a) as an example, if the real labels of training examples are available, an optimal decision boundary can be easily placed to the margin formed by the positive cluster and negative cluster. Therefore, the cluster information revealed by the data set is very important for determining the boundary, and here we incorporate the *hat loss* [13] to support vector machines (SVM) to explore the large-margin property inherited by both PU data. However, due to the unavailability of negative training examples, the hat loss for maximizing the between-class margin will generate a biased decision boundary that classifies all training data into positive [see Fig. 1(b)]. To alleviate this phenomenon, we design a novel *label calibration term* that can "push" the biased decision boundary to the correct one [see Fig. 1(c)]. Therefore, our designed model for PU classification is dubbed "Large-margin Label-calibrated SVM" (LLSVM).

Algorithmically, the hat loss will turn the objective function to be nonsmooth and nonconvex, and the introduced label calibration term is not decomposable with respect to each of the training examples, so these two regularizers make the LLSVM model difficult to solve. To facilitate the optimization, we use a smooth Gaussian-like function to approximate the hat loss and find an upper bound of label calibration term that specifically penalizes the label bias on every training data. As a result, the LLSVM model can be efficiently solved via minibatch stochastic gradient descent (SGD). Theoretically, we derived the generalization error bound of the proposed LLSVM, which suggests that LLSVM is guaranteed to obtain satisfactory performance when PU data are available for training. Experimentally, we tested our method on toy data set, benchmark data sets, and real-world data sets of different areas, and the results confirm that our LLSVM can obtain superior results to the state-of-the-art PU learning algorithms.

II. Related Work

As an important branch of weakly supervised learning [14], [15], the study of PU learning can be dated back to [3] and [4], which follow a two-stage strategy that initially identifies some reliable negative examples from the unlabeled set and then performs the traditional supervised learning. In this strategy, the first stage is critical and various methods have been developed to find the probable negative examples. Liu *et al.* [3] develop a "Spy Technique" that randomly puts a set of positive examples to $\mathcal{U}$, and thus, the expectation–maximization algorithm can be employed to detect the definite negative examples. Differently, Liu *et al.* [4] adopt a naive Bayesian classifier to determine the most likely negative documents for text classification. Xiao *et al.* [16] build a similarity graph and employ K-means to find the representative positive and negative prototypes. Above "detect-then-classify" strategy actually adopts the simple-to-complex sequence [17] to progressively decide the definite negative examples and, thus, is very straightforward, but the drawback is also obvious, namely, the detection of negative examples can be inaccurate and the final performance will be degraded as a result.

Considering that precisely detecting the negative examples is nontrivial, some later methods transform PU learning problem into the learning problem with label noise and thus some existing methodologies can be adapted to solve PU learning problem. Lee and Liu [9] regard all the unlabeled examples as negative, among which the original positive examples are treated as mislabeled. Then, they perform logistic regression on the data set and generate the real-valued probabilistic outputs rather than the thresholded binary outputs as previous "detect-then-classify" strategy.

However, above noise learning strategy cannot work properly if the noise rate is too high, which occurs when a large number of positive examples are hidden in the unlabeled

set. Therefore, the recent state-of-the-art methods usually formulate PU learning as cost-sensitive learning, by which the losses incurred by the unlabeled examples regarding positive and negative classes are re-weighted. For example, Elkan and Noto [18] first train a nontraditional classifier on $\mathcal{P}$ and $\mathcal{U}$ to estimate the examples' weights on a validation set, and then model the probability $P(y = 1|\mathbf{x})$ using a weighted SVM. In addition, Plessis *et al.* [11] reweight the per-class costs of every example according to the estimated class priors, and propose to use a nonconvex ramp loss to conduct PU classification. To avoid the nonconvexity, a convex unbiased double hinge loss is presented in [10], of which the key idea is to use a weighted ordinary convex loss for unlabeled data and a weighted composite convex loss function for positive data. However, the empirical risks of this method on the training data might be negative, and serious overfitting will happen in this case, so an improvement is made by Kiryo *et al.* [12] which develop a nonnegative risk estimator. Theoretically, Niu *et al.* [19] have proved that cost-sensitive PU learning is guaranteed to outperform fully supervised classifier in certain cases even though the negative data are absent. They also shed light on that the loss functions proposed in [10] and [11] are unbiased to the real loss for ordinary binary classification.

Apart from above-mentioned models, there are some other works that extend the ordinary PU learning to different settings. Zhou *et al.* [20] extend the ordinary single-view PU model to multiview cases based on density ratio estimation. Xu *et al.* [21] generalize the single positive class to multiple positive classes and propose multi-PU learning based on the discriminative multiclass formulation.

Although the existing methods have obtained encouraging performance to some extent, they lack the capability of discovering the large-margin effect revealed by the locations of training examples in the feature space. Therefore, this paper aims to design a novel discriminative PU learning model that maximizes the margin between real positive and negative classes without the aid of known negative examples. Note that [22] also advocates employing margin theory to conduct PU learning. However, this method heavily relies on the large positive margin assumption that all positive examples are located far away from the optimal decision boundary, which may not be true practically. In addition, the margin in [22] is caused by the truncated Gaussian distribution, which is different from the interclass margin to be maximized in our method.

III. Proposed Method

In this section, we first establish our proposed PU learning model LLSVM in Section III-A, and then explain the optimization process for solving LLSVM in Section III-B. Finally, the parameter estimation is detailed in Section III-C.

A. Model

In training stage, our target is to find a real-valued decision function $f : \mathbb{R}^d \to \mathbb{R}$ based on the training set $\mathcal{T} = \mathcal{P} \cup \mathcal{U}$, and the classification is performed according to its sign, namely, $y = sgn(f) \in \{-1, +1\}$. Given n, p, and u as the sizes of $\mathcal{T}$, $\mathcal{P}$, and $\mathcal{U}$, respectively, our model is formulated as

$$\min_{\boldsymbol{\omega},b,\boldsymbol{\xi},\eta} \quad \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{\alpha}{p}\sum_{i=1}^{p}\xi_i + \frac{\beta}{u}\sum_{i=p+1}^{p+u}\xi_i + \gamma\,\eta \tag{1}$$

$$\text{s.t. } \boldsymbol{\omega}^\top \mathbf{x}_i + b \geq 1 - \xi_i, \quad i = 1, 2, \ldots, p \tag{2}$$

$$\left|\boldsymbol{\omega}^\top \mathbf{x}_i + b\right| \geq 1 - \xi_i, \quad i = p+1, \ldots, p+u \tag{3}$$

$$\frac{1}{u}\sum_{i=p+1}^{p+u}\Phi(\boldsymbol{\omega}^\top \mathbf{x}_i + b) \leq t + \eta \tag{4}$$

$$\xi_i \geq 0, \quad i = 1, 2, \ldots, n$$

$$\eta \geq 0$$

where α, β, and γ are nonnegative tradeoff parameters; t is the variable to be estimated that will be detailed in Section III-C; and $\Phi(z) = (2/\pi)\arctan(z)$ squashes the value of z to the range $[-1, 1]$. The first term in the objective function (1) is a regularizer to present overfitting. The second term in (1) along with the constraint (2) imposed on the positive set $\mathcal{P}$ requires that every $\mathbf{x}_i \in \mathcal{P}$ is classified as positive by the trained model. The third term in (1) and the constraint (3) encourage the examples in $\mathcal{U}$ to obtain "clear" labels, namely the value of model output $f(\mathbf{x}_i)$ should ideally be larger than 1 or less than -1. The left-hand side of (4) computes the mean value of unlabeled examples' labels, and we hope that this mean value is no larger than a threshold t. This constraint informs the algorithm that the unlabeled set $\mathcal{U}$ contains a considerable amount of negative examples, and thus, the determination of their labels cannot be dominated by the observed positive examples. Note that above constraints (2), (3), and (4) are not "hard" as the nonnegative slack variables ξ_i $(i = 1, \ldots, n)$ and η are incorporated. However, we want these constraints to be satisfied as much as possible by minimizing the corresponding slack variables.

For simplicity, we compactly express the decision function as $f(\mathbf{x}) = \boldsymbol{\omega}^\top \mathbf{x}$ where an additional value 1 is added to the $(d+1)$th dimension of $\mathbf{x}$'s feature vector and $\boldsymbol{\omega}$ is augmented correspondingly. Therefore, above constrained optimization model can be rewritten in an unconstrained way, namely,

$$\begin{aligned}\min_{\boldsymbol{\omega}} \ & \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{\alpha}{p}\sum_{i=1}^{p}\max(1 - \boldsymbol{\omega}^\top \mathbf{x}_i, 0) \\ & + \frac{\beta}{u}\sum_{i=p+1}^{p+u}\max(1 - |\boldsymbol{\omega}^\top \mathbf{x}_i|, 0) \\ & + \gamma\,\max\left(\frac{1}{u}\sum_{i=p+1}^{p+u}\Phi(\boldsymbol{\omega}^\top \mathbf{x}_i) - t, 0\right).\end{aligned} \tag{5}$$

In (5), the second term which rewrites (2) is the *hinge loss* on p positive examples. The third term related to (3) is called *hat loss* [13] that drives our LLSVM to assign confident labels to unlabeled examples. It can be observed that the loss value will be above zero if the label value $f(\mathbf{x}) = \boldsymbol{\omega}^\top \mathbf{x}_i$ falls into the range $[-1, 1]$. Furthermore, the maximum loss 1 will be obtained if the label is 0, which means that the assigned label does not contain any class information. On the contrary, if the

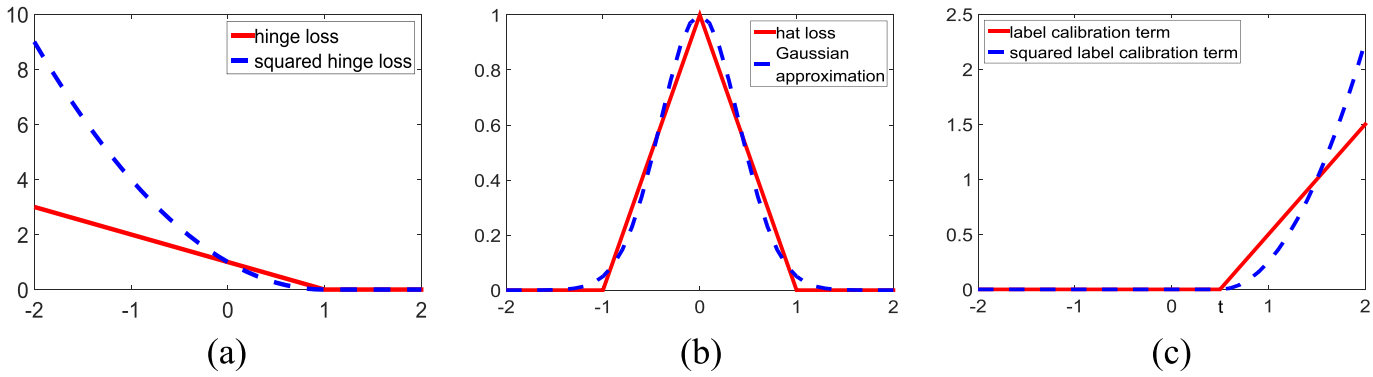

Fig. 2. Regularizers (red curves) adopted by our LLSVM and their approximations (blue dashed curves).

label is larger than 1 or smaller than -1, the loss will be 0, which suggests that $\mathbf{x}_i$ has a clear class label. In other words, we require the optimal decision boundary to travel thorough a margin such that the examples belonging to two classes are sufficiently separated. Therefore, by penalizing the label values that are within $[-1, 1]$, the obtained classifier is able to find the maximum margin between classes. Note that if we only minimize the first three terms, our algorithm will produce a biased classifier that classifies all training examples into positive [see Fig. 1(b)]. Consequently, an additional *label calibration term* [i.e., the last term in (5)] equivalent to (4) is introduced. By restricting the label mean of unlabeled examples to a value that is not larger than t, the LLSVM will be aware of the existence of negative examples, therefore the improper decision boundary will be calibrated to the correct one [see Fig. 1(c)]. The above-mentioned hinge loss, hat loss, and label calibration term are plotted in Fig. 2 (red curves), from which we observe that they are not smooth, and thus, posing a great difficulty for the gradient-based optimization approaches to solve (5). Therefore, to make them differentiable everywhere, we use squared hinge loss and squared label calibration term to replace the original hinge loss and label calibration term, and utilize a smooth Gaussian-like function [23] to approximate the hat loss. The three adopted surrogate regularizers are visualized by the blue dashed curves shown in Fig. 2. As a result, the model (5) is transformed to

$$\min_{\boldsymbol{\omega}} \frac{1}{2}\|\boldsymbol{\omega}\|^2+\frac{\alpha}{p}\sum_{i=1}^{p}[\max(1-\boldsymbol{\omega}^\top\mathbf{x}_i,0)]^2+\frac{\beta}{u}\sum_{i=p+1}^{p+u}e^{-3(\boldsymbol{\omega}^\top\mathbf{x}_i)^2} +\gamma\left[\max\left(\frac{1}{u}\sum_{i=p+1}^{p+u}\Phi(\boldsymbol{\omega}^\top\mathbf{x}_i)-t,0\right)\right]^2. \quad (6)$$

From (6), we see that all unlabeled examples should be visited when computing the last squared label calibration term. Consequently, the derivative of this regularizer to $\boldsymbol{\omega}$ will be very complicated due to the correlation of the unlabeled examples. To address this problem, we further derive the upper bound of $[\max((1/u)\sum_{i=p+1}^{p+u}\Phi(\boldsymbol{\omega}^\top\mathbf{x}_i)-t,0)]^2$ and then minimize this upper bound. According to Jensen's inequality, we have

$$\max\left(\frac{1}{u}\sum_{i=p+1}^{p+u}\Phi(\boldsymbol{\omega}^\top\mathbf{x}_i)-t,0\right) \le \frac{1}{u}\sum_{i=p+1}^{p+u}\max(\Phi(\boldsymbol{\omega}^\top\mathbf{x}_i)-t,0) \quad (7)$$

therefore the squared label calibration term is upper bounded by

$$\left[\max\left(\frac{1}{u}\sum_{i=p+1}^{p+u}\Phi(\boldsymbol{\omega}^\top\mathbf{x}_i)-t,0\right)\right]^2 \overset{1}{\le} \frac{1}{u^2}\left[\sum_{i=p+1}^{p+u}\max(\Phi(\boldsymbol{\omega}^\top\mathbf{x}_i)-t,0)\right]^2 \overset{2}{\le} \frac{1}{u}\sum_{i=p+1}^{p+u}\left[\max(\Phi(\boldsymbol{\omega}^\top\mathbf{x}_i)-t,0)\right]^2 \quad (8)$$

where the second inequality holds true due to the inequality of sum of squares. Finally, our model for training an LLSVM classifier is expressed as

$$\min_{\boldsymbol{\omega}} \frac{1}{2}\|\boldsymbol{\omega}\|^2+\frac{\alpha}{p}\sum_{i=1}^{p}[\max(1-\boldsymbol{\omega}^\top\mathbf{x}_i,0)]^2+\frac{\beta}{u}\sum_{i=p+1}^{p+u}e^{-3(\boldsymbol{\omega}^\top\mathbf{x}_i)^2} +\frac{\gamma}{u}\sum_{i=p+1}^{p+u}[\max(\Phi(\boldsymbol{\omega}^\top\mathbf{x}_i)-t,0)]^2. \quad (9)$$

B. Optimization

Note that the problem (9) is nonconvex due to the third term, so we use the well-known minibatch SGD for optimization [24]. Minibatch SGD differs from the conventional SGD algorithm in that it randomly splits the training data set into small batches that are used to compute the gradient and update model coefficients. For each minibatch, the average of the gradient values on all examples is adopted, which reduces the variance of the gradient when compared with the traditional SGD that processes only one example per iteration. Therefore, minibatch SGD seeks to find a balance

between the stability of SGD and the efficiency of full gradient descent. In our implementation, we shuffle the examples in the training set $\mathcal{T}$ and divide the entire $\mathcal{T}$ into N nonoverlapped minibatches, and then use the batches successively to compute the gradient and update $\boldsymbol{\omega}$. One cycle through all the minibatches constitutes a training epoch, and such training epoch is repeated $MaxEpoch$ times. Note that one can shuffle the data before every epoch, but in our work, we simply shuffle the training data at the beginning of training stage to simulate the independent identically distributed (i.i.d.) sampling.

By denoting the objective in (9) as $J(\boldsymbol{\omega})$, each summed function J_i associated with $\mathbf{x}_i$ (i takes a value from $1, \ldots, n$) is formulated as

$$J_i = \begin{cases} \dfrac{1}{2}\|\boldsymbol{\omega}\|^2 + \dfrac{\alpha}{p}[\max(1 - \boldsymbol{\omega}^\top \mathbf{x}_i, 0)]^2, \mathbf{x}_i \in \mathcal{P}; \\ \dfrac{1}{2}\|\boldsymbol{\omega}\|x^2 + \dfrac{\beta}{u}e^{-3(\boldsymbol{\omega}^\top \mathbf{x}_i)^2} + \dfrac{\gamma}{u}[\max(\Phi(\boldsymbol{\omega}^\top \mathbf{x}_i) - t, 0)]^2 \\ \qquad\qquad\qquad\qquad \mathbf{x}_i \in \mathcal{U}. \end{cases} \tag{10}$$

As a result, for $\mathbf{x}_i \in \mathcal{P}$, the gradient on $\boldsymbol{\omega}$ is

$$\nabla_{\boldsymbol{\omega}}(J_i) = \boldsymbol{\omega} + \frac{2\alpha}{p}\min(\boldsymbol{\omega}^\top \mathbf{x}_i - 1, 0)\mathbf{x}_i. \tag{11}$$

For $\mathbf{x}_i \in \mathcal{U}$, the gradient on $\boldsymbol{\omega}$ is

$$\nabla_{\boldsymbol{\omega}}(J_i) = \boldsymbol{\omega} - \frac{6\beta}{u}\boldsymbol{\omega}^\top \mathbf{x}_i e^{-3(\boldsymbol{\omega}^\top \mathbf{x}_i)^2}\mathbf{x}_i + \frac{4\gamma}{\pi u[1 + (\boldsymbol{\omega}^\top \mathbf{x}_i)^2]}\max(\Phi(\boldsymbol{\omega}^\top \mathbf{x}_i) - t, 0)\mathbf{x}_i. \tag{12}$$

Therefore, the updating rule for each iteration is

$$\boldsymbol{\omega} := \boldsymbol{\omega} - \frac{\tau}{m}\sum_{i=1}^{m}\nabla_{\boldsymbol{\omega}}(J_i), \tag{13}$$

where $m = n/N$ is the minibatch size and τ is step size that is fixed to 0.01 throughout this paper.

C. Estimation of t

For LLSVM, the parameter t in the proposed label calibration term determines the "strength" of pushing the biased decision boundary to the potentially correct one. Manually tuning this parameter may lead to the wrong placement of decision function in the feature space and, thus, degrading the performance. Here, we propose an adaptive way to estimate the optimal t.

Since t represents the upper bound of the mean value of unlabeled examples' real labels, it can be easily estimated if we know the class prior $\pi = P(y = 1)$ in $\mathcal{U}$. To this end, some existing methods for estimating the class prior can be deployed such as [25]–[30]. In this paper, we use the method proposed in [30] which properly penalizes the divergences for model fitting to mitigate the error caused by the absence of negative samples. Specifically, suppose $Q(\mathbf{x}; \pi) = \pi P(\mathbf{x}|y = 1) + (1 - \pi)P(\mathbf{x}|y = -1)$ is the partial model revealed by $\mathcal{P}$, and $P(\mathbf{x})$ is the density of the unlabeled data associated with $\mathcal{U}$, then [30] proposes to estimate the class prior by penalizing the divergence between $Q(\mathbf{x}; \pi)$ and $P(\mathbf{x})$, namely,

$$\pi := \arg\min_{0 \le \pi \le 1} Div_f(\pi) \tag{14}$$

where "Div" denotes the divergence that is defined by

$$Div_f(\pi) := \int f\left(\frac{Q(\mathbf{x}; \pi)}{P(\mathbf{x})}\right)P(\mathbf{x})d\mathbf{x}. \tag{15}$$

By utilizing the Fenchel duality bounding technique [31] for divergence and employing penalized L_1-distance $f(z) = |z - 1|$ as the practical divergence penalization, the problem (14) can be transformed to the following constrained optimization problem, namely,

$$(c_1^*, \ldots, c_B^*) = \arg\min_{c_1, \ldots, c_B} \sum_{i=1}^{B}\frac{\bar{r}}{2}c_i^2 - \sum_{i=1}^{B}c_i\bar{\rho}_i \quad \text{s.t.} \quad c_i \ge 0, \quad i = 1, \ldots, B \tag{16}$$

where $\bar{r} > 0$ is a tuning parameter, and

$$\bar{\rho}_i = \frac{\pi}{n}\sum_{j=1}^{n}\varphi_i(\mathbf{x}_j) - \frac{1}{u}\sum_{j=p+1}^{p+u}\varphi_i(\mathbf{x}_j). \tag{17}$$

In (17), $\varphi_i(\mathbf{x})$ is the ith element from a set of nonnegative Gaussian base functions $\{\varphi_i(\mathbf{x})\}_{i=1}^{B}$.

The solution for the optimization problem (16) can be analytically expressed as

$$c_i = \frac{1}{\bar{r}}\max(0, \bar{\rho}_i). \tag{18}$$

Therefore, the class prior π is selected to minimize the following estimator

$$\widehat{penL_1}(\pi) = \frac{1}{\bar{r}}\sum_{i=1}^{B}\max(0, \bar{\rho}_i)\bar{\rho}_i - \pi + 1 \tag{19}$$

where $\widehat{penL_1}(\pi)$ denotes the estimated L_1-distance related to a specific π. In our work, the optimal π is searched from 0.05 to 0.95 with the interval 0.05. Given the estimated class prior on $\mathcal{U}$ as π, the parameter t in (9) is computed as

$$t = \frac{u\pi - u(1 - \pi)}{u} = 2\pi - 1. \tag{20}$$

The complete LLSVM algorithm for PU classification is summarized in Algorithm 1.

IV. Theoretical Analyses

This section studies the proposed LLSVM algorithm from the theoretical aspects. First, we analyze the computational complexity of the involved SGD optimization process for model training, and second, we derive the generalization bound for our LLSVM.

Algorithm 1: Summarization of the Proposed LLSVM

Input: trade-off parameters α, β and γ; number of mini-batches N; maximum number of epochs $MaxEpoch$; step size $\tau = 0.01$

Output: the optimal $\boldsymbol{\omega}$

1 Compute π by minimizing (19);
2 Estimate t via (20);
3 Initialize $\boldsymbol{\omega}$ randomly;
4 **for** $epoch = 1 : MaxEpoch$ **do**
5 **for** $batch = 1 : N$ **do**
6 $\forall \mathbf{x}_i \in \mathcal{P}$, compute $\nabla_{\boldsymbol{\omega}}(J_i)$ via (11);
7 $\forall \mathbf{x}_i \in \mathcal{U}$, compute $\nabla_{\boldsymbol{\omega}}(J_i)$ via (12);
8 Update $\boldsymbol{\omega}$ via (13);
9 **end**
10 **end**

A. Computational Complexity

As mentioned in Section III-B, the designed LLSVM model can be efficiently solved via SGD, so here we deduce the computational complexity for model optimization. Given d as the data dimensionality, one can easily find that the complexities for calculating the gradients in (11) and (12) are $\mathcal{O}(d)$ (Lines 6 and 7 in Algorithm 1). By further considering that (13) also takes $\mathcal{O}(d)$ complexity (Line 8 in Algorithm 1), we know that the parameter updating on a minibatch of size m takes $\mathcal{O}(md + d)$ complexity. Suppose there are totally N minibatches and $MaxEpoch$ epochs, the total complexity for running the SGD for LLSVM is $\mathcal{O}(MaxEpoch \cdot N \cdot (md + d))$. Note that the complexity for solving LLSVM is linear to d, so the optimal solution can be efficiently figured out. Moreover, since the term $\boldsymbol{\omega}^\top \mathbf{x}_i$ appears several times in (12), it can be precomputed for $\mathbf{x}_i \in \mathcal{U}$. Therefore, the computational burden can be further reduced for practical implementation.

B. Generalization Bound

In this section, we study the generalizability of the proposed learning algorithm. We show that the empirical classification risk of any classifier learned by the proposed algorithm converges to its expected classification risk when the amounts of both PU examples are sufficiently large.

Specifically, we define the expected classification error of a classifier $\boldsymbol{\omega}$ by

$$R(\boldsymbol{\omega}) = \mathbb{E}[1_{Y\boldsymbol{\omega}^\top \boldsymbol{X} \le 0}] \tag{21}$$

where $1_{\{\cdot\}}$ is the indicator function representing the 0-1 loss function, and $(\boldsymbol{X}, Y)$ stands for a pair of random variables distributed from the unknown joint distribution P on the set $\mathcal{X} \times \mathcal{Y}$.

Since we only have PU data, we will rewrite the expected risk accordingly. Let

$$R_1(\boldsymbol{\omega}) = \int P(\boldsymbol{X}|Y = 1) 1_{\boldsymbol{\omega}^\top \boldsymbol{X} \le 0} d\boldsymbol{X} \tag{22}$$

and

$$R_{-1}(\boldsymbol{\omega}) = \int P(\boldsymbol{X}|Y = -1) 1_{\boldsymbol{\omega}^\top \boldsymbol{X} \ge 0} d\boldsymbol{X} \tag{23}$$

denote the false positive rate and false negative rate, respectively, we have

$$R(\boldsymbol{\omega}) = \mathbb{E}[1_{Y\boldsymbol{\omega}^\top \boldsymbol{X} \le 0}] = \pi R_1(\boldsymbol{\omega}) + (1 - \pi) R_{-1}(\boldsymbol{\omega}). \tag{24}$$

Moreover, we define

$$\begin{aligned} R_u(\boldsymbol{\omega}) &= \mathbb{E}[1_{\boldsymbol{\omega}^\top \boldsymbol{X} \ge 0}] \\ &= \int P(\boldsymbol{X}) 1_{\boldsymbol{\omega}^\top \boldsymbol{X} \ge 0} d\boldsymbol{X} \\ &= \int (P(\boldsymbol{X}|Y = 1) P(Y = 1) \\ &\quad + P(\boldsymbol{X}|Y = -1) P(Y = -1)) 1_{\boldsymbol{\omega}^\top \boldsymbol{X} \ge 0} d\boldsymbol{X} \\ &= \pi(1 - R_1(\boldsymbol{\omega})) + (1 - \pi) R_{-1}(\boldsymbol{\omega}). \end{aligned} \tag{25}$$

Therefore, by combining (24) and (25), we have

$$R(\boldsymbol{\omega}) = 2\pi R_1(\boldsymbol{\omega}) + R_u(\boldsymbol{\omega}) - \pi. \tag{26}$$

Accordingly, we may define the empirical margin error of the classifier $\boldsymbol{\omega}$ as

$$\hat{R}^\rho(\boldsymbol{\omega}) = \frac{2\pi}{p} \sum_{i=1}^{p} 1_{\boldsymbol{\omega}^\top \boldsymbol{X}_i \le \rho} + \frac{1}{u} \sum_{i=1}^{u} 1_{-\boldsymbol{\omega}^\top \boldsymbol{X}_i \le \rho} - \pi, \tag{27}$$

where ρ is the margin. Note that the prior π can be efficiently estimated from the PU data, and thus, the empirical margin error can be easily estimated from the data. Also note that when $\rho \to 0$, we have $\mathbb{E}[\hat{R}^\rho(\boldsymbol{\omega})] \to R(\boldsymbol{\omega})$.

Let $\hat{\boldsymbol{\omega}}$ be any learned classifier output by the proposed learning algorithm, so our target is to upper bound the generalization error $R(\hat{\boldsymbol{\omega}}) - \hat{R}^\rho(\hat{\boldsymbol{\omega}})$.

Theorem 1: Assume the feature space $\mathcal{X}$ is upper bounded, i.e., $\forall \mathbf{x} \in \mathcal{X}$, $\|\mathbf{x}\| \le b'$. Let p and u be the sizes of positive set and unlabeled set, respectively, α, β, and γ be the tradeoff parameters in (9), and $\hat{\boldsymbol{w}}$ be the classifier parameter learned by the proposed algorithm. For any $\delta > 0$, with probability at least $1 - \delta$, we have

$$\begin{aligned} R(\hat{\boldsymbol{\omega}}) - \hat{R}^\rho(\hat{\boldsymbol{\omega}}) \le{}& \frac{2b'\sqrt{2(\alpha + \beta + \gamma t^2)}}{\rho\sqrt{p}} \\ &+ \frac{2b'\sqrt{2(\alpha + \beta + \gamma t^2)}}{\rho\sqrt{u}} + \sqrt{\frac{\ln(1/\delta)}{2n}}. \end{aligned} \tag{28}$$

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

Definition 2 (Rademacher Complexity, [32]): Let $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_n\}$ be a set of independent Rademacher variables which are uniformly sampled from $\{-1, 1\}$. Let $\upsilon_1, \ldots, \upsilon_n$ be an independent distributed sample set and $\mathcal{F}$ be a function class. The Rademacher complexity is defined as

$$\mathfrak{R}_n(\mathcal{F}) = \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(\upsilon_i) \right]. \tag{29}$$

The Rademacher complexity is a data-dependent complexity measure, which is often used to derive the dimensionality independent generalization error bound of a decision function f, which is

Lemma 3 (Generalization Bound, [32]): Let $\mathcal{F}$ be a $[0, 1]$-valued function class on $\mathcal{X}$ and $f \in \mathcal{F}$. Given $X_1, \ldots, X_n \in$

$\mathcal{X}$ are i.i.d. variables, then for any $\delta > 0$, with probability at least $1-\delta$, we have

$$\sup_{f\in\mathcal{F}}\left(\mathbb{E}f(X)-\frac{1}{n}\sum_{i=1}^{n}f(X_i)\right)\le 2\mathfrak{R}_n(\mathcal{F})+\sqrt{\frac{\ln(1/\delta)}{2n}}. \quad (30)$$

In addition, the Rademacher complexity appeared in (30) can be usually upper bounded by following the Talagrand contraction Lemma [32], which is

Lemma 4 (Talagrand Contraction Lemma, [32]): If $\ell : \mathbb{R}\to\mathbb{R}$ is Lipschitz continuous with constant L and satisfies $\ell(0)=0$, then

$$\mathfrak{R}_n(\ell\circ\mathcal{F})\le L\mathfrak{R}_n(\mathcal{F}) \quad (31)$$

where $\ell\circ\mathcal{F}$ represents the composition of ℓ and all $f\in\mathcal{F}$.

Now, we present the formal proof for Theorem 1: To facilitate the proof, we first introduce the following function:

$$\ell^{\rho}(x)=\begin{cases}0 & \text{if } x\ge\rho\\ 1-x/\rho & \text{if } 0\le x\le\rho\\ 1 & \text{otherwise}\end{cases} \quad (32)$$

where ρ is the margin as defined in (27). Let

$$\begin{aligned}R(\ell^{\rho}\circ\boldsymbol{\omega}) &= \mathbb{E}[\ell^{\rho}(Y\boldsymbol{\omega}^{\top}\boldsymbol{X})]\\ &= 2\pi R_1(\ell^{\rho}\circ\boldsymbol{\omega})+R_u(\ell^{\rho}\circ\boldsymbol{\omega})-\pi \quad (33)\\ \hat{R}(\ell^{\rho}\circ\boldsymbol{\omega}) &= \frac{2\pi}{p}\sum_{i=1}^{p}\ell^{\rho}(\boldsymbol{\omega}^{\top}\boldsymbol{X}_i)\\ &\quad+\frac{1}{u}\sum_{i=1}^{u}\ell^{\rho}(-\boldsymbol{\omega}^{\top}\boldsymbol{X}_i)-\pi \quad (34)\end{aligned}$$

where $\ell^{\rho}\circ\boldsymbol{\omega}$ stands for the composite function, and

$$R_1(\ell^{\rho}\circ\boldsymbol{\omega})=\int P(\boldsymbol{X}|Y=1)\ell^{\rho}(\boldsymbol{\omega}^{\top}\boldsymbol{X})d\boldsymbol{X} \quad (35)$$

and

$$R_u(\ell^{\rho}\circ\boldsymbol{\omega})=\int P(\boldsymbol{X}|Y=1)\ell^{\rho}(-\boldsymbol{\omega}^{\top}\boldsymbol{X})d\boldsymbol{X}. \quad (36)$$

We have that $\mathbb{E}[\hat{R}(\ell^{\rho}\circ\boldsymbol{\omega})]=R(\ell^{\rho}\circ\boldsymbol{\omega})$. It can also be easily verified that $R(\boldsymbol{\omega})\le R(\ell^{\rho}\circ\boldsymbol{\omega})$ and that $\hat{R}^{\rho}(\boldsymbol{\omega})\ge\hat{R}(\ell^{\rho}\circ\boldsymbol{\omega})$, which implies

$$R(\boldsymbol{\omega})-\hat{R}^{\rho}(\boldsymbol{\omega})\le R(\ell^{\rho}\circ\boldsymbol{\omega})-\hat{R}(\ell^{\rho}\circ\boldsymbol{\omega}). \quad (37)$$

Let $\mathcal{W}$ be the set of all possible learned classifiers, then we have

$$R(\hat{\boldsymbol{\omega}})-\hat{R}^{\rho}(\hat{\boldsymbol{\omega}})\le\sup_{\boldsymbol{\omega}\in\mathcal{W}}(R(\ell^{\rho}\circ\boldsymbol{\omega})-\hat{R}(\ell^{\rho}\circ\boldsymbol{\omega})). \quad (38)$$

We are now going to upper bound the defect $R(\ell^{\rho}\circ\boldsymbol{\omega})-\hat{R}(\ell^{\rho}\circ\boldsymbol{\omega})$. According to Lemma 3, with probability at least $1-\delta$, we have

$$\begin{aligned}&\sup_{\boldsymbol{\omega}\in\mathcal{W}}(R(\ell^{\rho}\circ\boldsymbol{\omega})-\hat{R}(\ell^{\rho}\circ\boldsymbol{\omega}))\\ &\quad\le 2\mathfrak{R}_p(\ell^{\rho}\circ\mathcal{W})+2\mathfrak{R}_u(\ell^{\rho}\circ\mathcal{W})+\sqrt{\frac{\ln(1/\delta)}{2n}} \quad (39)\end{aligned}$$

where

$$\mathfrak{R}_p(\ell^{\rho}\circ\mathcal{W})=\mathbb{E}\left[\sup_{\boldsymbol{\omega}\in\mathcal{W}}\frac{1}{p}\sum_{i=1}^{p}\sigma_i\ell^{\rho}(\boldsymbol{\omega}^{\top}\boldsymbol{X}_i)\right] \quad (40)$$

and

$$\mathfrak{R}_u(\ell^{\rho}\circ\mathcal{W})=\mathbb{E}\left[\sup_{\boldsymbol{\omega}\in\mathcal{W}}\frac{1}{u}\sum_{i=p+1}^{p+u}\sigma_i\ell^{\rho}(\boldsymbol{\omega}^{\top}\boldsymbol{X}_i)\right]. \quad (41)$$

To upper bound the Rademacher complexities in (40) and (41), we first derive an upper bound for the classifiers in $\mathcal{W}$. Specifically, due to the optimality of any $\boldsymbol{\omega}\in\mathcal{W}$, we have

$$\begin{aligned}&\frac{1}{2}\|\boldsymbol{\omega}\|^2+\frac{\alpha}{p}\sum_{i=1}^{p}[\max(1-\boldsymbol{\omega}^{\top}\mathbf{x}_i,0)]^2\\ &\quad+\frac{\beta}{u}\sum_{i=p+1}^{p+u}e^{-3(\boldsymbol{\omega}^{\top}\mathbf{x}_i)^2}+\frac{\gamma}{u}\sum_{i=p+1}^{p+u}[\max(\Phi(\boldsymbol{\omega}^{\top}\mathbf{x}_i)-t,0)]^2\\ &\le\frac{1}{2}\|\mathbf{0}\|^2+\frac{\alpha}{p}\sum_{i=1}^{p}[\max(1-\mathbf{0}^{\top}\mathbf{x}_i,0)]^2\\ &\quad+\frac{\beta}{u}\sum_{i=p+1}^{p+u}e^{-3(\mathbf{0}^{\top}\mathbf{x}_i)^2}+\frac{\gamma}{u}\sum_{i=p+1}^{p+u}[\max(\Phi(\mathbf{0}^{\top}\mathbf{x}_i)-t,0)]^2\\ &=\alpha+\beta+\gamma t^2. \quad (42)\end{aligned}$$

This implies that $\|\boldsymbol{\omega}\|^2\le 2\alpha+2\beta+2\gamma t^2$.

Now, we are going to upper bound $\mathfrak{R}_p(\ell^{\rho}\circ\mathcal{W})$ and $\mathfrak{R}_u(\ell^{\rho}\circ\mathcal{W})$. Specifically, since the function $\ell^{\rho}(x)$ is $1/\rho$-Lipschtiz, by using Lemma 4, we have

$$\begin{aligned}\mathfrak{R}_p(\ell^{\rho}\circ\mathcal{W}) &\le\frac{1}{\rho}\mathfrak{R}_p(\mathcal{W})\\ &=\frac{1}{\rho}\mathbb{E}\left[\sup_{\boldsymbol{\omega}\in\mathcal{W}}\frac{1}{p}\sum_{i=1}^{p}\sigma_i\boldsymbol{\omega}^{\top}\boldsymbol{X}_i\right]\\ &=\frac{1}{\rho}\mathbb{E}\left[\sup_{\boldsymbol{\omega}\in\mathcal{W}}\left\langle\boldsymbol{\omega},\frac{1}{p}\sum_{i=1}^{p}\sigma_i\boldsymbol{X}_i\right\rangle\right]\\ &\overset{1}{\le}\frac{1}{\rho p}\mathbb{E}\left[\sup_{\boldsymbol{\omega}\in\mathcal{W}}\|\boldsymbol{\omega}\|\left\|\sum_{i=1}^{p}\sigma_i\boldsymbol{X}_i\right\|\right]\\ &\le\frac{\sqrt{2\alpha+2\beta+2\gamma t^2}}{\rho p}\mathbb{E}\left[\left\|\sum_{i=1}^{p}\sigma_i\boldsymbol{X}_i\right\|\right]\\ &\overset{2}{\le}\frac{\sqrt{2\alpha+2\beta+2\gamma t^2}}{\rho p}\sqrt{\sum_{i=1}^{p}\mathbb{E}[\|\boldsymbol{X}_i\|^2]}\\ &\le\frac{b'\sqrt{2(\alpha+\beta+\gamma t^2)}}{\rho\sqrt{p}} \quad (43)\end{aligned}$$

where the Inequality 1 holds because of the Cauchy–Schwarz inequality, and the Inequality 2 holds because of the Jensen's inequality (the square root function is concave) and that $\mathbb{E}[\sigma_i\sigma_j]=0$ when $i\ne j$.

Similarly, we have

$$\mathfrak{R}_u(\ell^{\rho}\circ\mathcal{W})\le\frac{b'\sqrt{2(\alpha+\beta+\gamma t^2)}}{\rho\sqrt{u}}. \quad (44)$$

By combining inequalities (38), (39), (43), and (44), for any learned $\hat{\boldsymbol{\omega}}$, with probability at least $1-\delta$, we have

$$R(\hat{\boldsymbol{\omega}}) - \hat{R}^{\rho}(\hat{\boldsymbol{\omega}}) \leq \frac{2b'\sqrt{2(\alpha+\beta+\gamma t^2)}}{\rho\sqrt{p}} + \frac{2b'\sqrt{2(\alpha+\beta+\gamma t^2)}}{\rho\sqrt{u}} + \sqrt{\frac{\ln(1/\delta)}{2n}} \quad (45)$$

which concludes the proof of Theorem 1. □

Theorem 1 shows that by either increasing the sample size of positive data or unlabeled data, the upper bound of our LLSVM decreases, which justifies the usefulness of PU data in PU learning. Also, when the training sample sizes of the PU data go to infinity, the upper bound will vanish. This also guarantees the generalization ability of the proposed learning algorithm.

V. Experimental Results

In this section, we first study the closeness of the decision boundaries of our LLSVM and fully supervised SVM on a toy data set, and then compare LLSVM with some state-of-the-art PU learning models on benchmark and practical data sets. Finally, we investigate the running time and also parametric sensitivity of the proposed LLSVM.

A. Toy Data

To visualize the effectiveness of our proposed PU learning algorithm LLSVM, we run LLSVM on a synthetic 2-D data set called *DoubleGaussian*. This data set is composed of two data clusters generated from two Gaussian distributions centered at $(0, 0)$ and $(2.5, 2.5)$, respectively (see Fig. 3). The covariance matrix of both Gaussians is $Cov = \left(\begin{smallmatrix} 0.5 & 0 \\ 0 & 0.5 \end{smallmatrix}\right)$. Each Gaussian forms a class with 100 examples and only a part of the positive examples are labeled while the remaining examples are regarded as unlabeled.

We first randomly pick up 30 positive examples (i.e., $p = 30$) which are represented by the red triangles shown in Fig. 3(a), and the ground truth labels of all examples are indicated by different colors shown in Fig. 3(b). Based on the ground truth labels, we adopt the liblinear[1] toolbox to train a standard fully supervised binary SVM, and plot the generated decision boundary in Fig. 3(b) (cyan dashed line). This decision boundary is assumed to be ideal as it is built with the aid of all positive and negative training labels. Furthermore, we conduct our proposed LLSVM on this data set with only 30 known positive examples as shown in Fig. 3(a), and the produced decision boundary is illustrated by the magenta line in Fig. 3(b). After that, we randomly select 20 more positive examples such that the total number of positive examples is 50 [see Fig. 3(c)]. Then, we run the fully supervised SVM and our LLSVM for PU learning again to investigate their performances, and the obtained decision boundaries are visualized in Fig. 3(d).

From the right column of Fig. 3, we see that the decision boundaries of LLSVM are very close to that of SVM when

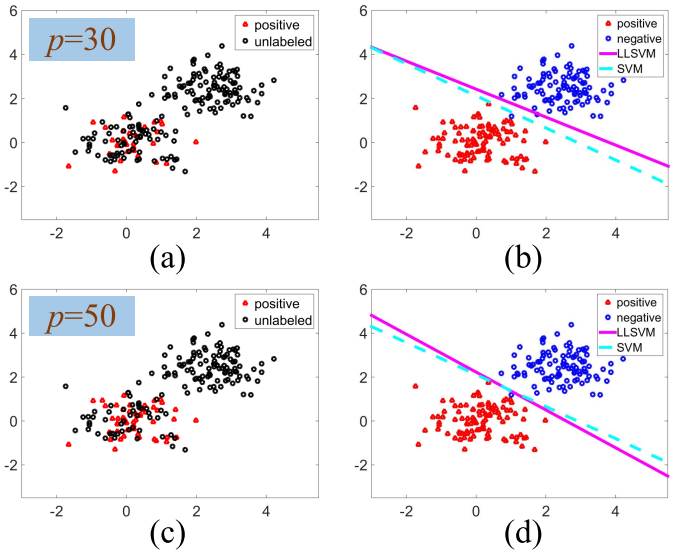

Fig. 3. Experiments on *TwoGaussian* data set. The first row compares the decision boundaries of LLSVM (magenta line) for PU learning with 30 positive examples and the fully supervised binary SVM (cyan dashed line) with all known labels. The second row conducts the same comparison when the proposed LLSVM is trained on 50 positive examples.

TABLE I

Overview of the Adopted Data Sets. $n = p' + n'$ Is the Total Number of Examples, Where p' and n' Are the Real Amounts of Positive Examples and Negative Examples, Respectively; d Denotes the Feature Dimensionality of Every Example

Datasets	n	p'	n'	d
fri	1000	443	557	25
phoneme	5404	1586	3818	5
bank8FM	8192	4885	3307	8
ailerons	13750	5828	7922	40

$p = 30$ and $p = 50$. It means that although our PU algorithm LLSVM is trained without using the negative examples, its performance is still comparable with the ideal decision boundary that is established under a fully supervised case. Therefore, our model is effective and is promising to generate satisfactory performance. Moreover, by comparing Fig. 3(b) and (d), we see that the decision boundary of LLSVM under $l = 50$ is closer to that of standard SVM than $l = 30$, which demonstrates that bringing more positive examples into PU training is beneficial to improving the model performance. This is consistent with our theoretical findings in Section IV, and also suggests that the bound derived in (28) is meaningful.

B. Benchmark Data

In this section, we compare the proposed LLSVM with other representative PU learning methods on OpenML[2] benchmark data sets. Specifically, four binary data sets are adopted for algorithm evaluation including *fri*, *phoneme*, *bank8FM*, and *ailerons*, and their configurations are listed in Table I.

[1] https://www.csie.ntu.edu.tw/čjlin/liblinear/

[2] https://www.openml.org/search?type=data

TABLE II

ACCURACIES OF VARIOUS METHODS ON FOUR OPENML BENCHMARK DATA SETS WHEN 60% AND 90% POSITIVE EXAMPLES ARE LABELED. THE BEST RECORD UNDER EACH NOISE LEVEL IS MARKED IN **BOLD**. "$\surd$" ("$\times$") INDICATES THAT LLSVM IS SIGNIFICANTLY BETTER (WORSE) THAN THE CORRESPONDING METHOD VIA PAIRED T-TEST

Datasets	Methods	60%	90%
fri	WSVM [18]	0.444 ± 0.004 $\surd$	0.560 ± 0.019 $\surd$
	DH [10]	0.570 ± 0.028 $\surd$	0.557 ± 0.003 $\surd$
	NNPU-MLP [12]	0.640 ± 0.056 $\surd$	0.650 ± 0.055
	NNPU-Linear [12]	0.570 ± 0.051 $\surd$	0.620 ± 0.029 $\surd$
	LPM [22]	0.651 ± 0.047 $\surd$	0.619 ± 0.042 $\surd$
	LLSVM (ours)	**0.680 ± 0.056**	**0.661 ± 0.058**
phoneme	WSVM [18]	0.306 ± 0.027 $\surd$	0.735 ± 0.014 $\surd$
	DH [10]	**0.749 ± 0.013**	0.746 ± 0.018 $\surd$
	NNPU-MLP [12]	0.703 ± 0.029 $\surd$	0.745 ± 0.031 $\surd$
	NNPU-Linear [12]	0.393 ± 0.078 $\surd$	0.415 ± 0.050 $\surd$
	LPM [22]	0.706 ± 0.001 $\surd$	0.706 ± 0.819 $\surd$
	LLSVM (ours)	0.735 ± 0.010	**0.777 ± 0.008**
bank8FM	WSVM [18]	0.853 ± 0.005 $\surd$	0.906 ± 0.006 $\surd$
	DH [10]	0.761 ± 0.010 $\surd$	0.726 ± 0.003 $\surd$
	NNPU-MLP [12]	0.886 ± 0.020 $\surd$	0.925 ± 0.011
	NNPU-Linear [12]	0.492 ± 0.095 $\surd$	0.453 ± 0.135 $\surd$
	LPM [22]	0.843 ± 0.028 $\surd$	0.819 ± 0.016 $\surd$
	LLSVM (ours)	**0.905 ± 0.007**	**0.932 ± 0.007**
ailerons	WSVM [18]	0.786 ± 0.014 $\surd$	0.779 ± 0.004 $\surd$
	DH [10]	**0.829 ± 0.045** $\times$	0.782 ± 0.065 $\surd$
	NNPU-MLP [12]	0.764 ± 0.035 $\surd$	0.796 ± 0.022 $\surd$
	NNPU-Linear [12]	0.540 ± 0.029 $\surd$	0.576 ± 0.038 $\surd$
	LPM [22]	0.744 ± 0.031 $\surd$	0.692 ± 0.034 $\surd$
	LLSVM (ours)	0.807 ± 0.013	**0.838 ± 0.006**

The compared methodologies include:

1) Classical method: Weighted SVM (denoted as "WSVM") [18].
2) State-of-the-art methods: Double Hinge loss (denoted as "DH") [10], multilayer perceptron with nonnegative PU risk estimator (denoted as "NNPU-MLP") [12], linear classifier with nonnegative PU risk estimator (denoted as "NNPU-Linear") [12], and large positive margin approach (LPM) [22].

Note that LPM is also developed on the margin theory, so it is incorporated for comparison. For fair comparison, the codes of WSVM, DH, NNPU-MLP, and NNPU-Linear for our experiments are directly provided by the authors, and LPM was implemented by ourselves based on the liblinear toolbox.

For each data set, we randomly treat $r = 60\%$ and $r = 90\%$ positive examples as labeled and leave the remaining 40% and 10% positive examples as well as all negative examples as unlabeled. Under each r, we conduct five-fold cross validation on every compared method and report the average test accuracy and standard deviation over the five independent implementations. As a result, every model under a certain implementation is trained with 80% examples and then tested on the rest 20% examples. Note that the selected positive examples and the data set splits are kept identical for all the compared methodologies. The parameters of every algorithm have been carefully tuned to achieve the best performance. To be specific, the optimal tradeoff parameter C for WSVM is searched from $\{0.01, 0.1, 1, 10, 100\}$ via cross validation. The parameter τ governing the positive margin in LPM is set to the 75% quantile of positive decision values predicted by the initial model according to [22], and the number of queried examples Q in each iteration is adaptively decided as $Q = (3/4)u$ for all data sets. To achieve a fair comparison, the minibatch size of NNPU-MLP and NNPU-Linear for SGD is tuned to the same value as LLSVM, and the class prior π is also estimated via [30] like our LLSVM. For the proposed LLSVM, the tradeoff parameters α, β, and γ in (9) are tuned by searching the grid $\{0.1, 1, 10, 100, 1000\}$, and the amount of mini-batches is kept to $N = 40$ for all the adopted data sets.

The test accuracies of all methods on the four benchmark data sets including *fri*, *phoneme*, *bank8FM*, and *ailerons* are compared in Table II. We also adopt the paired t-test with significance level 0.1 to investigate whether our LLSVM is significantly better than the compared baselines. It can be easily found that our LLSVM is able to obtain higher test accuracy than other baseline methods in most cases. Exceptional cases are that when $r = 60\%$, LLSVM is significantly worse than DH on *ailerons* and is comparable with DH on *phoneme*. Another finding is that the accuracies obtained by the proposed LLSVM are above 70% on *phoneme*, *bank8FM*, and *ailerons*, which suggests that LLSVM generates very

TABLE III

ACCURACIES OF VARIOUS METHODS ON *CIFAR* DATA SET WHEN 60% AND 90% POSITIVE EXAMPLES ARE LABELED. THE BEST RECORD UNDER EACH NOISE LEVEL IS MARKED IN **BOLD**. "$\surd$" INDICATES THAT LLSVM IS SIGNIFICANTLY BETTER THAN THE CORRESPONDING METHOD VIA PAIRED T-TEST

Methods \ r	60%	90%
WSVM [18]	0.883 ± 0.010 $\surd$	0.932 ± 0.005 $\surd$
DH [10]	0.941 ± 0.003 $\surd$	0.935 ± 0.016 $\surd$
NNPU-MLP [12]	0.951 ± 0.006 $\surd$	0.960 ± 0.012
NNPU-Linear [12]	0.935 ± 0.020 $\surd$	0.949 ± 0.013 $\surd$
LPM [22]	0.861 ± 0.020 $\surd$	0.889 ± 0.027 $\surd$
LLSVM (ours)	**0.966 ± 0.005**	**0.970 ± 0.004**

TABLE IV

ACCURACIES OF VARIOUS METHODS ON *GermanCredit* DATA SET WHEN 60% AND 90% POSITIVE EXAMPLES ARE LABELED. THE BEST RECORD UNDER EACH NOISE LEVEL IS MARKED IN **BOLD**. "$\surd$" INDICATES THAT LLSVM IS SIGNIFICANTLY BETTER THAN THE CORRESPONDING METHOD VIA PAIRED T-TEST

Methods \ r	60%	90%
WSVM [18]	0.703 ± 0.015 $\surd$	0.736 ± 0.010 $\surd$
DH [10]	0.717 ± 0.019 $\surd$	0.730 ± 0.038 $\surd$
NNPU-MLP [12]	0.659 ± 0.101 $\surd$	0.690 ± 0.080 $\surd$
NNPU-Linear [12]	0.553 ± 0.028 $\surd$	0.620 ± 0.035 $\surd$
LPM [22]	0.702 ± 0.004 $\surd$	0.701 ± 0.008 $\surd$
LLSVM (ours)	**0.739 ± 0.027**	**0.760 ± 0.026**

impressive classification results, although it is trained without the help of negative examples. Specifically, our LLSVM touches the accuracy of 83.8% on *ailerons* when $r = 90\%$, which leads the second best method NNPU-MLP with a significant margin of 4.2%.

C. Image Data

In this section, we investigate the performance of WSVM, DH, NNPU-MLP, NNPU-Linear, LPM, and LLSVM on the task of natural image classification. To this end, we extract the images of "cat" and "dog", which are usually difficult to discriminate, from the *CIFAR10* data set [33], [34], and target to classify every image example into one of the above two classes. In this data set, each class has 6000 images and we use the output of the first fully connected layer of VGGNet-16 [35] to form a 4096-dimensional feature vector to express each image example. Similar to the experiments in Section V-B, we also conduct five-fold cross validation on all compared methods and record their mean test accuracies and standard deviations. Furthermore, we adopt the paired t-test with significance level 0.1 to investigate whether our LLSVM is significantly better than the compared baselines.

The tradeoff parameter C in WSVM is tuned to 0.01 to obtain the optimal results. The number of minibatches N for NNPU-MLP, NNPU-Linear, and LLSVM is set to 40. The results obtained by various algorithms are presented in Table III, which suggest that our LLSVM achieves the highest classification accuracy among all methodologies when 60% and 90% positive examples are in the labeled positive set $\mathcal{P}$. Such advantage of LLSVM to other baselines has been statistically confirmed by the paired t-test unless to the NNPU-MLP method when $r = 90\%$. In this case, our LLSVM and NNPU-MLP achieve comparable performance. Another notable fact is that LLSVM achieves more than 96% accuracy on this data set, reflecting that the discriminability of LLSVM is very encouraging. Comparatively, LPM performs unsatisfactorily on this data set of which the accuracy is always below 90%. Therefore, the proposed LLSVM is effective in handling image data.

D. Banking Data

Evaluating a customer's credit is very important for a bank to make a decision whether he/she is qualified to hold a credit card or apply for a loan. Usually, the credit evaluation system of a bank can automatically specify a fraction of customers whose credits are low. However, it is not true that the customers not "caught" by the system are always good, which means that some customers with low credits might also pass the examination of the credit evaluation system. Therefore, by treating the "bad" customers identified by the system as positive examples and the remaining customers as unlabeled examples, we can use PU learning algorithms to distinguish bad guys from good guys.

Specifically, we use the *GermanCredit* data set [36] for our experiment. This data set provides the personal information of totally 1000 German such as credit history, personal property, employment condition, marital status, and so on. Therefore, the task of compared methods such as WSVM, DH, NNPU-MLP, NNPU-Linear, LPM, and LLSVM is to identify the credit of every customer as good or bad. Note that the *GermanCredit* data set contains 300 positive examples (i.e., bad guys) and 700 negative examples (i.e., good guys), which is often the case in practical situations that the subjects with low credit are much less than the normal ones. Such data imbalance also poses a great difficulty for accurate classification. The parameters in our LLSVM are set to $\alpha = 1$, $\beta = 0.01$, and $\gamma = 1000$ via cross validation, and the number of minibatches is also kept to 40 as above. The parameter C in WSVM and the γ in NNPU-MLP and NNPU-Linear have been tuned to the optimal value 1. For LPM, the number queried examples in each iteration is decided as $Q = 90$ to achieve the best performance.

Similar to the experiments in Sections V-B and V-C, we investigate the performances of all compared methods when $r = 60\%$ and $r = 90\%$ positive examples are labeled. The results of these comparators are produced by five-fold cross validation and the mean test accuracies of their five independent runs are reported in Table IV. In addition, we also use the t-test to statistically validate the superiority of our LLSVM to the remaining baselines. From the table, we see that LLSVM touches the highest test accuracy among all methods when $r = 60\%$ and $r = 90\%$, and this advantage has also been verified by the t-test. WSVM and DH are usually in the second place while NNPU-MLP and NNPU-Linear perform unsatisfactorily on this data set. Therefore, our LLSVM shows very

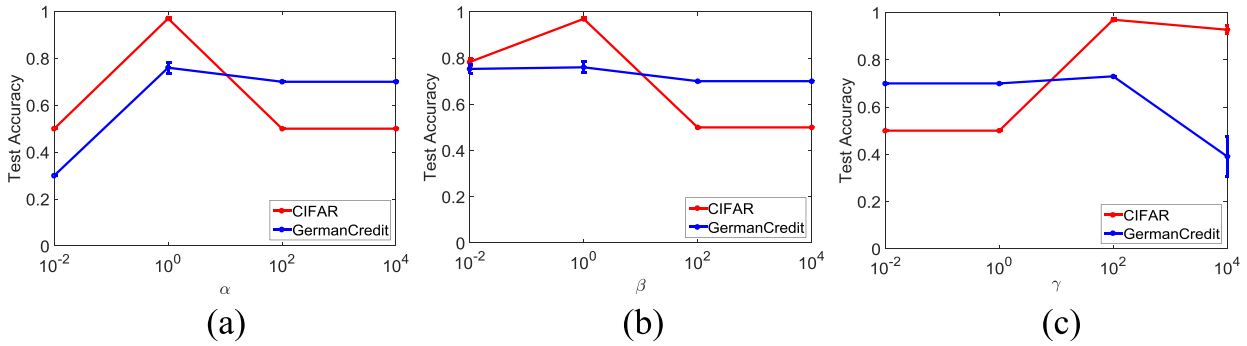

Fig. 4. Parametric sensitivity of (a) α, (b) β, and (c) γ of our LLSVM on *CIFAR* and *GermanCredit* data sets.

TABLE V
CPU TIME (IN SECONDS) OF VARIOUS METHODS ON *CIFAR* DATA SET

	duration	code type
WSVM [18]	1605.80	Python
DH [10]	32.60	Matlab
NNPU-MLP [12]	382.22	Python
NNPU-Linear [12]	402.28	Python
LPM [22]	23.43	Matlab
LLSVM (ours)	37.05	Matlab

TABLE VI
CPU TIME (IN SECONDS) OF VARIOUS METHODS ON *GermanCredit* DATA SET

	duration	code type
WSVM [18]	0.20	Python
DH [10]	0.17	Matlab
NNPU-MLP [12]	1.12	Python
NNPU-Linear [12]	0.32	Python
LPM [22]	0.04	Matlab
LLSVM (ours)	0.14	Matlab

encouraging performance in handling the real-world banking data.

E. Running Time

In Section IV-A, we analyzed the complexity for solving our LLSVM model and the results indicate that the related optimization problem can be efficiently solved via SGD. In this section, we compare the training time of our method with WSVM, DH, NNPU-MLP, NNPU-Linear, and LPM on the two adopted real-world data sets including *CIFAR* and *GermanCredit*. Specifically, all algorithms are implemented on a desktop with Intel i7-6700 CPU at 3.40 GHz and 16-G memory, and their CPU time is compared when $r = 60\%$ positive examples are labeled on each data set.

The running time of the compared methods on *CIFAR* and *GermanCredit* data sets are recorded in Tables V and VI, respectively. We see that LPM is usually more efficient than our LLSVM, and the time costs of DH and our LLSVM are also comparable, but their classification accuracies are much lower than those of our LLSVM as revealed by Tables III and IV. In addition, NNPU-MLP, NNPU-Linear, and WSVM usually require more computational time than LLSVM. Therefore, we conclude that the proposed LLSVM is able to complete the model training under an acceptable time cost.

F. Parametric Sensitivity

The objective function (9) of our LLSVM model contains three tradeoff parameters α, β, and γ that should be pretuned manually. Therefore, in this section, we study their influences on the final performance of LLSVM. To this end, we examine the test accuracy of LLSVM by varying one of α, β, and γ, and meanwhile fixing every remaining parameter to a constant value [37], [38]. The two practical data sets from Sections V-C and V-D are adopted including *CIFAR* and *GermanCredit*. By changing these three parameters from 10^{-2} to 10^4, the results on the two practical data sets when $r = 90\%$ positive examples are labeled are shown in Fig. 4.

From the curves presented in Fig. 4, we find that these three parameters are critical for our LLSVM to achieve good performance. To be specific, Fig. 4(a) and (b) shows that $\alpha = 1$ and $\beta = 1$ usually lead to high classification accuracy, and therefore, these two parameters are consistently set to 1 throughout our experiments. In contrast, Fig. 4(c) reveals that the optimal γ should be chosen around 100, which is generally consistent with its setting 100 and 1000 on the real-world *CIFAR* and *GermanCredit* data sets.

VI. CONCLUSION

In this paper, we proposed a novel PU learning algorithm named LLSVM. To enable our LLSVM to generate discriminative decision function with the absence of negative examples, three critical regularizers are introduced, i.e., the hinge loss for fitting the positive examples, the hat loss for achieving the max-margin effect, and the label calibration term for calibrating the biased decision boundary to the potentially correct one. The proposed optimization problem can be efficiently solved via minibatch SGD, and its generalization bound has also been theoretically proved. We compared our LLSVM with state-of-the-art PU methods on both synthetic and real-world data sets, and the results suggest that LLSVM is superior to other compared baseline methods in most cases.

Since the performance of LLSVM is a bit sensitive to the selections of α, β, and γ, we plan to find a suitable way to adaptively determine them in the future. In addition, it is also worthwhile to apply the proposed method to solving more practical problems.

REFERENCES

[1] H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao, "Spotting fake reviews via collective positive-unlabeled learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2014, pp. 899–904.

[2] W. Li, Q. Guo, and C. Elkan, "A positive and unlabeled learning algorithm for one-class classification of remote-sensing data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 2, pp. 717–725, Feb. 2011.

[3] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 2, Jul. 2002, pp. 387–394.

[4] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu, "Building text classifiers using positive and unlabeled examples," in *Proc. 3rd IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2003, pp. 179–186.

[5] B. Frénay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014.

[6] B. Han, I. W. Tsang, L. Chen, C. P. Yu, and S.-F. Fung, "Progressive stochastic learning for noisy labels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 5136–5148, Oct. 2018.

[7] B. Han *et al.*, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 8535–8545.

[8] H. Shi, S. Pan, J. Yang, and C. Gong, "Positive and unlabeled learning via loss decomposition and centroid estimation," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, vol. 1, no. 1, pp. 1–7.

[9] W. S. Lee and B. Liu, "Learning with positive and unlabeled examples using weighted logistic regression," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, vol. 3, Aug. 2003, pp. 448–455.

[10] M. C. du Plessis, G. Niu, and M. Sugiyama, "Convex formulation for learning from positive and unlabeled data," in *Proc. 32nd Int. Conf. Mach. Learn. (PMLR)*, Jun. 2015, pp. 1386–1394.

[11] M. C. du Plessis, G. Niu, and M. Sugiyama, "Analysis of learning from positive and unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 703–711.

[12] R. Kiryo, G. Niu, M. Plessis, and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1674–1684.

[13] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*. San Rafael, CA, USA: Morgan & Claypool, 2009.

[14] T. Durand, N. Thome, and M. Cord, "SyMIL: MinMax latent SVM for weakly labeled data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28. no. 12, pp. 6099–6112, Dec. 2018.

[15] Z. Ma, X. Chang, Y. Yang, N. Sebe, and A. G. Hauptmann, "The many shades of negativity," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1558–1568, Jul. 2017.

[16] Y. Xiao, B. Liu, J. Yin, L. Cao, C. Zhang, and Z. Hao, "Similarity-based approach for positive and unlabeled learning," in *Proc. 22nd Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2011, vol. 22, no. 1, pp. 1577–1582.

[17] M. Luo, X. Chang, Z. Li, L. Nie, A. G. Hauptmann, and Q. Zheng, "Simple to complex cross-modal learning to rank," *Comput. Vis. Image Understand.*, vol. 163, pp. 67–77, Oct. 2017.

[18] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Aug. 2008, pp. 213–220.

[19] G. Niu, M. C. Plessis, T. Sakai, Y. Ma, and M. Sugiyama, "Theoretical comparisons of positive-unlabeled learning against positive-negative learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 1199–1207.

[20] J. T. Zhou, S. Pan, Q. Mao, and I. W. Tsang, "Multi-view positive and unlabeled learning," in *Proc. Asian Conf. Mach. Learn. (ACML)*, Nov. 2012, pp. 555–570.

[21] Y. Xu, C. Xu, C. Xu, and D. Tao, "Multi-positive and unlabeled learning," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2017, pp. 3182–3188.

[22] T. Gong, G. Wang, J. Ye, Z. Xu, and M. Lin, "Margin based PU learning," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, Apr. 2018, pp. 1–8.

[23] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, Jan. 2005, pp. 57–64.

[24] C. Gong, T. Liu, Y. Tang, J. Yang, J. Yang, and D. Tao, "A regularization approach for instance-based superset label learning," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 967–978, Mar. 2018.

[25] J. Bekker and J. Davis, "Estimating the class prior in positive and unlabeled data through decision tree induction," in *Proc. 32nd Conf. Artif. Intell. (AAAI)*, Apr. 2018, pp. 2712–2719.

[26] S. Jain, M. White, and P. Radivojac, "Estimating the class prior and posterior from noisy positives and unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 2693–2701.

[27] A. Iyer, S. Nath, and S. Sarawagi, "Maximum mean discrepancy for class ratio estimation: Convergence bounds and Kernel selection," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jan. 2014, pp. 530–538.

[28] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, "Domain adaptation with conditional transferable components," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jun. 2016, pp. 2839–2848.

[29] H. Ramaswamy, C. Scott, and A. Tewari, "Mixture proportion estimation via kernel embeddings of distributions," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jun. 2016, pp. 2052–2060.

[30] M. Plessis, G. Niu, and M. Sugiyama, "Class-prior estimation for learning from positive and unlabeled data," in *Proc. Asian Conf. Mach. Learn. (ACML)*, Feb. 2015, pp. 221–236.

[31] A. Keziou, "Dual representation of φ-divergences and applications Représentation duale des φ-divergences et applications," *Comp. Rendus Mathematique*, vol. 336, no. 10, pp. 857–862, May 2003.

[32] P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *J. Mach. Learn. Res.*, vol. 3, pp. 463–482, Nov. 2002.

[33] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009, vol. 1, no. 4.

[34] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang, "Multi-modal curriculum learning for semi-supervised image classification," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3249–3260, Jul. 2016.

[35] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[36] D. Dheeru and E. Karra, *UCI Machine Learning Repository*. 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[37] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed graph Laplacian for semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2261–2274, Oct. 2015.

[38] C. Gong, D. Tao, W. Liu, L. Liu, and J. Yang, "Label propagation via teaching-to-learn and learning-to-teach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1452–1465, Jun. 2017.

Chen Gong (M'16) received the B.E. degree from the East China University of Science and Technology, Shanghai, China, in 2010, and the dual doctoral degree from the Shanghai Jiao Tong University (SJTU), Shanghai, and the University of Technology Sydney, Sydney, Australia, in 2016 and 2017, respectively.

He is currently a Full Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. He has published more than 50 technical papers at prominent journals and conferences, such as the IEEE T-NNLS, IEEE T-IP, IEEE T-CYB, IEEE T-CSVT, IEEE T-MM, IEEE T-ITS, CVPR, AAAI, IJCAI, ICDM, and so on. His research interests mainly include machine learning, data mining, and learning-based vision problems.

Dr. Gong serves as a PC member for several top-tier conferences such as ICML, AAAI, IJCAI, ICDM, AISTATS, and so on. He received the Excellent Doctoral Dissertation awarded by SJTU and the Chinese Association for Artificial Intelligence. He was also enrolled by the "Summit of the Six Top Talents" Program of Jiangsu Province, China, and the "Lift Program for Young Talents" of the China Association for Science and Technology. He also serves as a Reviewer for over 20 international journals such as the *AIJ*, IEEE T-NNLS, and IEEE T-IP.

Tongliang Liu received the B.E. degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, and the Ph.D. degree in information systems from the University of Technology Sydney, Ultimo, NSW, Australia.

He is currently a Lecturer with the School of Computer Science, Faculty of Engineering and Information Technologies, The University of Sydney, Sydney, NSW, Australia. He is also a Core Member of the UBTECH Sydney AI Centre, The University of Sydney. He has authored or co-authored over 40 research papers, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, ICML, CVPR, and KDD. His current research interests include statistical learning theory, machine learning, computer vision, and optimization.

Jian Yang (M'06) received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002.

In 2003, he was a Post-Doctoral Researcher with the University of Zaragoza, Zaragoza, Spain. From 2004 to 2006, he was a Post-Doctoral Fellow with the Biometrics Center, Hong Kong Polytechnic University, Hong Kong. From 2006 to 2007, he was a Post-Doctoral Fellow with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Chang-Jiang Professor with the School of Computer Science and Technology, NUST. He has authored more than 200 scientific papers in pattern recognition and computer vision. His papers have been cited more than 5000 times in the Web of Science and 13000 times in the Google Scholar. His current research interests include pattern recognition, computer vision, and machine learning.

Dr. Yang is a fellow of IAPR. He is /was an Associate Editor of *Pattern Recognition*, *Pattern Recognition Letters*, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Neurocomputing*.

Dacheng Tao (F'15) was a Professor of computer science and the Director of Center for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW, Australia. He is currently a Professor of computer science with the School of Information Technologies, Faculty of Engineering and Information Technologies, The University of Sydney, Sydney, NSW, Australia. He is involved in the application of statistics and mathematics to Artificial Intelligence and data science. His current research interests include computer vision, data science, image processing, machine learning, and video surveillance. His research results have expounded in one monograph and more than 200 publications at prestigious journals and prominent conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, *Journal of Machine Learning Research*, *International Journal of Computer Vision*, NIPS, ICML, CVPR, ICCV, ECCV, AISTATS, ICDM, and ACM SIGKDD.

Mr. Tao is a fellow of the Australian Academy of Science, AAAS, IAPR, OSA, and SPIE. He was a recipient of several best paper awards, including the Best Theory/Algorithm Paper Runner Up Award in IEEE ICDM'07, the Best Student Paper Award in IEEE ICDM'13, the 2014 ICDM 10-Year Highest-Impact Paper Award, the 2015 Australian Scopus-Eureka Prize, the 2015 ACS Gold Disruptor Award, and the 2015 University of Technology Sydney Vice-Chancellor's Medal for Exceptional Research.