Instance-Dependent Positive and Unlabeled Learning with Labeling Bias Estimation

Chen Gong, *Member, IEEE,* Qizhou Wang, Tongliang Liu, *Member, IEEE,*
Bo Han, Jane You, Jian Yang, *Member, IEEE,* Dacheng Tao, *Fellow, IEEE*

Abstract—This paper studies instance-dependent Positive and Unlabeled (PU) classification, where whether a positive example will be labeled (indicated by s) is not only related to the class label y, but also depends on the observation $\mathbf{x}$. Therefore, the labeling probability on positive examples is not uniform as previous works assumed, but is biased to some simple or critical data points. To depict the above dependency relationship, a graphical model is built in this paper which further leads to a maximization problem on the induced likelihood function regarding $P(s,y|\mathbf{x})$. By utilizing the well-known EM and Adam optimization techniques, the labeling probability of any positive example $P(s=1|y=1,\mathbf{x})$ as well as the classifier induced by $P(y|\mathbf{x})$ can be acquired. Theoretically, we prove that the critical solution always exists, and is locally unique for linear model if some sufficient conditions are met. Moreover, we upper bound the generalization error for both linear logistic and non-linear network instantiations of our algorithm, with the convergence rate of expected risk to empirical risk as $\mathcal{O}(1/\sqrt{k}+1/\sqrt{n-k}+1/\sqrt{n})$ (k and n are the sizes of positive set and the entire training set, respectively). Empirically, we compare our method with state-of-the-art instance-independent and instance-dependent PU algorithms on a wide range of synthetic, benchmark and real-world datasets, and the experimental results firmly demonstrate the advantage of the proposed method over the existing PU approaches.

Index Terms—Instance-Dependent PU Learning, Labeling Bias, Maximum Likelihood Estimation, Solution Uniqueness, Generalization Bound.

1 Introduction

The recent years have witnessed a surge of research interest in Positive and Unlabeled learning (*i.e.*, PU learning) [1], [2], [3], [4], [5], [6], [7], [8], [9], of which the target is to find a suitable classifier simply based on a set of positive and unlabeled training data. Note that each of the unlabeled data here can be positive or negative, but its groundtruth label is undiscovered during the training stage. Up to now, PU learning has found its wide application in various fields such as visual anomaly detection [10], [11], disease gene identification [12], hyperspectral image classification [13], [14], etc.

Existing PU learning methodologies usually follow two different settings, one is "case-control PU learning" [15], and the other is "single-training-set PU learning" [16]. Given $\mathbf{x}\in\mathbb{R}^d$ (d is the dimensionality) as the input random variable in the feature space $\mathcal{X}$ and $y\in\mathbb{R}$ be the output random variable in the binary label space $\mathcal{Y}=\{1,0\}$ ("1" denotes positive class, and "0" represents negative class), the class-conditional density on positive data and the marginal density of $\mathbf{x}$ are respectively $P(\mathbf{x}|y=1)$ and $P(\mathbf{x})$, where $P(\cdot)$ denotes the probability throughout this paper. Case-control PU learning assumes that the positive set $S_P=\{\mathbf{x}_i\}_{i=1}^k$ that consists of k positive examples is independently and identically generated from the conditional distribution $P(\mathbf{x}|y=1)$, and the unlabeled set $S_U=\{\mathbf{x}_i\}_{i=k+1}^n$ that contains $n-k$ unlabeled examples is independently and identically generated from the marginal distribution $P(\mathbf{x})$, where n is the total number of positive and unlabeled examples in the training set $S=S_P\cup S_U$.

In contrast to the case-control PU learning which follows a two-sample setting, single-training-set PU learning follows a one-sample setting which simply assumes that all n examples in $S=\{\mathbf{x}_i\}_{i=1}^n$ are randomly drawn from $P(\mathbf{x})$. After that, if the hidden groundtruth label of $\mathbf{x}_i$ (*i.e.*, y_i) is 1, its label is observed with probability η (*i.e.*, $\mathbf{x}_i\in S_P$), and remains undisclosed with probability $1-\eta$ (*i.e.*, $\mathbf{x}_i\in S_U$). If the groundtruth label of $\mathbf{x}_i$ is 0, its label will never be observed and it belongs to S_U with probability 1. The single-training-set PU methods can be regarded

- *This work was supported by NSF of China (Nos: 61973162, U1713208, 62006202), the Fundamental Research Funds for the Central Universities (No: 30920032202), CCF-Tencent Open Fund (No: RAGR20200101), the "Young Elite Scientists Sponsorship Program" by CAST (No: 2018QNR-C001), Hong Kong Scholars Program (No: XJ2019036), "111 Program" (No: B13022), the RGC Early Career Scheme (No. 22200720), HKBU CSD Departmental Incentive Grant, the Hong Kong Polytechnic University grants (Nos: YZ3K, UAJP/UAGK, and ZVRH), and the ARC (No: DE190101473).*

- *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 Department of Computing, Hong Kong Polytechnic University, Hong Kong, P.R. China.*
  E-mail: chen.gong@njust.edu.cn
- *Q. Wang and B. Han are with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, P.R. China.*
  E-mail: qizhouwang.nanjing@gmail.com; bhanml@comp.hkbu.edu.hk
- *T. Liu is with the Trustworthy Machine Learning Lab, the University of Sydney, 6 Cleveland St, Darlington, NSW 2008, Australia.*
  E-mail: tongliang.liu@sydney.edu.au
- *J. You is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, P.R. China.*
  E-mail: jane.you@polyu.edu.hk
- *J. Yang is with the PCA Lab, the Jiangsu Key Laboratory of Image and Video Understanding for Social Security, and the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, P.R. China.*
  E-mail: csjyang@njust.edu.cn
- *D. Tao is with JD Explore Academy at JD.com, Beijing, P.R. China.*
  E-mail: dacheng.tao@jd.com
- *Corresponding authors: J. Yang and J. You.*

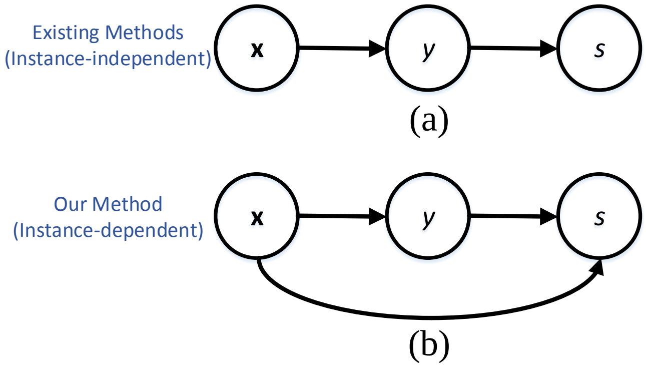

Fig. 1: Setting comparison of our PU method and existing methods. (a) shows the existing instance-independent models, and (b) illustrates our instance-dependent setting.

as a special case of class-conditional label noise learning problem [17] by treating the unlabeled data as noisy negative examples. In other words, the examples in S_U with groundtruth label 1 are mistakenly labeled as negative, while the labels of $\mathbf{x} \in S_P$ are all clean.

Although a certain amount of methods [1], [3], [4], [5], [6], [18] belonging to the above two settings have been developed for PU learning, they all assume that whether a positive data will be labeled is irrelevant to its feature, so every positive data receives an equal probability η to be observed. In other words, if we use a random variable $s = \{0, 1\}$ to indicate whether an example $\mathbf{x}$ is observed as positive (*i.e.*, $s = 1$) or not (*i.e.*, $s = 0$), the existing methods are all built on the fact that $P(s = 1|y = 1, \mathbf{x}) = P(s = 1|y = 1) = \eta$ (see Fig. 1(a)). However, this is not true in real-world problems as the labeling bias often exists during the practical labeling process [19], [20]. For example, the doctors are more likely to annotate the CT images that they are sure about the result in medical diagnosis, and the annotators prefer to label the objects that they are familiar with in crowdsourcing scenario. That is to say, the positive examples in PU learning should not have an equal probability η to be labeled, and whether a positive example will be labeled should not only depend on its label y, but also depend on its feature representation $\mathbf{x}$. This fact is mathematically depicted as $P(s = 1|y = 1, \mathbf{x}) \neq P(s = 1|y = 1)$ and $P(s = 1|y = 1, \mathbf{x}) = \eta(\mathbf{x})$ where $\eta(\mathbf{x})$ is a value related to $\mathbf{x}$. Therefore, this paper aims to study the instance-dependent PU learning[1] with a labeling bias on positive data.

Due to the dependency of $\eta(\mathbf{x})$ on $\mathbf{x}$ in instance-dependent PU learning, the probability estimation in our problem is much more difficult than that in the conventional instance-independent setting illustrated in Fig. 1(a). Concretely, if η is a constant and is irrelevant to $\mathbf{x}$, it can be easily estimated by $\eta = P(s = 1|y = 1) = \frac{P(s=1,y=1)}{P(y=1)} = \frac{P(s=1)}{P(y=1)}$ where $P(s = 1)$ and $P(y = 1)$ can be directly estimated from data [16], [24], [25]. However, if every $\mathbf{x}$ has its own $\eta(\mathbf{x})$ as investigated in this paper, we have $\eta(\mathbf{x}) = P(s = 1|y = 1, \mathbf{x}) = \frac{P(s=1,y=1|\mathbf{x})}{P(y=1|\mathbf{x})} = \frac{P(s=1|\mathbf{x})}{P(y=1|\mathbf{x})}$, from which we observe that the value of $\eta(\mathbf{x})$ and the class posterior probability $P(y = 1|\mathbf{x})$ co-occur. Therefore, we need to find a new way to jointly estimate these two probabilities.

In this paper, we present a probabilistic approach named "Labeling Bias Estimation" (LBE) via graphical model to explicitly establish the relationship among the input feature variable $\mathbf{x} \in \mathbb{R}^d$, groundtruth label $y \in \{0, 1\}$, and labeling condition $s \in \{0, 1\}$ (see Fig. 1(b)), from which we see that the generation process of biased positive data can be clearly described. Notably, the groundtruth label y is related to feature $\mathbf{x}$, and the labeling situation s is conditioned on both $\mathbf{x}$ and y. Given $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ respectively being the parameters of score function $P(y = 1|\mathbf{x}; \boldsymbol{\theta}_1)$ and labeling model $\eta(\mathbf{x}; \boldsymbol{\theta}_2)$, they can be easily estimated by the method of Maximum Likelihood Estimation (MLE). Specifically, the joint conditional probability $P(y, s|\mathbf{x}; \boldsymbol{\theta})$ with $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ can be maximized by using the Expectation Maximization (EM) algorithm, and then the parameters $\boldsymbol{\theta}$ can be easily identified. In our LBE, both the score function $P(y = 1|\mathbf{x}; \boldsymbol{\theta}_1)$ and the labeling probability $\eta(\mathbf{x}; \boldsymbol{\theta}_2)$ can be instantiated by different models according to different practical requirements of users. In this paper, we present a non-deep and a deep implementations of our LBE, where the non-deep model formats $P(y = 1|\mathbf{x}; \boldsymbol{\theta}_1)$ and $\eta(\mathbf{x}; \boldsymbol{\theta}_2)$ as a linear-in-parameter Logistic Function (denoted "LBE-LF"), and the non-linear deep model employs a Multi-Layer Perceptron for realizing $P(y = 1|\mathbf{x}; \boldsymbol{\theta}_1)$ and $\eta(\mathbf{x}; \boldsymbol{\theta}_2)$ (denoted "LBE-MLP"). Theoretically, we reveal that our model can be regarded as a rectified Logistic regression governed by $\eta(\mathbf{x}; \boldsymbol{\theta}_2)$, and the performance of our model will approach to that of a fully-supervised classifier when $\eta(\mathbf{x}; \boldsymbol{\theta}_2) \to 1$. Besides, the existence and uniqueness of the solution yielded by MLE in our method are proved, which demonstrates the validity of the obtained model. Furthermore, the generalization error of LBE is also theoretically proved, which suggests that our method can achieve accurate classifications on unseen test examples if $\eta(\mathbf{x}; \boldsymbol{\theta}_2)$ is accurately estimated. Intensive experimental results on both synthetic and real-world datasets indicate that our LBE is very effective in dealing with the instance-dependent PU learning with a labeling bias.

2 Related Work

As a new branch of weakly-supervised learning [26], PU learning has drawn intensive research interests in recent years. Most of the existing PU models can be categorized into two types, namely "case-control PU learning" and "single-training-set PU learning", according to the assumptions on how the unlabeled data are generated.

Case-control PU learning [15] assumes that the positive data and unlabeled data are generated from $P(\mathbf{x}|y = 1)$ and $P(\mathbf{x})$, respectively. Specifically, Liu *et al.* [1] propose a "spy" technique which inserts a fraction of positive data into the unlabeled set to identify some definite negative examples. By employing the spy technique as the first step, Liu *et al.* [27] further design a two-step method in which a biased SVM with different penalty weights on positive and negative classes is specifically devised. A similar two-step method can also be found in [28] which utilizes the Rocchio method [29] to pick up the reliable negative examples and then uses a traditional SVM for the subsequent classification. However, the identification of negative examples in above methods can be inaccurate, which may heavily degrade the performance in some practical situations. Therefore, recent works mainly focus on design various unbiased or consistent loss functions to resist the negative impact of the absence of negative training data. For example, [30] shows that the conventional loss function such as

[1] Instance-dependent PU learning is also known as "Selected At Random" (SAR) setting in some prior works such as [21]. In this paper, we follow [22], [23] and use the term "instance-dependent PU learning" to refer to our problem.

hinge loss will bring about incorrect decision boundary, and then reveals that the non-convex loss function such as ramp loss is helpful for PU learning. After that, du Plessis *et al.* [3] discovered that the convex risk estimator can also be applied to PU learning as long as we use different loss functions for positive and unlabeled samples. Based on this finding, they design a double hinge loss which is convex and is also statistically unbiased to the loss value under fully supervised case. However, the empirical version of the theoretically-sound loss function in [3] may be negative and lead to the overfitting problem, so [4] makes an improvement on [3] which requires the loss value to be nonnegative. Differently, Hou *et al.* [31] made the first attempt to deal with PU learning via generative adversarial networks. However, the implementation of case-control PU learning often needs to pre-estimate the class prior $P(y=1)$, which is very difficult under PU data. Although several methods [24], [25], [32] have been proposed to make such estimation, the effect is often far from perfect in real-world situations, especially when the data dimensionality is high.

Single-training-set PU learning assumes that both the positive data and unlabeled data are generated from $P(\mathbf{x})$, in which the labels of a set of originally positive examples are covered. Therefore, one common way to tackling single-training-set PU learning is to regard the unlabeled set as noisy negative set with false negative examples, and then transform PU learning as a one-sided label noise learning problem. For example, Lee *et al.* [18] firstly treat all unlabeled data as negative, and then develop a weighted logistic regression in which the weights are selected from a validation set to reduce the disturbance of noisy labels. Similar idea can also be found in [16], where Elkan *et al.* propose a weighted SVM methodology and decide the weights based on the principle of "selected completely at random". Inspired by [33], [5] and [6] decompose the upper bound of the traditional hinge loss into a label-independent term and a label-dependent term, where only the latter is influenced by the label noise. Furthermore, they find an unbiased estimate of the label-dependent term by exploiting the centroid of unlabeled set, therefore the centroid and classifier parameters can be jointly estimated. Recently, Li *et al.* [34] employ reinforcement learning to jointly estimate the label noise rate and classifier parameter.

Other typical PU learning works include [7] established on label calibration, [14] based on multi-manifold data structure, [8] utilizing label disambiguation, [35] based on positive margin shrinkage, etc. A more thorough literature review on PU learning can be found in the surveys [36], [37]. As mentioned in the Introduction, the PU models mentioned above did not consider the labeling bias and thus are instance-independent. Therefore, recently there are some preliminary works that investigate the instance-dependent PU learning problem, *e.g.*, [21], [22], [38]. Among them, [38] is a case-control PU learning algorithm. To make the class prior $P(y=1|\mathbf{x})$ partially identifiable, the authors of [38] introduce the assumption of "invariance of order" which means that $P(s=1|y=1,\mathbf{x})$ and $P(y=1|\mathbf{x})$ induce the same ordering on the input space. Different from [38], [21] and [22] follow the single-training-set PU learning setting. Specifically, [21] considers that whether a positive example will be labeled depends on a subset of "propensity features", and [22] defines a "probabilistic gap" and then relates the likelihood of an example being labeled to such probabilistic gap.

In contrast, our work assumes that the labeling probability $\eta(\mathbf{x};\boldsymbol{\theta}_2)$ is less than 0.5 for $\mathbf{x}\in S_U$, so that the labeling probability $\eta(\mathbf{x};\boldsymbol{\theta}_2)$ and classifier parameter $\boldsymbol{\theta}_1$ can be directly learned from the given PU data. Regarding this aspect, our work is similar to [21], as both of them introduce the labeling probability $\eta(\mathbf{x};\boldsymbol{\theta}_2)$ (a.k.a. propensity score in [21]), and also contain two unknown variables which are optimized via the iterative EM method. However, there are also some differences between them. Firstly, [21] presents a propensity-weighted estimator which weights every example by using the propensity score, while our method starts from a graphical model to describe the data generation process regarding the variables $\mathbf{x}$, y and s, and then maximize the resulting likelihood function. Secondly, since different models are developed, the specific formulations for E step and M step in solving the model are also different.

It is also worth mentioning that our algorithm is designed under the single-training-set PU learning framework, and the main reasons are two-fold. Firstly, as suggested by [38], the case-control PU setting usually needs an assumption of "invariance of order" such that some probabilities are identifiable. When we do not have any domain knowledge on a practical task, this assumption is sometimes strong and may be inconsistent with the underlying labeling mechanism. Secondly, the single-training-set PU learning setting provides an easy way to define and model the positive data labeling condition (*i.e.*, the variable s) which is the main focus of our work.

3 The Proposed Method

This section explains our proposed LBE algorithm in a detailed way, which includes the general graphical model construction (Section 3.1), model instantiations (Section 3.2), and parameter learning (Section 3.3). In our method, each training datum is represented by a triplet $(\mathbf{x},y,s)$, where $\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^d$, $y\in\mathcal{Y}\subset\{0,1\}$ and $s\in\{0,1\}$ have been defined in the Introduction. Therefore, the entire training set that consists of n training data can be represented by $S=\{S_P;S_U\}=\{(\mathbf{x}_1,y_1,s_1),\cdots,(\mathbf{x}_k,y_k,s_k);(\mathbf{x}_{k+1},y_{k+1},s_{k+1}),\cdots,(\mathbf{x}_n,y_n,s_n)\}$ where $S_P=\{(\mathbf{x}_i,y_i,s_i)\}_{i=1}^k$ is the positive set with size k and $S_U=\{(\mathbf{x}_i,y_i,s_i)\}_{i=k+1}^n$ denotes the unlabeled set with size $n-k$. According to the definition of single-training-set PU learning mentioned above, we have $y_i=s_i=1$ for $\mathbf{x}_i\in S_P$, and $y_i=\mathrm{unknown}$, $s_i=0$ for $\mathbf{x}_i\in S_U$. Then our target is to find a suitable probabilistic score function $h\colon\mathcal{X}\to[0,1]$ on S, such that the unobserved test example $\mathbf{x}$ can obtain the correct label $sgn(h(\mathbf{x})-0.5)$ assigned by h.

3.1 General Graphical Model Construction

The relationship among $\mathbf{x}$, y and s can be depicted by the graphical model as illustrated in Fig. 1(b). Firstly, the real label y should obviously depend on the features $\mathbf{x}$ of data. Secondly, whether the label of an input datum will be observed is related to two factors: one is the real label y, and the other is the properties or representations $\mathbf{x}$. Note that almost all existing PU models only considers the relationship $y\to s$, but our method also considers the dependency $\mathbf{x}\to s$ in addition to $y\to s$, and that is the reason that our method is instance-dependent. In fact, $\mathbf{x}\to s$ critically models the labeling bias in realistic situations caused by various factors such as the labeling difficulty of data and the professional specialty of labeler. That is to say, we do not assume that all positive examples have an equal probability to be labeled, and how likely they are labeled should depend on the observed features.

Based on above definitions, we have the following generative formulation regarding $\mathbf{x}$, y and s according to Fig. 1(b), which is

$$P(y,s|\mathbf{x}) = P(y|\mathbf{x})P(s|y,\mathbf{x}). \tag{1}$$

In Eq. (1), the term $P(y|\mathbf{x})$ outputs a probabilistic value of an example to be class y, so it can be used to construct the score function $h(\mathbf{x};\boldsymbol{\theta}_1)$ parameterized by $\boldsymbol{\theta}_1$. Besides, the probability $P(s|y,\mathbf{x})$ explicitly describes the labeling process of $\mathbf{x}$. The detailed formations of $P(y|\mathbf{x})$ and $P(s|y,\mathbf{x})$ will be specified in Section 3.2.

By denoting $\{\mathbf{x}_1,\cdots,\mathbf{x}_n\}$, $\{y_1,\cdots,y_n\}$ and $\{s_1,\cdots,s_n\}$ as the observations of $\mathbf{x}$, y and s correspondingly, and assuming that the n training examples $\{\mathbf{x}_i,y_i,s_i\}_{i=1}^n$ are independently and identically sampled, then the joint conditional distribution in accordance to Eq. (1) is expressed as

$$P(y,s|\mathbf{x}) = \prod_{i=1}^{n} P(y_i,s_i|\mathbf{x}_i) = \prod_{i=1}^{n} P(y_i|\mathbf{x}_i)P(s_i|y_i,\mathbf{x}_i). \tag{2}$$

3.2 Model Instantiations

Eq. (1) provides a hybrid formation for the general graphical model, next we need to define the forms of the conditional probabilities $P(y|\mathbf{x})$ and $P(s|y,\mathbf{x})$ to make our model tractable.

For $P(y|\mathbf{x})$, it is the posterior probability on the input $\mathbf{x}$ determined by some score function $h(\mathbf{x};\boldsymbol{\theta}_1)$ with parameter $\boldsymbol{\theta}_1$. For $P(s|y,\mathbf{x})$, since only the positive examples will have a probability to be labeled, and the labels of negative examples will never be observed, we have the following facts:

$$P(s=0|y=0,\mathbf{x}) = 1, \tag{3}$$
$$P(s=1|y=0,\mathbf{x}) = 0, \tag{4}$$
$$P(s=1|y=1,\mathbf{x}) = \eta(\mathbf{x};\boldsymbol{\theta}_2), \tag{5}$$
$$P(s=0|y=1,\mathbf{x}) = 1-\eta(\mathbf{x};\boldsymbol{\theta}_2), \tag{6}$$

where $\eta(\mathbf{x};\boldsymbol{\theta}_2)$ is the probability of $\mathbf{x}$ with $y=1$ to be labeled as defined in the Introduction, and it relates to $\mathbf{x}$ by parameter $\boldsymbol{\theta}_2$. Note that $\eta(\mathbf{x};\boldsymbol{\theta}_2)$ is also called "propensity score" in [21]. The above Eqs. (3)$\sim$(6) can be concisely rewritten as:

$$P(s=s'|y,\mathbf{x}) = \begin{cases} (1-\eta(\mathbf{x};\boldsymbol{\theta}_2))^{1-s'}\eta(\mathbf{x};\boldsymbol{\theta}_2)^{s'}, & y=1 \\ 1-s', & y=0 \end{cases}. \tag{7}$$

In this paper, we provide two expressions for realizing $h(\mathbf{x};\boldsymbol{\theta}_1)$ and $\eta(\mathbf{x};\boldsymbol{\theta}_2)$. The first one is based on the linear-in-parameter Logistic Function (termed "LBE-LF"), which results in

$$h(\mathbf{x};\boldsymbol{\theta}_1) = P(y=1|\mathbf{x}) = \left(1+\exp(-\boldsymbol{\theta}_1^\top\mathbf{x})\right)^{-1} \tag{8}$$

and

$$\eta(\mathbf{x};\boldsymbol{\theta}_2) = P(s=1|y=1,\mathbf{x}) = \left(1+\exp(-\boldsymbol{\theta}_2^\top\mathbf{x})\right)^{-1}. \tag{9}$$

The second one is based on a typical neural network named Multi-Layer Perceptron (MLP), and the induced model is dubbed as "LBE-MLP". As a deep model, LBE-MLP offers more flexibility for modeling $h(\mathbf{x};\boldsymbol{\theta}_1)$ and $\eta(\mathbf{x};\boldsymbol{\theta}_2)$ than the non-deep LBE-LF, as it can handle more uncertain and complex mappings from $\mathbf{x}$ to $h(\mathbf{x};\boldsymbol{\theta}_1)$ and $\eta(\mathbf{x};\boldsymbol{\theta}_2)$ practically. One may also use Convolutional Neural Networks (CNN) for implementing $h(\mathbf{x};\boldsymbol{\theta}_1)$ and $\eta(\mathbf{x};\boldsymbol{\theta}_2)$, but considering the applicability to different types of data beyond images, in this paper we choose to use MLP for model establishment.

Here we want to remark that our algorithm does not require the explicit value of class prior $P(y=1)$ which actually needs to be pre-estimated in many PU learning methods such as [3], [4], [38]. In fact, the estimation for $P(y=1)$ is practically non-trivial. The reason for our LBE in avoiding such estimation is that we directly estimate $h(\mathbf{x};\boldsymbol{\theta}_1) = P(y=1|\mathbf{x})$ and $\eta(\mathbf{x};\boldsymbol{\theta}_2) = P(s=1|y=1,\mathbf{x})$ which implicitly contains $P(y=1)$.

3.3 Parameter Learning

In our model, we have to estimate the parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2\}$ where $\boldsymbol{\theta}_1$ is the parameter in classifier $h(\mathbf{x};\boldsymbol{\theta}_1)$ and $\boldsymbol{\theta}_2$ is the parameter in labeling function $\eta(\mathbf{x};\boldsymbol{\theta}_2)$. To this end, we need to solve the following maximization problem, namely:

$$\arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{n} P(s_i|\mathbf{x}_i;\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{n}\sum_{y_i} P(s_i,y_i|\mathbf{x}_i;\boldsymbol{\theta}). \tag{10}$$

By taking the logarithm on the right-hand side of the above equation, the maximization problem on likelihood function is equivalent to

$$\arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n}\log\sum_{y_i} P(s_i,y_i|\mathbf{x}_i;\boldsymbol{\theta}). \tag{11}$$

Here we have the groundtruth labels y_i $(i=1,2,\cdots,n)$ as latent variables, so we naturally employ the EM algorithm to solve the optimization problem (11), which alternates between the E-step and M-step until convergence.

E-step: In E-step, we compute the latent variables y_i $(y=1,2,\cdots,n)$ which is the class posterior probability for every data point, namely $\tilde{P}(y_i) = P(y_i|\mathbf{x}_i,s_i)$. Since $P(y_i,s_i|\mathbf{x}_i) = P(s_i|\mathbf{x}_i)P(y_i|\mathbf{x}_i,s_i)$, we obtain the updating rule for E-step as

$$\tilde{P}(y_i) = P(y_i|\mathbf{x}_i,s_i) \propto P(y_i,s_i|\mathbf{x}_i) = P(s_i|y_i,\mathbf{x}_i)P(y_i|\mathbf{x}_i), \tag{12}$$

where the parameters of $P(s_i|y_i,\mathbf{x}_i)$ and $P(y_i|\mathbf{x}_i)$ are found by the following M-step. Note that in Eq. (12), the "$\propto$" notation is used and the term $P(s_i|\mathbf{x}_i)$ is dropped, as we do not need to explicitly compute $P(s_i|\mathbf{x}_i)$ for practical implementation. Specifically, we first compute the values of $P(y_i=1,s_i|\mathbf{x}_i)$ and $P(y_i=0,s_i|\mathbf{x}_i)$, and then conduct normalization on them in which $P(s_i|\mathbf{x}_i)$ actually plays a role as a normalization factor. Therefore, $\tilde{P}(y_i)$ is accurately computed and there are no approximations in Eq. (12).

M-step: M-step aims to optimize the parameters of the model in the presence of the training data and the new data assignments output by the E-step. That is to say, the parameter $\boldsymbol{\theta}$ should be updated by maximizing the expectation $\sum_i \mathbb{E}_{\tilde{P}(y_i)}[\log P(y_i,s_i|\mathbf{x}_i;\boldsymbol{\theta})]$, which leads to

$$\max_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) \tag{13}$$

with

$$\begin{aligned}\mathcal{J}(\boldsymbol{\theta}) &= \sum_i \mathbb{E}_{\tilde{P}(y_i)}[\log P(y_i,s_i|\mathbf{x}_i;\boldsymbol{\theta})] \\ &= \sum_i \mathbb{E}_{\tilde{P}(y_i)}[\log P(y_i|\mathbf{x}_i;\boldsymbol{\theta}_1) + \log P(s_i|y_i,\mathbf{x}_i;\boldsymbol{\theta}_2)].\end{aligned} \tag{14}$$

Since there is no closed-form solution for Eq. (13), here we adopt the Adam optimization algorithm [39] to update $\boldsymbol{\theta}$. For

LBE-MLP, the gradients $\nabla_{\boldsymbol{\theta}_1}\mathcal{J}(\boldsymbol{\theta})$ and $\nabla_{\boldsymbol{\theta}_2}\mathcal{J}(\boldsymbol{\theta})$ are respectively computed as

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_1}\mathcal{J}(\boldsymbol{\theta}) &= \sum_i \nabla_{\boldsymbol{\theta}_1}\mathbb{E}_{\tilde{P}(y_i)}[\log P(y_i|\mathbf{x}_i;\boldsymbol{\theta}_1)] \\ &= \sum_i \nabla_{\boldsymbol{\theta}_1}[\tilde{P}(y_i=0)\log P(y_i=0|\mathbf{x}_i;\boldsymbol{\theta}_1) \\ &\quad + \tilde{P}(y_i=1)\log P(y_i=1|\mathbf{x}_i;\boldsymbol{\theta}_1)] \\ &= \sum_i\sum_{y_i}\frac{\tilde{P}(y_i)}{P(y_i|\mathbf{x}_i;\boldsymbol{\theta}_1)}\nabla_{\boldsymbol{\theta}_1}P(y_i|\mathbf{x}_i;\boldsymbol{\theta}_1)\end{aligned} \tag{15}$$

and

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_2}\mathcal{J}(\boldsymbol{\theta}) &= \sum_i \nabla_{\boldsymbol{\theta}_2}\mathbb{E}_{\tilde{P}(y_i)}[\log P(s_i|y_i,\mathbf{x}_i;\boldsymbol{\theta}_2)] \\ &= \sum_i \nabla_{\boldsymbol{\theta}_2}[\tilde{P}(y_i=0)\log P(s_i|y_i=0,\mathbf{x}_i;\boldsymbol{\theta}_2) \\ &\quad + \tilde{P}(y_i=1)\log P(s_i|y_i=1,\mathbf{x}_i;\boldsymbol{\theta}_2)] \\ &= \sum_i\sum_{y_i}(-1)^{s_i+1}\frac{\mathbb{1}\{y_i=1\}\tilde{P}(y_i)}{P(s_i|y_i,\mathbf{x}_i;\boldsymbol{\theta}_2)}\nabla_{\boldsymbol{\theta}_2}\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)\end{aligned} \tag{16}$$

where "$\mathbb{1}\{\cdot\}$" is the indicator function which equals to 1 if the argument inside the bracket holds, and 0 otherwise.

For LBE-LF, by plugging $\nabla_{\boldsymbol{\theta}_1}h(\mathbf{x}_i;\boldsymbol{\theta}_1) = h(\mathbf{x}_i;\boldsymbol{\theta}_1)(h(\mathbf{x}_i;\boldsymbol{\theta}_1)-1)\mathbf{x}_i$ and $\nabla_{\boldsymbol{\theta}_2}\eta(\mathbf{x}_i;\boldsymbol{\theta}_2) = \eta(\mathbf{x}_i;\boldsymbol{\theta}_2)(\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)-1)\mathbf{x}_i$ for logistic function into Eq. (15) and Eq. (16) accordingly, we obtain the gradient expressions as

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}_1}\mathcal{J}(\boldsymbol{\theta}) \\ &= \sum_i \frac{\tilde{P}(y_i=1)}{P(y_i=1|\mathbf{x}_i;\boldsymbol{\theta}_1)}\nabla_{\boldsymbol{\theta}_1}P(y_i=1|\mathbf{x}_i;\boldsymbol{\theta}_1) \\ &\quad + \frac{\tilde{P}(y_i=0)}{1-P(y_i=1|\mathbf{x}_i;\boldsymbol{\theta}_1)}\nabla_{\boldsymbol{\theta}_1}(1-P(y_i=1|\mathbf{x}_i;\boldsymbol{\theta}_1)) \\ &= \sum_i \tilde{P}(y_i=1)(h(\mathbf{x}_i;\boldsymbol{\theta}_1)-1)\mathbf{x}_i + \tilde{P}(y_i=0)h(\mathbf{x}_i;\boldsymbol{\theta}_1)\mathbf{x}_i \\ &= \sum_i\sum_{y_i}\tilde{P}(y_i)(h(\mathbf{x}_i;\boldsymbol{\theta}_1)-y_i)\mathbf{x}_i\end{aligned} \tag{17}$$

and

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}_2}\mathcal{J}(\boldsymbol{\theta}) \\ &= \sum_i\sum_{y_i}(-1)^{s_i+1}\frac{\mathbb{1}\{y_i=1\}\tilde{P}(y_i)}{P(s_i|y_i,\mathbf{x}_i;\boldsymbol{\theta}_2)}\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)(\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)-1)\mathbf{x}_i.\end{aligned} \tag{18}$$

Above E-step and M-step iterates until convergence. The entire algorithm is summarized in Algorithm 1. When the optimal model parameter $\boldsymbol{\theta}_1^*$ is obtained, one can make label inference on the unseen test data according to the score function $h(\mathbf{x};\boldsymbol{\theta}_1^*)$.

4 Model Interpretation

In this section, we reveal that the proposed LBE algorithm can be understood as a rectified logistic regression on noisily labeled training data.

Algorithm 1 An outline of our LBE algorithm.

```
Input: The training set $S = \{S_P; S_U\}$; the initial parameters $\boldsymbol{\theta}_1^{init}$ and $\boldsymbol{\theta}_2^{init}$; the parameters in Adam including step size $\tau$ and exponential decay rates $\rho_1, \rho_2$.
Output: The optimal parameters $\boldsymbol{\theta}_1^*$ for the classifier $h(\mathbf{x};\boldsymbol{\theta}_1)$; the optimal parameters $\boldsymbol{\theta}_2^*$ for estimating $\eta(\mathbf{x};\boldsymbol{\theta}_2)$.
Initialize $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_1^{init}$ and $\boldsymbol{\theta}_2 = \boldsymbol{\theta}_2^{init}$;
while Not Converged do
    # E-step (predict the class posterior probability)
    Compute $\tilde{P}(y_i) \propto P(s_i|y_i,\mathbf{x}_i)P(y_i|\mathbf{x}_i)$ for $i = 1,\cdots,n$ via Eq. (12);
    # M-step (update model parameters)
    Call Adam [39] to update $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, where $\nabla_{\boldsymbol{\theta}_1}\mathcal{J}(\boldsymbol{\theta})$ and $\nabla_{\boldsymbol{\theta}_2}\mathcal{J}(\boldsymbol{\theta})$ are computed via Eq. (17) and Eq. (18) if LBE-LF, and are computed via Eq. (15) and Eq. (16) if LBE-MLP;
end while
$\boldsymbol{\theta}_1^* = \boldsymbol{\theta}_1$; $\boldsymbol{\theta}_2^* = \boldsymbol{\theta}_2$;
return $\boldsymbol{\theta}_1^*$ and $\boldsymbol{\theta}_2^*$.
```

According to the logarithm of likelihood function shown in Eq. (11), we may derive

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \sum_{i=1}^{n}\log\sum_{y_i}P(s_i|y_i,\mathbf{x}_i;\boldsymbol{\theta}_2)P(y_i|\mathbf{x}_i;\boldsymbol{\theta}_1) \\ &= \sum_{i=1}^{n}\log[P(y_i=1|\mathbf{x}_i;\boldsymbol{\theta}_1)(1-\eta(\mathbf{x}_i;\boldsymbol{\theta}_2))^{1-s_i}\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)^{s_i} \\ &\qquad + P(y_i=0|\mathbf{x}_i;\boldsymbol{\theta}_1)(1-s_i)] \\ &\overset{1}{=} \sum_{i=1}^{k}\log[P(y_i=1|\mathbf{x}_i;\boldsymbol{\theta}_1)\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)] \\ &\quad + \sum_{i=k+1}^{n}\log[P(y_i=1|\mathbf{x}_i;\boldsymbol{\theta}_1)(1-\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)) + P(y_i=0|\mathbf{x}_i;\boldsymbol{\theta}_1)] \\ &= \sum_{i=1}^{n}s_i\log[\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)P(y_i=1|\mathbf{x}_i;\boldsymbol{\theta}_1)] \\ &\quad + (1-s_i)\log[1-\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)P(y_i=1|\mathbf{x}_i;\boldsymbol{\theta}_1)], \\ &\overset{2}{=} \sum_{i=1}^{n}s_i\log\bar{h}(\mathbf{x}_i;\boldsymbol{\theta}) + (1-s_i)\log(1-\bar{h}(\mathbf{x}_i;\boldsymbol{\theta})),\end{aligned} \tag{19}$$

where the 1st identity uses the fact that $s_i = 1$ and 0 for $\mathbf{x}_i \in S_P$ and $\mathbf{x}_i \in S_U$, respectively; and in the 2nd identity, $\bar{h}(\mathbf{x}_i;\boldsymbol{\theta}) = \eta(\mathbf{x}_i;\boldsymbol{\theta}_2)P(y_i=1|\mathbf{x}_i;\boldsymbol{\theta}_1) = \eta(\mathbf{x}_i;\boldsymbol{\theta}_2)h(\mathbf{x}_i;\boldsymbol{\theta}_1)$ parameterized by $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2\}$ can be regarded as a rectified score function by imposing a factor $\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)$ to the plain score function $h(\mathbf{x}_i;\boldsymbol{\theta}_1)$. From Eq. (19), we have the following interesting findings:

1) If we regard s_i as the label of $\mathbf{x}_i$ for $i = 1,2,\cdots,n$, the maximization of Eq. (19) soon becomes the formulation of standard logistic regression with cross-entropy loss. Here all unlabeled examples $\mathbf{x}_i \in S_U$ are assigned negative labels $s_i = 0$, which are actually inaccurate as some of the original positive data are hidden in S_U. In contrast, the labels s_i for positive examples $\mathbf{x}_i \in S_P$ are all correct. Therefore, our LBE algorithm can also be interpreted as a one-sided label noise learning problem as mentioned in Section 2.
2) To remedy the inaccuracy of s_i ($i = k+1,\cdots,n$), the plain score function $h(\mathbf{x}_i;\boldsymbol{\theta}_1)$ is associated with an extra term $\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)$ to form $\bar{h}(\mathbf{x}_i;\boldsymbol{\theta})$. Here $\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)$, which

should also be estimated from data, critically controls the "strength" of $h(\mathbf{x}_i;\boldsymbol{\theta}_1)$ in obtaining the optimal parameter $\boldsymbol{\theta}_1^*$. Concretely, when the label of a positive example is observed with high probability (*i.e.*, $\eta(\mathbf{x}_i)$ approaches to 1), the rectified score function $\bar{h}(\mathbf{x}_i;\boldsymbol{\theta})$ is trustable and its output will approach to that of $h(\mathbf{x}_i;\boldsymbol{\theta}_1)$. This also indicates that the performance of our algorithm will get close to the ideal fully supervised classifier if the probability of every positive data being labeled approaches to 1.

3) For the network-structured LBE-MLP, we may regard $\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)$ as an adaptation layer to $h(\mathbf{x}_i;\boldsymbol{\theta}_1)$ for handling the one-sided label noise mentioned in 1). Consequently, LBE-MLP has a similar framework to some approaches with various noise adaptation layers or loss correction techniques for handling noisy labels or complementary labels [40], [41], [42], [43], by which different classifier-consistent deep learning algorithms are constructed.

5 Theoretical Analyses

In this section, we conduct some theoretical analyses on our proposed LBE algorithm.

5.1 Existence and Uniqueness of MLE

It should be noted that the solution for a general MLE problem may not exist, and the solutions may even not be unique when they exist. Here we show that the solution of our method always exists, and the solution is unique under certain conditions.

To prove the existence of solution to our LBE, we first provide an existing result for a general MLE problem:

Proposition 1. *(Sufficient condition for existence of estimator, [44]) Given an MLE problem $max_{\boldsymbol{\alpha}\in\Gamma}\mathcal{L}(x;\boldsymbol{\alpha})$ where Γ is parameter space, x is a random variable, and $\mathcal{L}(x;\boldsymbol{\alpha})$ is (log)-likelihood function. If Γ is compact and $\mathcal{L}(x;\boldsymbol{\alpha})$ is continuous on Γ, then there exists a maximum likelihood estimator.*

By checking the conditions in the above theorem, we see that the existence of LBE solution is obvious. Firstly, the parameter $\boldsymbol{\theta}$ in our derived model satisfies $\|\boldsymbol{\theta}\| < +\infty$, and the related parameter space is also closed, so it is compact. Besides, as our likelihood function $\mathcal{L}(\boldsymbol{\theta})$ is made up of several elementary functions, so it is continuous on the parameter space.

To prove the solution uniqueness, we also present a sufficient condition for general MLE problem, namely:

Proposition 2. *(Sufficient condition for uniqueness of estimator, [44]) Let $\boldsymbol{\alpha} \in \Gamma$ and $\mathcal{L}(\boldsymbol{\alpha})$ be a twice continuously differentiable real-valued function on Γ. If the Hessian matrix $\mathbf{H} = \partial^2\mathcal{L}(\boldsymbol{\alpha})/\partial\boldsymbol{\alpha}^2$ of second partial derivatives is negative definite at every point $\boldsymbol{\alpha} \in \Gamma$ for which the gradient vector $\nabla\mathcal{L}(\boldsymbol{\alpha}) = \partial\mathcal{L}(\boldsymbol{\alpha})/\partial\boldsymbol{\alpha}$ vanishes, then $\mathcal{L}(\boldsymbol{\alpha})$ has a unique local (and hence global) maximum and no other critical points.*

Above proposition can also be understood from the viewpoint of optimization. If the Hessian matrix of likelihood function $\mathcal{L}(\boldsymbol{\alpha})$ is negative definite, $\mathcal{L}(\boldsymbol{\alpha})$ is concave and thus its maximization problem will have a unique global solution. In our case, since both $y|\mathbf{x}$ and $s|y=1,\mathbf{x}$ obey Bernoulli distribution, if they are independent to each other, the distribution of their multiplication will also be Bernoulli which belongs to the exponential family. Therefore, the conditions in Proposition 2 are satisfied [45], and the solution of Eq. (11) exists and is also unique. However, since this paper considers the instance-dependent PU learning setting, such independency between $y|\mathbf{x}$ and $s|y=1,\mathbf{x}$ does not hold. As a result, below we show that our linear-in-parameter model LBE-LF leads to a local unique solution under certain conditions.

Theorem 3. *(Uniqueness of LBE-LF estimator) Given the log-likelihood function $\mathcal{L}(\boldsymbol{\theta})$ expressed as Eq. (11), and $h(\mathbf{x};\boldsymbol{\theta}_1) = P(y=1|\mathbf{x};\boldsymbol{\theta}_1)$ and $\eta(\mathbf{x};\boldsymbol{\theta}_2) = P(s=1|y=1,\mathbf{x};\boldsymbol{\theta}_2)$ are modeled by logistic function, if $\bar{h}(\mathbf{x};\boldsymbol{\theta}_1) > 2h(\mathbf{x};\boldsymbol{\theta}_1) - 1$; and $\eta(\mathbf{x};\boldsymbol{\theta}_2) < 0.5$ when $\mathbf{x} \in S_U$, the likelihood function $\mathcal{L}(\boldsymbol{\theta})$ regarding $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ is concave, respectively, which leads to a local unique solution for maximizing $\mathcal{L}(\boldsymbol{\theta})$.*

The proof of Theorem 3 is provided in Section 8. This theorem reveals that the MLE problem defined by our LBE-LF method is meaningful, and the obtained classifier will not be too bad from the perspective of maximizing the likelihood function. Besides, this theorem can be understood as a weak explanation for identifiability of solution in LBE-LF. That is to say, under certain conditions, a unique solution can be identified, which helps to yield satisfactory results. Whether the optimal solution can be fully identified without these conditions still requires further strict theoretical investigations which may relate to the irreducibility of distributions [46], [47]. However, although the identifiability of our LBE-LF method is only partially explained, its performance is still quite encouraging as empirically illustrated by the experimental results in Section 6. For the deep LBE-MLP model, due to the high non-linearity of neural network, the above Theorem 3 may not be applicable. Nevertheless, the experimental results presented in the following Section 6 also empirically demonstrate the satisfactory performance.

5.2 Generalization Error

This section studies the generalizability of the proposed LBE-MLP and LBE-LF algorithms. Specifically, we focus on the rectified classifier $\bar{h}(\mathbf{x}_i;\boldsymbol{\theta})$ (abbreviated as $\bar{h}$ when no confusion is incurred) according to the explanations in Section 4, of which the expected risk $R(\bar{h})$ and empirical risk $R_S(\bar{h})$ are respectively defined as:

$$R(\bar{h}) = \mathbb{E}_{(\mathbf{x},s)\sim\mathcal{D}_s}[\ell(\bar{h}(\mathbf{x};\boldsymbol{\theta}),s)] \tag{20}$$

and

$$R_S(\bar{h}) = \frac{1}{n}\sum_{i=1}^{n}\ell(\bar{h}(\mathbf{x}_i;\boldsymbol{\theta}),s_i) \tag{21}$$

where $\mathcal{D}_s$ is the distribution from which $\{(\mathbf{x}_i,s_i)\}_{i=1}^n$ are generated, and the loss function $\ell(\bar{h}(\mathbf{x}_i;\boldsymbol{\theta}),s_i)$ here is the cross-entropy loss. To prove the generalizability of the proposed LBE method, some definitions and existing theoretical results are necessary.

Definition 4. *(Empirical Rademacher complexity, [48]) Let $r = \{r_1,\cdots,r_n\}$ be a set of independent Rademacher variables which are uniformly sampled from $\{-1,1\}$, $\ell(\cdot)$ be a loss function, $S=\{\mathbf{x}_1,\cdots,\mathbf{x}_n\}$ be an independent distributed sample set, and $\mathcal{H}$ a function class, then the empirical Rademacher complexity of the composition of ℓ and all $\bar{h} \in \mathcal{H}$ (i.e., $\ell\circ\mathcal{H}$) is defined as:*

$$\hat{\mathcal{R}}_S(\ell\circ\mathcal{H}) = \mathbb{E}_r[\sup_{\bar{h}\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n}r_i\ell(\bar{h}(\mathbf{x}_i),s_i)]. \tag{22}$$

Proposition 5. *(Generalization bound, [48]) Given $\mathbf{x}_1, \cdots, \mathbf{x}_n$ are i.i.d variables, and the loss function $\ell(\cdot)$ is upper bound by A, then for any $\delta > 0$, with probability at least $1 - \delta$, we have*

$$\sup_{\bar{h} \in \mathcal{H}} \left| R(\bar{h}) - R_S(\bar{h}) \right| \leq 2\hat{\mathcal{R}}_S(\ell \circ \mathcal{H}) + 3A\sqrt{\frac{\log 2/\delta}{2n}}. \tag{23}$$

Next we bound the generalization error of LBE-MLP, and then that of LBE-LF.

5.2.1 Generalization Error of LBE-MLP

For LBE-MLP, we assume that $h(\mathbf{x}; \boldsymbol{\theta}_1)$ is a network consisted of l_1 layers with parameters $\boldsymbol{\theta}_1^{(1)}, \cdots, \boldsymbol{\theta}_1^{(l_1)}$ and activation functions $\sigma_1^{(1)}, \cdots, \sigma_1^{(l_1-1)}$ with $\sigma_1^{(i)}(\mathbf{0}) = 0$ for $i = 1, \cdots, l_1 - 1$, that is, $h(\mathbf{x}; \boldsymbol{\theta}_1) = \mathrm{softmax}(h'(\mathbf{x}))$ where $h'(\mathbf{x}) = \boldsymbol{\theta}_1^{(l_1)} \sigma_1^{(l_1-1)}(\boldsymbol{\theta}_1^{(l_1-1)} \sigma_1^{(l_1-2)}(\cdots \sigma_1^{(1)}(\boldsymbol{\theta}_1^{(1)} \mathbf{x}))) = (h'_0(\mathbf{x}) \ h'_1(\mathbf{x})) \in \mathbb{R}_+^2$ outputs the nonnegative responses of network on the unlabeled data with $s_i = 0$ and positive data with $s_i = 1$, and the notation $h(\mathbf{x}; \boldsymbol{\theta}_1) = \mathrm{softmax}(h'(\mathbf{x})) = \exp(h'_i(\mathbf{x})) / \sum_{j=0}^{1} \exp(h'_j(\mathbf{x})) = (h_0(\mathbf{x}) \ h_1(\mathbf{x}))$ $(i = 0, 1)$ is also slightly abused to denote a two-dimensional nonnegative vector. Similarly, $\eta(\mathbf{x}; \boldsymbol{\theta}_2)$ is represented as $\eta(\mathbf{x}; \boldsymbol{\theta}_2) = \boldsymbol{\theta}_2^{(l_2)} \sigma_2^{(l_2-1)}(\boldsymbol{\theta}_2^{(l_2-1)} \sigma_2^{(l_2-2)}(\cdots \sigma_2^{(1)}(\boldsymbol{\theta}_2^{(1)} \mathbf{x})))$ with $\boldsymbol{\theta}_2^{(1)}, \cdots, \boldsymbol{\theta}_2^{(l_2)}$ being the parameters of totally l_2 layers and $\sigma_2^{(1)}, \cdots, \sigma_2^{(l_2-1)}$ being the activation functions. Therefore, the classifier $\hat{\bar{h}}(\mathbf{x}) = \arg\max_{i=0,1} \hat{\bar{h}}_i(\mathbf{x})$ learned in the hypothesis space $\mathcal{H}$ is denoted by $\hat{\bar{h}}(\mathbf{x}) = \arg\min_{\bar{h} \in \mathcal{H}} R_S(\bar{h}(\mathbf{x}))$.

Lemma 6. *Given $\bar{h}_0(\mathbf{x}_i)$ and $\bar{h}_1(\mathbf{x}_i)$ as the responses of rectified classifier $\bar{h}(\mathbf{x}_i)$ on the unlabeled data and positive data correspondingly, with $\bar{h}_0(\mathbf{x}_i) = 1 - \bar{h}_1(\mathbf{x}_i)$, then the adopted cross-entropy loss $\ell(\bar{h}(\mathbf{x}_i), s_i) = -[s_i \log \bar{h}_1(\mathbf{x}_i) + (1 - s_i) \log \bar{h}_0(\mathbf{x}_i)]$ is 1-Lipschitz continuous w.r.t. $h'_j(\mathbf{x}_i)$ for $j = 0, 1$, namely*

$$\left| \frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h'_j(\mathbf{x}_i)} \right| < 1. \tag{24}$$

Due to the 1-Lipschitz continuity of the cross-entropy loss illustrated in Lemma 6, we have the following lemma regarding the empirical Rademacher complexity $\hat{\mathcal{R}}_S(\ell \circ \mathcal{H})$ mentioned in Definition 4, which is

Lemma 7. *If the loss function $\ell(\bar{h}(\mathbf{x}_i), s_i)$ is 1-Lipschitz continuous, and $\mathcal{H}'$ is the hypothesis space of $h'(\mathbf{x}_i)$, we have*

$$\hat{\mathcal{R}}_S(\ell \circ \mathcal{H}) \leq \mathbb{E}_r[\sup_{h' \in \mathcal{H}'} \frac{1}{n} \sum_{i=1}^{n} r_i(h'(\mathbf{x}_i))]. \tag{25}$$

The above Lemmas 6 and 7 are proved in Section 8. Apart from them, we also need the following lemma:

Lemma 8. *[49] Given a l-layer neural network $h'(\mathbf{x}_i)$ with the layer parameters $\left\| \boldsymbol{\theta}^{(i)} \right\|_{\mathrm{F}} \leq M^{(i)}$ for $i = 1, \cdots, l$, $\|\mathbf{x}_i\|_2 \leq B$ for any $\mathbf{x}_i \in \mathcal{X}$, and the activation functions being 1-Lipschitz, positive-homogeneous, and applied element-wise, then we have*

$$\mathbb{E}_r[\sup_{h' \in \mathcal{H}'} \frac{1}{n} \sum_{i=1}^{n} r_i h'(\mathbf{x}_i)] \leq \frac{B(\sqrt{2l \log 2} + 1) \prod_{i=1}^{l} M^{(i)}}{\sqrt{n}}. \tag{26}$$

Based on the above lemmas, we are ready to present the generalization error bound for LBE-MLP in the following theorem:

Theorem 9. *(Generalization bound of LBE-MLP) Assume that the Frobenius norm of the parameters $\boldsymbol{\theta}_1^{(1)}, \cdots, \boldsymbol{\theta}_1^{(l_1)}$ are respectively upper bounded by $M_1^{(1)}, \cdots, M_1^{(l_1)}$, i.e. $\forall i = 1, \cdots, l_1$, $\left\| \boldsymbol{\theta}_1^{(i)} \right\|_{\mathrm{F}} \leq M_1^{(i)}$; the feature vector of any example $\mathbf{x} \in \mathcal{X}$ satisfies $\|\mathbf{x}\|_2 \leq B$; and the activation functions to be 1-Lipschitz continuous, positive-homogeneous, and applied element-wise (such as the ReLU). Given the loss function $\ell(\bar{h}(\mathbf{x}_i), s_i)$ upper bound by A, then for any $\delta > 0$, with probability at least $1 - \delta$, we have*

$$\begin{aligned} & R(\hat{\bar{h}}) - R_S(\hat{\bar{h}}) \\ & \leq 2B(\sqrt{2l_1 \log 2} + 1) \prod_{i=1}^{l_1} M_1^{(i)} \left(\frac{1}{\sqrt{k}} + \frac{1}{\sqrt{n-k}} \right) + 3A\sqrt{\frac{\log 2/\delta}{2n}}, \end{aligned} \tag{27}$$

where k and n are the sizes of S_P and S correspondingly.

Proof. It is apparent that

$$R(\hat{\bar{h}}) - R_S(\hat{\bar{h}}) \leq \sup_{\bar{h} \in \mathcal{H}} \left| R(\bar{h}) - R_S(\bar{h}) \right|, \tag{28}$$

of which the right-hand side can be upper bounded by using Proposition 5. Therefore, the main problem here is to bound the empirical Rademacher complexity $\hat{\mathcal{R}}_S(\ell \circ \mathcal{H})$, which is further bounded by $\hat{\mathcal{R}}_S(\ell \circ \mathcal{H}) \leq \hat{\mathcal{R}}_P(\ell \circ \mathcal{H}) + \hat{\mathcal{R}}_U(\ell \circ \mathcal{H})$ according to the triangle inequality [48]. Here $\hat{\mathcal{R}}_P(\ell \circ \mathcal{H}) = \mathbb{E}_{r^P}[\sup_{\bar{h} \in \mathcal{H}} \frac{1}{k} \sum_{i=1}^{k} r_i^P \ell(\bar{h}(\mathbf{x}_i), s_i)]$ is the Rademacher complexity corresponding to positive set S_P with $r^P = \{r_i^P\}_{i=1}^{k}$ being the related Rademacher variables, and $\hat{\mathcal{R}}_U(\ell \circ \mathcal{H}) = \mathbb{E}_{r^U}[\sup_{\bar{h} \in \mathcal{H}} \frac{1}{n-k} \sum_{i=k+1}^{n} r_i^U \ell(\bar{h}(\mathbf{x}_i), s_i)]$ is the Rademacher complexity corresponding to unlabeled set S_U with $r^U = \{r_i^U\}_{i=k+1}^{n}$ being the associated Rademacher variables.

Due to that the adopted cross-entropy loss function for MLP is 1-Lipschitz continuous, then according to Lemmas 7 and 8, $\hat{\mathcal{R}}_P(\ell \circ \mathcal{H})$ and $\hat{\mathcal{R}}_U(\ell \circ \mathcal{H})$ can be respectively upper bounded by

$$\hat{\mathcal{R}}_P(\ell \circ \mathcal{H}) \leq \frac{B(\sqrt{2l_1 \log 2} + 1) \prod_{i=1}^{l_1} M^{(i)}}{\sqrt{k}} \tag{29}$$

$$\hat{\mathcal{R}}_U(\ell \circ \mathcal{H}) \leq \frac{B(\sqrt{2l_1 \log 2} + 1) \prod_{i=1}^{l_1} M^{(i)}}{\sqrt{n-k}}. \tag{30}$$

Therefore, by plugging Eqs. (29) and (30) into Eq. (23) in Proposition 5, Theorem 9 is proved. □

Theorem 9 shows that the generalization error of the learned rectified $\hat{\bar{h}}(\mathbf{x}; \boldsymbol{\theta})$ will converge to the empirical error on the labeling condition variables $\{s_i\}_{i=1}^{n}$ by increasing the PU sample size n. Therefore, if the labeling probability $\eta(\mathbf{x}; \boldsymbol{\theta}_2)$ is accurately estimated, the generalization error of $\hat{h}(\mathbf{x}; \boldsymbol{\theta}_1)$ induced by our algorithm will also be upper bounded on the training set with groundtruth labels $\{y_i\}_{i=1}^{n}$.

5.2.2 Generalization Error of LBE-LF

The generalization error bound of LBE-LF is presented in the following theorem:

Theorem 10. *(Generalization bound of LBE-LF) Assume that the model parameter $\|\boldsymbol{\theta}_1\|_2 \leq M$, and the input features $\|\mathbf{x}\| \leq B$,* *then given the loss function $\ell(\bar{h}(\mathbf{x}_i), s_i)$ upper bound by A, we have for any $\delta > 0$, with probability at least $1-\delta$,*

$$R(\hat{\bar{h}}) - R_S(\hat{\bar{h}}) \le 2BM\left(\frac{1}{\sqrt{k}}+\frac{1}{\sqrt{n-k}}\right)+3A\sqrt{\frac{\log 2/\delta}{2n}}. \tag{31}$$

Proof. The pipeline for proving the generalization error bound of LBE-LF is similar to that of LBE-MLP. The only difference is to bound the empirical Rademacher complexity of $\hat{\mathcal{R}}_S(\ell \circ \mathcal{H})$ appeared in Proposition 5. Note that the logistic function (8) is a generalized linear model with a nonlinear transformation $u(z) = 1/(1+\exp(-z))$ where $z = \boldsymbol{\theta}_1^\top \mathbf{x} = \langle \boldsymbol{\theta}_1, \mathbf{x}\rangle$ is the variable, and $u(z)$ is 1-Lipschitz continuous since $|\nabla_z u(z)| = |u(z)(1-u(z))| < 1$. Therefore, for LBE-LF, the empirical Rademacher complexity on positive dataset S_P satisfies

$$\begin{aligned}
\hat{\mathcal{R}}_P(\ell \circ \mathcal{H}) &= \mathbb{E}_{r^P}[\sup_{\bar{h}\in\mathcal{H}} \frac{1}{k}\sum_{i=1}^{k} r_i^P \ell(\bar{h}(\mathbf{x}_i), s_i)] \\
&= \mathbb{E}_{r^P}[\sup_{\boldsymbol{\theta}_1:\|\boldsymbol{\theta}_1\|_2\le M} \frac{1}{k}\sum_{i=1}^{k} r_i^P \ell(u(\langle\boldsymbol{\theta}_1, \mathbf{x}_i\rangle), s_i)] \\
&\overset{1}{\le} \mathbb{E}_{r^P}[\sup_{\boldsymbol{\theta}_1:\|\boldsymbol{\theta}_1\|_2\le M} \frac{1}{k}\sum_{i=1}^{k} r_i^P u(\langle\boldsymbol{\theta}_1, \mathbf{x}_i\rangle)] \\
&\overset{2}{\le} \mathbb{E}_{r^P}[\sup_{\boldsymbol{\theta}_1:\|\boldsymbol{\theta}_1\|_2\le M} \frac{1}{k}\sum_{i=1}^{k} r_i^P \langle\boldsymbol{\theta}_1, \mathbf{x}_i\rangle] \\
&= \frac{1}{k}\mathbb{E}_{r^P}[\sup_{\boldsymbol{\theta}_1:\|\boldsymbol{\theta}_1\|_2\le M} \boldsymbol{\theta}_1^\top \sum_{i=1}^{k} r_i^P \mathbf{x}_i] \\
&\overset{3}{\le} \frac{1}{k}\mathbb{E}_{r^P}\left[\sup_{\boldsymbol{\theta}_1:\|\boldsymbol{\theta}_1\|_2\le M} \|\boldsymbol{\theta}_1\|_2 \left\|\sum\nolimits_{i=1}^{k} r_i^P \mathbf{x}_i\right\|_2\right] \\
&= \frac{M}{k}\mathbb{E}_{r^P}\left[\left\|\sum\nolimits_{i=1}^{k} r_i^P \mathbf{x}_i\right\|_2\right] \\
&= \frac{M}{k}\mathbb{E}_{r^P}\left[\sqrt{\sum\nolimits_{i=1}^{k}\sum\nolimits_{j=1}^{k} r_i^P r_j^P \langle\mathbf{x}_i, \mathbf{x}_j\rangle}\right] \\
&\overset{4}{\le} \frac{M}{k}\sqrt{\mathbb{E}_{r^P}\left[\sum\nolimits_{i=1}^{k}\sum\nolimits_{j=1}^{k} r_i^P r_j^P \langle\mathbf{x}_i, \mathbf{x}_j\rangle\right]} \\
&= \frac{M}{k}\sqrt{\sum\nolimits_{i=1}^{k}\sum\nolimits_{j=1}^{k} \langle\mathbf{x}_i, \mathbf{x}_j\rangle \mathbb{E}_{r^P}[r_i^P r_j^P]} \\
&\overset{5}{=} \frac{M}{k}\sqrt{\sum\nolimits_{i=1}^{k} \|\mathbf{x}_i\|_2^2} \\
&= \frac{MB}{\sqrt{k}}.
\end{aligned} \tag{32}$$

In above derivations, the 1st and 2nd inequalities are due to the 1-Lipschitz continuity of cross-entropy loss and $u(z)$, respectively. The 3rd inequality is according to the Cauchy-Schwarz inequality. The 4^{th} inequality holds due to Jensen's inequality and the concavity of "$\sqrt{\cdot}$". Finally, the 5^{th} equality is obtained since $\mathbb{E}_{r^P}[r_i^P r_j^P]$ equals to 1 for $i = j$, and 0 for $i \neq j$.

Similarly, for the unlabeled set S_U, we may obtain

$$\hat{\mathcal{R}}_U(\ell \circ \mathcal{H}) \le \frac{MB}{\sqrt{n-k}}. \tag{33}$$

Therefore, Theorem 10 can be easily proved by substituting Eqs. (32) and (33) into Eq. (23). □

Theorems 9 and 10 indicate that the expected risks of the classifiers induced by our LBE-MLP and LBE-LF methods will converge to their empirical errors on the training set when the number of positive data or unlabeled data increases, and the convergence rate is $\mathcal{O}\big(\frac{1}{\sqrt{k}}+\frac{1}{\sqrt{n-k}}+\frac{1}{\sqrt{n}}\big)$. Therefore, our method is theoretically guaranteed to achieve satisfactory prediction performance for various PU classification tasks.

6 Experiments

In this section, we compare our proposed LBE algorithm (including two implementations LBE-LF and LBE-MLP) with several state-of-the-art PU learning methods on synthetic datasets, benchmark datasets, and real-world datasets. The compared baseline methods include the typical instance-independent algorithms such as unbiased PU model (uPU) [3], non-negative PU model (nnPU) [4], Loss Decomposition and Centroid Estimation (LDCE) [6]; and instance-dependent algorithms such as PU learning with a Selection Bias (PUSB) [38], Propensity-Weighted Estimator (PWE) [21]. To achieve fair comparison, we report the results generated by the linear models of uPU, nnPU, LDCE, PUSB, PWE and LBE-LF for our experiments. Besides, we also present the results of non-linear LBE-MLP, in which the normal three-layer MLP with hyperbolic tangent activation function is employed, and the dimension of the hidden layer is fixed to 10 unless otherwise specified. The Adam optimizer is adopted for parameter learning in each dataset with the default parameters specified in [39]. For our LBE, the target classifier $h(\mathbf{x}; \boldsymbol{\theta}_1)$ is initialized by training it on the dataset that naively takes the unlabeled examples as negative ones, and $\boldsymbol{\theta}_2$ in $\eta(\mathbf{x}; \boldsymbol{\theta}_2)$ is initialized to zero to achieve good convergence result [21].

To generate instance-dependent PU data with labeling bias, we first train a linear model by logistic regression on the data with groundtruth positive and negative labels. Then based on the predicted class probabilities of examples output by logistic regression, we select a subset of the positive training data as positive set S_P, and then combine the remaining positive data with negative data to compose the unlabeled set S_U. Specifically, in every dataset, the positive examples are respectively sampled according to the two sampling strategies below:

- Strategy 1: $\eta(\mathbf{x}) = \left[\left(1+\exp(-\boldsymbol{\theta}_{lgt}^{*\top}\mathbf{x})\right)^{-1}\right]^{\kappa}$;
- Strategy 2: $\eta(\mathbf{x}) = \left[1-\left(1+\exp(-\boldsymbol{\theta}_{lgt}^{*\top}\mathbf{x})\right)^{-1}\right]^{\kappa}$,

where $\boldsymbol{\theta}_{lgt}^{*}$ is the optimal parameter learned from logistic regression, and κ is set to 10 by following [38] to make the selected positive data more skewed than $\kappa = 1$. In Strategy 1, the positive data that are far from the potential ideal decision boundary are more likely to be labeled. This sampling strategy models the situations that the human annotators prefer to label the positive examples that they are almost sure. In contrast, in Strategy 2, the positive data that are close to the ideal decision boundary will have large probability to be labeled. This sampling strategy mimics the annotation preference similar to active learning [50] in which some critical data points in determining the final classifier are more likely to be labeled. Note that all positive data in S_P are sampled with replacement such that the obtained positive examples are identically and independently distributed.

6.1 Synthetic Datasets

First, we create a two-dimensional binary dataset that consists of two clusters of data generated from two Gaussians, and each

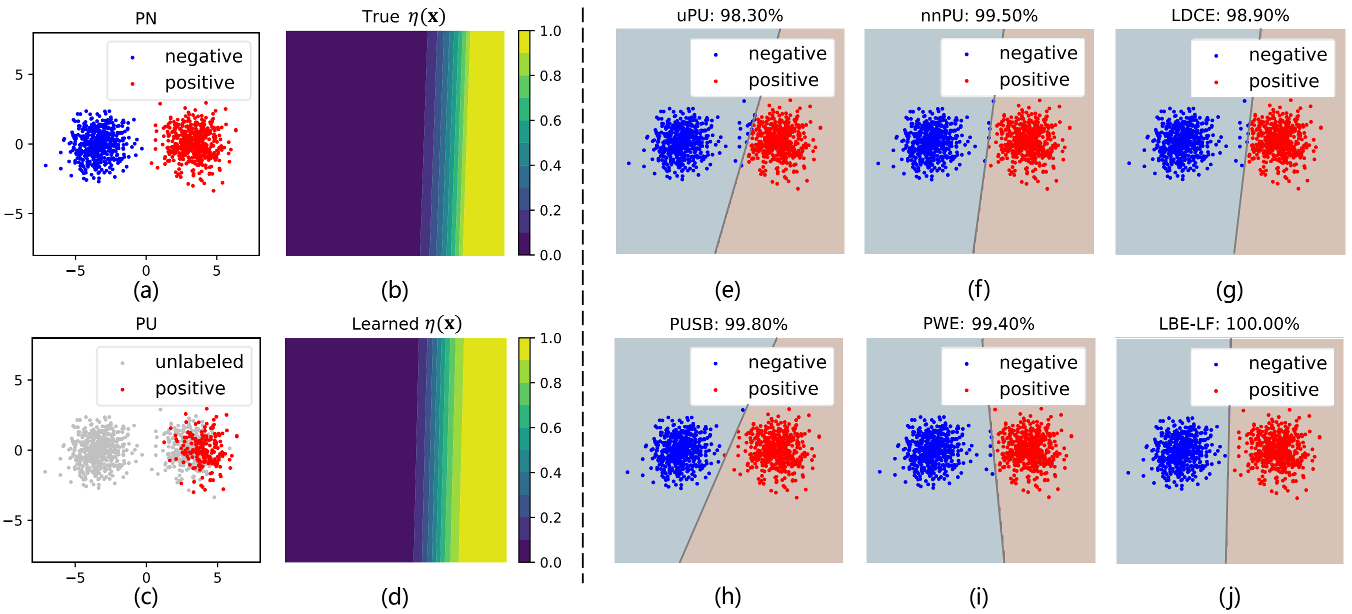

Fig. 2: The performances of various methods on the synthetic dataset under Strategy 1. (a) shows the real positive and negative data; (c) shows the unlabeled and biased positive data for model training; (b) and (d) present the true $\eta(\mathbf{x})$ and the estimated $\eta(\mathbf{x})$ of LBE; (e)$\sim$(j) display the classification results generated by uPU, nnPU, LDCE, PUSB, PWE, and LBE. The classification accuracy of every method is presented above the corresponding subfigure.

Gaussian corresponds to a class (positive/negative) as shown in Fig. 2(a) and Fig. 3(a). The centers of two Gaussians are $(2.7, 0)$ and $(-2.7, 0)$, respectively, and their variances are set to the same number 1. The entire dataset contains 1000 data points, and they are equally divided into two classes. After that, two instance-dependent PU datasets are generated based on the two sampling strategies mentioned above, which are illustrated in Fig. 2(c) for Strategy 1 and Fig. 3(c) for Strategy 2. The ratio of positive examples that are unlabeled to all original positive examples (denoted as π hereinafter) is set to 60% under each sampling strategy.

The classification results of LBE-LF and the compared methods are shown in Figs. 2(e)$\sim$(j) for Strategy 1 and in Figs. 3(e)$\sim$(j) for Strategy 2. On both datasets, LBE-LF is the only method that can achieve 100% accuracy, which is better than the results obtained by two state-of-the-art instance-dependent methods PUSB and PWE. For the instance-independent algorithms such as uPU, nnPU and LDCE, a considerable number of data points are mislabeled due to the biased sampling of positive examples. For example, in Fig. 2 corresponding to Strategy 1, some unlabeled examples that are originally positive near the decision boundary are classified as negative by uPU and LDCE, which suggests that the conventional instance-independent PU methods cannot well handle the labeling bias on positive data.

Moreover, we visualize the real probability value of $\eta(\mathbf{x})$ generated by Strategy 1 (Fig. 2(b)) and Strategy 2 (Fig. 3(b)), and also plot the estimated $\eta(\mathbf{x})$ by our LBE-LF (Fig. 2(d) and Fig. 3(d)). By comparing Fig. 2(b) vs. Fig. 2(d) and Fig. 3(b) vs. Fig. 3(d), we can easily observe that LBE-LF can correctly identify the biased labeling probabilities for positive examples, which is the key reason for our method to achieve satisfactory performance.

The results of LBE-MLP under Strategy 1 and Strategy 2 are presented in Fig. 4, from which we see that the output decision boundaries under both strategies are over-complicated and cannot reflect the real data distribution. In other words, LBE-MLP overfits the dataset, as this dataset is too simple for the "big" and complex neural network model. More experiments on this synthetic dataset under inseparable case and other positive data sampling strategy can be found in the supplementary material.

TABLE 1: The characteristics of six UCI datasets. n_+ and n_- are the amounts of positive and negative examples.

Dataset	n	d	n_+	n_-
australian	690	14	370	383
madelon	2000	500	1000	1000
phishing	11055	30	6157	4898
vote	435	16	267	168
banknote	1372	4	610	762
breast	683	10	143	540

6.2 UCI Benchmark Datasets

To demonstrate the effectiveness and robustness of LBE, we conduct extensive experiments on six UCI benchmark datasets[2] including *australian*, *madelon*, *phishing*, *vote*, *banknote*, and *breast*. The brief information of these datasets are presented in Table 1, from which we see that the number of examples in the employed datasets ranges from 435 to 11055. In our experiments, we conduct five-fold cross validation for our method and all the counterparts on each dataset. The mean test accuracies and the standard deviations over five independent trials are particularly investigated. Furthermore, we also applied the paired t-test with confidence level 95% to statistically examine whether our LBE (including LBE-LF and LBE-MLP) is significantly better/worse than other methods. As described in the beginning of Section 6, for every dataset, we apply two sampling strategies to establish the biased PU training set, with $\pi \in \{0.2, 0.3, 0.4\}$.

In our LBE-LF and LBE-MLP, the maximum iteration number is set to 100. For LBE-MLP, the weight decay parameter is chosen

[2] http://archive.ics.uci.edu/ml/index.php

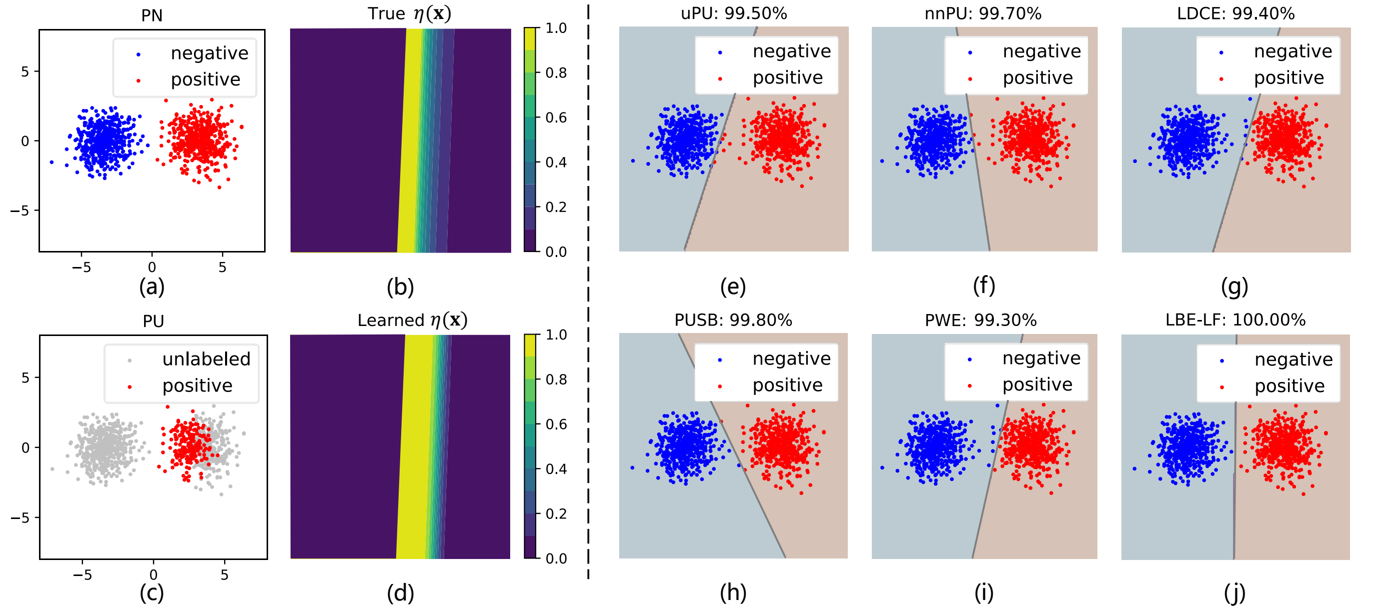

Fig. 3: The performances of various methods on the synthetic dataset under Strategy 2. (a) shows the real positive and negative data; (c) shows the unlabeled and biased positive data for model training; (b) and (d) present the true $\eta(\mathbf{x})$ and the estimated $\eta(\mathbf{x})$ of LBE; (e)∼(j) display the classification results generated by uPU, nnPU, LDCE, PUSB, PWE, and LBE. The classification accuracy of every method is presented above the corresponding subfigure.

from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$, and they are set to the optimal values on the six datasets by observing the validation accuracy. Regarding the baseline methods, the step discounted parameter γ and the tolerance parameter β in nnPU are respectively fixed to 0.001 and 0 as suggested by [4]. In LDCE, the regularization parameter λ and the parameter β are respectively selected from $\{2^{-4}, \cdots, 2^{4}\}$ and $\{0.1, 0.2, \cdots, 0.9\}$ via cross validation according to [6]. In PUSB, the value of $r(\mathbf{x}) = \frac{P(\mathbf{x}|y=1,s=1)}{P(\mathbf{x})}$ is estimated via minimizing the pseudo classification risk. In PWE, all data features are deemed as "propensity attributes" to estimate the propensity scores[3]. Among the incorporated compared methods, uPU, nnPU and LDCE require the class prior $P(Y = 1)$, and here we simply input the real value of $P(Y = 1)$ to these approaches. Besides, the loss functions employed by the compared methods are indicated by the corresponding papers [3], [4], [6], [21], [38], namely sigmoid loss for uPU, nnPU and PUSB, hinge loss for LDCE, and cross-entropy loss for PWE.

The experimental results are reported in Table 2. As we can see, our approach achieves better or comparable performance when compared with the remaining baseline methods in most cases. Generally, LBE-MLP and LBE-LF are the best two methods among the compared methodologies. The nonlinear LBE-MLP is slightly better than the linear LBE-LF in most cases as the network in LBE-MLP can produce more flexible classifier than the logistic function in LBE-LF. Besides, we observe that the methods that consider the labeling bias (*e.g.*, LBE, PUSB and PWE) can usually obtain higher classification accuracy than those that do not take the labeling bias into account (*e.g.*, uPU, nnPU and LDCE), and this again shows the necessity of modeling the biased positive data selection in instance-dependent PU learning.

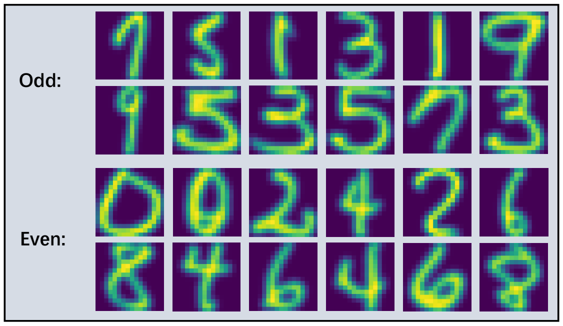

Fig. 5: Examples of even and odd images in the adopted *USPS* dataset.

Fig. 6: Examples of fighting and non-fighting video frames in *HockeyFight* dataset.

6.3 Real-World Datasets

To further evaluate the ability of LBE in handling complex problems in reality, we conduct experiments on three real-world datasets in this section, namely *USPS*[4], *HockeyFight*[5], and *Swis-*

[3] In [21], the PWE method requires a subset of the original features to be propensity attributes, so the authors conduct clustering on the dataset and then assign the additional artificial binary propensity attributes to the data points in the clusters according to some distribution. Here we do not use this propensity attributes generation strategy as this will modify the original datasets and make the comparison setting for various methods not consistent.

[4] https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/multiclass.html#usps

[5] http://visilab.etsii.uclm.es/personas/oscar/FightDetection/index.html

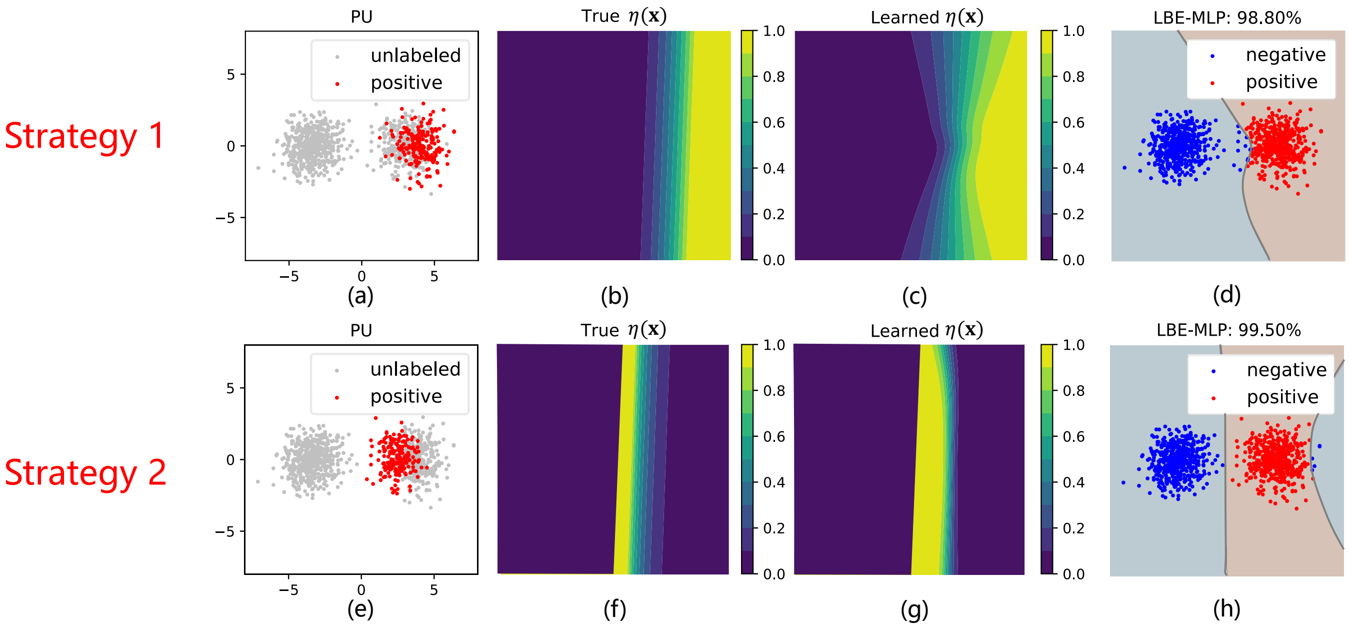

Fig. 4: The performances of LBE-MLP on the synthetic dataset under Strategies 1 and 2. The upper panel refers to Strategy 1, where (a)∼(d) show the unlabeled and biased positive data for model training, true $\eta(\mathbf{x})$, the estimated $\eta(\mathbf{x})$ by LBE-MLP, and the classification result of LBE-MLP, respectively. The lower panel refers to Strategy 2, where (e)∼(h) have the same meanings as (a)∼(d).

sProt [6].

The *USPS* dataset is a typical benchmark dataset for handwritten digit recognition. It contains 9298 digit images of size 16×16 that are classified into 10 categories, *i.e.*, "0"∼"9". In this paper, every image is represented by a $16 \times 16 = 256$-dimensional feature vector with the elements being the gray values of image pixels. The images of odd numbers are chosen as positive data, and the images of even numbers are taken as negative ones. Some examples of this dataset are shown in Fig. 5.

The *HockeyFight* dataset is a challenging dataset for detecting violent behaviors in various ice hockey games. It is made up of 1000 video clips collected from different ice hockey games, in which 500 clips are with fighting behavior (*i.e.*, positive class) and 500 clips are without fighting behavior (*i.e.*, negative class). Similar to [51], we first apply the space-time interest point (STIP) and motion SIFT as action descriptors, and then transform each video clip into a histogram over 100 visual words by further using the Bag-of-Words quantization. As a result, each clip can be characterized by a 100-dimensional feature vector. Fig. 6 provides some example frames of this dataset.

The *SwissProt* is a document classification dataset with 2453 human-labeled positive examples and 4906 unlabeled examples [16]. This dataset contains natural labeling bias during the data annotation process, and has been widely used to evaluate the instance-dependent PU learning approaches [38], [22]. Therefore, different from the *USPS* and *HockeyFight* datasets of which the biased positive sets S_P are artificially generated via the two sampling strategies, the observed positive data in *SwissProt* are inheritably biased. In *SwissProt* dataset, approximately 10% of the unlabeled data are actually positive. Following [38], we use Bag-of-Words technique to transform each document into a 78,894-dimensional sparse feature vector. Moreover, since the resulting data dimension is extremely high, the hidden layer dimension of MLP in LBE-MLP is set to 300.

Similar to the experimental setting in Section 6.2, for the *USPS* and *HockeyFight* datasets, we also use five-fold cross validation to evaluate the performances of compared methods on each dataset, and the mean test accuracies as well as the standard deviations are reported. Because the training set (including positive set and unlabeled set) and test set have already been specified by the original *SwissProt* dataset, so we simply report the test accuracy of each compared method for one trial. The parameter configurations of all methods are identical to those as mentioned in Section 6.2. Table 3 presents the test accuracies rendered by the compared methods, which clearly validates the top-level performance of LBE among all the other methods in all these datasets. As we can see, our method outperforms the instance-independent PU methods such as uPU, nnPU, and LDCE under different sampling strategies, revealing the effectiveness and generality of our proposed graphical model in handling instance-dependent PU learning problems. Besides, we note that on the *SwissProt* dataset with natural labeling bias, our LBE also shows better results than other methods. The above results indicate that our method can precisely capture different types of underlying labeling bias during the annotation stage.

7 Conclusion

In this paper, we proposed a new instance-dependent PU learning algorithm termed "LBE" which jointly estimates the labeling bias and learns a classifier. The advantages of LBE are four-fold:

- **Generality**. The generality of the proposed framework lies in two aspects. One one hand, LBE can accommodate to a wide range of popular classifiers such as LF and MLP presented in this paper. On the other hand, LBE can flexibly characterize various kinds of labeling bias as long as the user-defined $\eta(\mathbf{x};\boldsymbol{\theta}_2)$ is changed.
- **Optimality**. LBE can be interpreted as the rectified logistic regression with a clear objective function, of which the

[6] http://cseweb.ucsd.edu/ elkan/posonly/

TABLE 2: Comparison of test accuracies (mean±std) for our proposed method and the baselines on six UCI datasets under different sampling strategies and π's. The best and the second best results on each dataset are indicated in red and blue, respectively. The black "✓"("×") denotes that LBE-LF is siginificantly better (worse) than the corresponding methods revealed by the paired t-test with confidence level 95%. Similarly, the magenta "✓"("×") denotes that LBE-MLP is significantly better (worse) than the corresponding methods revealed by the paired t-test.

Dataset	Strategy	π	uPU [3]	nnPU [4]	LDCE [6]	PUSB [38]	PWE [21]	LBE-LF	LBE-MLP
australian	1	0.2	0.8188 ± 0.0258 ✓ ✓	0.8235 ± 0.0248 ✓	0.8105 ± 0.0078 ✓ ✓	0.8202 ± 0.0090 ✓ ✓	0.8429 ± 0.0104 ✓	0.8484 ± 0.0091	0.8641 ± 0.0057
		0.3	0.7913 ± 0.0679 ✓	0.8156 ± 0.0049 ✓ ✓	0.8116 ± 0.0063 ✓ ✓	0.8232 ± 0.0251 ✓	0.8380 ± 0.0171	0.8336 ± 0.0173	0.8577 ± 0.0152
		0.4	0.8194 ± 0.0359	0.8061 ± 0.0271 ✓ ✓	0.7652 ± 0.0022 ✓ ✓	0.8169 ± 0.0071 ✓ ✓	0.8299 ± 0.0205 ✓	0.8397 ± 0.0162	0.8525 ± 0.0095
	2	0.2	0.8099 ± 0.0487 ✓ ✓	0.7362 ± 0.0737 ✓ ✓	0.8290 ± 0.0035 ✓ ✓	0.8319 ± 0.0099 ✓ ✓	0.7939 ± 0.1262 ✓ ✓	0.8745 ± 0.0033	0.8719 ± 0.0039
		0.3	0.7728 ± 0.0387 ✓ ✓	0.8084 ± 0.0582	0.7958 ± 0.0053 ✓ ✓	0.8351 ± 0.0121 ✓	0.8310 ± 0.0049 ✓	0.8371 ± 0.0101	0.8528 ± 0.0048
		0.4	0.7849 ± 0.0519 ✓ ✓	0.7875 ± 0.0376 ✓ ✓	0.7638 ± 0.0190 ✓ ✓	0.8182 ± 0.0058 ✓	0.8235 ± 0.0299	0.8496 ± 0.0093	0.8397 ± 0.0106
madelon	1	0.2	0.6584 ± 0.0039 ✓ ✓	0.7113 ± 0.0035 ✓ ✓	0.6608 ± 0.0158 ✓ ✓	0.7647 ± 0.0022 ✓ ✓	0.7000 ± 0.0082 ✓ ✓	0.7986 ± 0.0038	0.8301 ± 0.0088
		0.3	0.6612 ± 0.0165 ✓ ✓	0.6998 ± 0.0039 ✓ ✓	0.6528 ± 0.0186 ✓ ✓	0.7345 ± 0.0086 ✓ ✓	0.6910 ± 0.0120 ✓ ✓	0.7723 ± 0.0041	0.7947 ± 0.0065
		0.4	0.6432 ± 0.0026 ✓ ✓	0.6546 ± 0.0106 ✓ ✓	0.6609 ± 0.0060 ✓ ✓	0.6813 ± 0.0048 ✓ ✓	0.6500 ± 0.0055 ✓ ✓	0.7149 ± 0.0065	0.7735 ± 0.0040
	2	0.2	0.7022 ± 0.0134 ✓ ✓	0.7242 ± 0.0045 ✓ ✓	0.7039 ± 0.0086 ✓ ✓	0.7656 ± 0.0035 × ✓	0.7099 ± 0.0106 ✓ ✓	0.7464 ± 0.0043	0.8250 ± 0.0069
		0.3	0.6776 ± 0.0053 ✓ ✓	0.6508 ± 0.0121 ✓ ✓	0.6504 ± 0.0102 ✓ ✓	0.7174 ± 0.0047 ✓	0.6300 ± 0.0152 ✓ ✓	0.7199 ± 0.0092	0.7705 ± 0.0094
		0.4	0.6276 ± 0.0052 ✓ ✓	0.6124 ± 0.0031 ✓ ✓	0.6264 ± 0.0133 ✓ ✓	0.6809 ± 0.0079 ✓	0.5500 ± 0.0258 ✓ ✓	0.6811 ± 0.0047	0.7211 ± 0.0089
phishing	1	0.2	0.9325 ± 0.0018 ✓	0.8817 ± 0.0023 ✓ ✓	0.8960 ± 0.0199 ✓ ✓	0.9022 ± 0.0017 ✓ ✓	0.9295 ± 0.0027 ✓ ✓	0.9341 ± 0.0003	0.9373 ± 0.0024
		0.3	0.9311 ± 0.0002 ✓ ✓	0.8777 ± 0.0089 ✓ ✓	0.9027 ± 0.0071 ✓ ✓	0.9001 ± 0.0029 ✓ ✓	0.9319 ± 0.0016 ✓	0.9336 ± 0.0009	0.9394 ± 0.0016
		0.4	0.9311 ± 0.0003 ✓ ✓	0.8889 ± 0.0071 ✓ ✓	0.8737 ± 0.0197 ✓ ✓	0.9027 ± 0.0014 ✓ ✓	0.9325 ± 0.0025 ✓	0.9344 ± 0.0016	0.9411 ± 0.0006
	2	0.2	0.9255 ± 0.0013 ✓ ✓	0.8903 ± 0.0033 ✓ ✓	0.8935 ± 0.0051 ✓ ✓	0.9090 ± 0.0010 ✓ ✓	0.9330 ± 0.0005 ✓ ✓	0.9379 ± 0.0001	0.9440 ± 0.0041
		0.3	0.9245 ± 0.0005 ✓ ✓	0.9045 ± 0.0036 ✓ ✓	0.8535 ± 0.0301 ✓ ✓	0.9103 ± 0.0009 ✓ ✓	0.9320 ± 0.0011 ✓ ✓	0.9378 ± 0.0001	0.9468 ± 0.0020
		0.4	0.9280 ± 0.0011 ✓ ✓	0.9127 ± 0.0022 ✓ ✓	0.9152 ± 0.0041 ✓ ✓	0.8873 ± 0.0206 ✓ ✓	0.9322 ± 0.0015 ✓ ✓	0.9368 ± 0.0006	0.9445 ± 0.0035
vote	1	0.2	0.9191 ± 0.0038 ✓ ✓	0.8814 ± 0.0245 ✓ ✓	0.9014 ± 0.0034 ✓ ✓	0.9056 ± 0.0017 ✓ ✓	0.9568 ± 0.0055 ×	0.9389 ± 0.0026	0.9563 ± 0.0043
		0.3	0.9131 ± 0.0073 ✓ ✓	0.8690 ± 0.0056 ✓ ✓	0.8754 ± 0.0168 ✓ ✓	0.9049 ± 0.0033 ✓ ✓	0.9131 ± 0.0451 ✓ ✓	0.9439 ± 0.0097	0.9513 ± 0.0050
		0.4	0.9030 ± 0.0070 ✓	0.8634 ± 0.0179 ✓ ✓	0.8749 ± 0.0151 ✓ ✓	0.9064 ± 0.0026 ✓	0.8947 ± 0.0225 ✓	0.9269 ± 0.0401	0.9393 ± 0.0107
	2	0.2	0.9389 ± 0.0084 ✓	0.8648 ± 0.0208 ✓✓	0.8644 ± 0.0137 ✓ ✓	0.9079 ± 0.0033 ✓ ✓	0.9103 ± 0.0787 ✓ ✓	0.9701 ± 0.0002	0.9513 ± 0.0435
		0.3	0.9494 ± 0.0040 ✓ ✓	0.8814 ± 0.0187 ✓ ✓	0.8653 ± 0.0053 ✓ ✓	0.9079 ± 0.0043 ✓ ✓	0.9655 ± 0.0036 ✓	0.9687 ± 0.0013	0.9710 ± 0.0031
		0.4	0.9499 ± 0.0050	0.8998 ± 0.0066 ✓ ✓	0.8805 ± 0.0118 ✓ ✓	0.9273 ± 0.0086 ✓	0.9430 ± 0.0376 ✓ ✓	0.9595 ± 0.0198	0.9513 ± 0.0450
banknote	1	0.2	0.9000 ± 0.0630 ✓ ✓	0.8994 ± 0.0161 ✓ ✓	0.8713 ± 0.0025 ✓ ✓	0.8203 ± 0.0282 ✓ ✓	0.8774 ± 0.0554 ✓ ✓	0.9652 ± 0.0238	0.9797 ± 0.0066
		0.3	0.8299 ± 0.0813 ✓ ✓	0.8010 ± 0.0112 ✓ ✓	0.8571 ± 0.0011 ✓ ✓	0.8134 ± 0.0355 ✓ ✓	0.9096 ± 0.0444 ✓ ✓	0.9638 ± 0.0117	0.9784 ± 0.0061
		0.4	0.8299 ± 0.0813 ✓ ✓	0.8010 ± 0.0112 ✓ ✓	0.8671 ± 0.0034 ✓ ✓	0.8134 ± 0.0355 ✓ ✓	0.9096 ± 0.0444 ✓ ✓	0.9638 ± 0.0117	0.9784 ± 0.0061
	2	0.2	0.8872 ± 0.0161 ✓ ✓	0.9523 ± 0.0009 ✓ ✓	0.9413 ± 0.0078 ✓ ✓	0.9619 ± 0.0060 ✓	0.9452 ± 0.0258 ✓ ✓	0.9708 ± 0.0040	0.9668 ± 0.0121
		0.3	0.9210 ± 0.0211 ✓ ✓	0.9708 ± 0.0090	0.9261 ± 0.0197 ✓ ✓	0.9729 ± 0.0066	0.9194 ± 0.0759 ✓ ✓	0.9758 ± 0.0029	0.9765 ± 0.0107
		0.4	0.9633 ± 0.0102 ✓ ✓	0.9669 ± 0.0083 ✓	0.9672 ± 0.0025 ✓	0.9256 ± 0.0118 ✓ ✓	0.9599 ± 0.0314 ✓ ✓	0.9742 ± 0.0068	0.9800 ± 0.0030
breast	1	0.2	0.9628 ± 0.0061 ✓ ✓	0.9556 ± 0.0023 ✓ ✓	0.9503 ± 0.0068 ✓ ✓	0.9628 ± 0.0008 ✓ ✓	0.9672 ± 0.0034 ✓	0.9698 ± 0.0013	0.9716 ± 0.0017
		0.3	0.9687 ± 0.0008 ✓ ✓	0.9643 ± 0.0063 ✓ ✓	0.9529 ± 0.0077 ✓ ✓	0.9506 ± 0.0019 ✓ ✓	0.9613 ± 0.0120 ✓ ✓	0.9728 ± 0.0018	0.9739 ± 0.0016
		0.4	0.9698 ± 0.0017	0.9672 ± 0.0056	0.9585 ± 0.0028 ✓ ✓	0.9540 ± 0.0030 ✓ ✓	0.9567 ± 0.0290 ✓ ✓	0.9707 ± 0.0029	0.9716 ± 0.0013
	2	0.2	0.9698 ± 0.0013 ✓ ✓	0.9523 ± 0.0023 ✓ ✓	0.9617 ± 0.0031 ✓ ✓	0.9672 ± 0.0052 ✓ ✓	0.9363 ± 0.1432 ✓ ✓	0.9760 ± 0.0008	0.9783 ± 0.0019
		0.3	0.9675 ± 0.0024 ✓ ✓	0.9548 ± 0.0035 ✓ ✓	0.9567 ± 0.0024 ✓ ✓	0.9716 ± 0.0008 ✓ ✓	0.9425 ± 0.1756 ✓ ✓	0.9766 ± 0.0004	0.9786 ± 0.0008
		0.4	0.9701 ± 0.0013 ✓ ✓	0.9654 ± 0.0027 ✓ ✓	0.9645 ± 0.0043 ✓ ✓	0.9540 ± 0.0005 ✓ ✓	0.9513 ± 0.0008 ✓ ✓	0.9751 ± 0.0022	0.9769 ± 0.0007

TABLE 3: Comparison of test accuracies for our proposed method and the baselines on three real-world datasets including *HockeyFight*, *USPS*, and *SwiffProt*. The best two results on each dataset are indicated in red and blue, respectively. The black "✓"("×") denotes that LBE-LF is siginificantly better (worse) than the corresponding methods revealed by the paired t-test with confidence level 95%. Similarly, the magenta "✓"("×") denotes that LBE-MLP is significantly better (worse) than the corresponding methods revealed by the paired t-test.

Dataset	Strategy	π	uPU [3]	nnPU [4]	LDCE [6]	PUSB [38]	PWE [21]	LBE-LF	LBE-MLP
HockeyFight	1	0.2	0.8524 ± 0.0022 ✓ ✓	0.8738 ± 0.0053 ✓ ✓	0.8664 ± 0.0047 ✓ ✓	0.8756 ± 0.0033 ✓ ✓	0.8402 ± 0.0193 ✓ ✓	0.9020 ± 0.0099	0.9236 ± 0.0043
		0.3	0.8502 ± 0.0034 ✓ ✓	0.8764 ± 0.0071 ✓	0.8656 ± 0.0041 ✓ ✓	0.8820 ± 0.0032 ✓	0.8546 ± 0.0018 ✓ ✓	0.8800 ± 0.0107	0.9102 ± 0.0036
		0.4	0.8962 ± 0.0027 ✓	0.8816 ± 0.0050 ✓ ✓	0.8616 ± 0.0089 ✓ ✓	0.8760 ± 0.0014 ✓ ✓	0.8472 ± 0.2017 ✓ ✓	0.8996 ± 0.0078	0.9030 ± 0.0040
	2	0.2	0.9086 ± 0.0019 ✓	0.8850 ± 0.0034 ✓ ✓	0.8476 ± 0.0168 ✓ ✓	0.8820 ± 0.0014 ✓ ✓	0.8200 ± 0.0023 ✓ ✓	0.8994 ± 0.0062	0.9526 ± 0.0029
		0.3	0.8996 ± 0.0030 ✓ ✓	0.8912 ± 0.0019 ✓ ✓	0.8442 ± 0.0104 ✓ ✓	0.8884 ± 0.0062 ✓ ✓	0.7828 ± 0.0051 ✓ ✓	0.9262 ± 0.0008	0.9494 ± 0.0042
		0.4	0.8940 ± 0.0051 ✓ ✓	0.8968 ± 0.0038 ✓ ✓	0.8400 ± 0.0129 ✓ ✓	0.8884 ± 0.0068 ✓ ✓	0.8072 ± 0.0017 ✓ ✓	0.9274 ± 0.0011	0.9436 ± 0.0035
USPS	1	0.2	0.9058 ± 0.0059 ✓	0.8273 ± 0.0249 ✓ ✓	0.9064 ± 0.0037 ✓	0.8276 ± 0.0044 ✓ ✓	0.8977 ± 0.0117 ✓ ✓	0.9171 ± 0.0104	0.9266 ± 0.0044
		0.3	0.9080 ± 0.0061 ✓	0.8431 ± 0.0080 ✓ ✓	0.8603 ± 0.0064 ✓ ✓	0.8375 ± 0.0085 ✓ ✓	0.9141 ± 0.0147	0.9246 ± 0.0148	0.9275 ± 0.0071
		0.4	0.9072 ± 0.0060 ✓ ✓	0.8529 ± 0.0064 ✓ ✓	0.8836 ± 0.0045 ✓ ✓	0.8414 ± 0.0128 ✓ ✓	0.9187 ± 0.0075	0.9232 ± 0.0108	0.9230 ± 0.0084
	2	0.2	0.9033 ± 0.0160 ✓ ✓	0.8541 ± 0.0320 ✓ ✓	0.8825 ± 0.0016 ✓ ✓	0.9002 ± 0.0236 ✓ ✓	0.9119 ± 0.0085 ✓ ✓	0.9380 ± 0.0178	0.9374 ± 0.0175
		0.3	0.8815 ± 0.0153 ✓ ✓	0.8413 ± 0.0347 ✓ ✓	0.8601 ± 0.0120 ✓ ✓	0.8916 ± 0.0151 ✓ ✓	0.9135 ± 0.0064 ✓	0.9234 ± 0.0084	0.9429 ± 0.0148
		0.4	0.8796 ± 0.0188 ✓ ✓	0.8723 ± 0.0256 ✓ ✓	0.8897 ± 0.0030 ✓ ✓	0.8879 ± 0.0275 ✓ ✓	0.9166 ± 0.0062 ✓ ✓	0.9386 ± 0.0146	0.9289 ± 0.0102
SwiffProt	-	-	0.9256	0.9450	0.9174	0.9216	0.9436	0.9477	0.9581

existence and local uniqueness of the solution have been theoretically proved.

- **Generalizability**. The obtained LBE model is theoretically proved to generalize well on unseen data, as the expected risk will converge to the empirical risk if the amounts of positive and unlabeled data are sufficiently large.
- **Practicability**. Unlike many existing PU classifiers that should pre-estimate the class prior $P(y=1)$, LBE does not require this prior knowledge which is practically non-trivial to obtain. Besides, the LBE model does not contain any tuning hyperparameters. Therefore, it can be easily implemented under various practical scenarios.

Due to the above reasons, our method has shown superior performance to various state-of-the-art PU learning approaches on typical benchmark and real-world datasets.

For the future work, although our LBE is designed under the setting of single-training-set PU learning, it would be interesting to find a way to adapt our method to case-control PU learning. Besides, it is also worthwhile to extend our LBE framework to tackle the sampling bias-inherited semi-supervised learning [52], [53] and label noise learning [19], [23] problems.

8 Appendix

This section provides the proofs for some key lemmas and theorems in the main body.

8.1 Proof of Theorem 3

For the simplicity of presentation, in the following proof of Theorem 3, we use $\mathcal{L}$, h_i, $\bar{h}_i$, η_i to abbreviate $\mathcal{L}(\boldsymbol{\theta})$, $h(\mathbf{x}_i;\boldsymbol{\theta}_1)$, $\bar{h}(\mathbf{x}_i;\boldsymbol{\theta}_1)$ and $\eta(\mathbf{x}_i;\boldsymbol{\theta}_2)$, respectively. According to Proposition 2, we should investigate the first-order derivative and Hessian matrix of the log-likelihood function Eq. (19) to $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. Specifically, the first-order derivative of $\mathcal{L}(\boldsymbol{\theta})$ to $\boldsymbol{\theta}_1$ is in the form of

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_1} &= \sum_{i=1}^{n}(1-s_i)\frac{\eta_i h_i(1-h_i)}{1-\eta_i h_i}\mathbf{x}_i + s_i(h_i-1)\mathbf{x}_i \\ &= \sum_{i=1}^{n}(1-s_i)\frac{(\eta_i-1)h_i}{1-\eta_i h_i}\mathbf{x}_i + (1-s_i)h_i\mathbf{x}_i + s_i(h_i-1)\mathbf{x}_i,\end{aligned} \tag{34}$$

and the Hessian matrix is consequently computed as

$$\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1} = \mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P} + \mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_U} \tag{35}$$

where

$$\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P} = \sum_{i=1}^{k} \frac{(1-s_i)\eta_i + s_i + \eta_i^2 h_i^2 - 2\eta_i h_i}{(1-\eta_i h_i)^2} h_i(h_i-1)\mathbf{x}_i\mathbf{x}_i^{\top}; \tag{36}$$

$$\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_U} = \sum_{i=k+1}^{n} \frac{(1-s_i)\eta_i + s_i + \eta_i^2 h_i^2 - 2\eta_i h_i}{(1-\eta_i h_i)^2} h_i(h_i-1)\mathbf{x}_i\mathbf{x}_i^{\top}. \tag{37}$$

Therefore, next we should study the negative definitiveness of $\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}$ in Eq. (35). For the $\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P}$ part corresponding to $\mathbf{x}_i \in S_P$, we know $s_i = 1$, so Eq. (36) degenerates to

$$\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P} = \sum_{i=1}^{k} h_i(h_i-1)\mathbf{x}_i\mathbf{x}_i^{\top} = \mathbf{X}_P \mathbf{D}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P} \mathbf{X}_P^{\top}, \tag{38}$$

where $\mathbf{X}_P = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k) \in \mathbb{R}^{d\times k}$ is positive data matrix with each column representing an example, and $\mathbf{D}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P}$ is a $k \times k$ diagonal matrix with the i-th ($i = 1, \cdots, k$) diagonal elements being $h_i(h_i-1) \leq 0$. To demonstrate that the Hessian matrix $\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P}$ corresponding to the positive examples is negative semi-definite, we consider its opposite, *i.e.*, $-\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P}$. Given any d-dimensional column vector $\mathbf{v} \neq \mathbf{0}$, we have $\mathbf{v}^{\top}(-\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P})\mathbf{v} = \mathbf{v}^{\top}\mathbf{X}_P(-\mathbf{D}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P})\mathbf{X}_P^{\top}\mathbf{v} = \mathbf{v}^{\top}\mathbf{X}_P(-\mathbf{D}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P})^{1/2}\big((-\mathbf{D}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P})^{1/2}\big)^{\top}\mathbf{X}_P^{\top}\mathbf{v} = \left\|(-\mathbf{D}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P})^{1/2}\mathbf{X}_P^{\top}\mathbf{v}\right\|_2^2 \geq 0$, which indicates the positive semi-definitiveness of $-\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P}$. Therefore, $\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_P}$ is negative semi-definite.

For the $\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_U}$ part corresponding to $\mathbf{x}_i \in S_U$, since $s_i = 0$, Eq. (37) becomes

$$\begin{aligned}\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_U} &= \sum_{i=k+1}^{n} \frac{\eta_i^2\left[(h_i - \frac{1}{\eta_i})^2 + \frac{\eta_i - 1}{\eta_i^2}\right]}{(1-\eta_i h_i)^2} h_i(h_i-1)\mathbf{x}_i\mathbf{x}_i^{\top} \\ &= \mathbf{X}_U \mathbf{D}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_U} \mathbf{X}_U^{\top},\end{aligned} \tag{39}$$

where $\mathbf{X}_U = (\mathbf{x}_{k+1}, \mathbf{x}_{k+2}, \cdots, \mathbf{x}_n) \in \mathbb{R}^{d\times(n-k)}$ is unlabeled data matrix similar to $\mathbf{X}_P$, and $\mathbf{D}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_U}$ is an $(n–k)\times(n–k)$ diagonal matrix with the i-th ($i = k+1, \cdots, n$) diagonal elements being $\eta_i^2\left[(h_i - \frac{1}{\eta_i})^2 + \frac{\eta_i-1}{\eta_i^2}\right] h_i(h_i-1)/(1-\eta_i h_i)^2$. Here we utilize the fact that $\eta_i \neq 0$ as in this case the labels of all positive examples will be unobserved, which obviously violates the setting of PU learning. By observing Eq. (39) and considering that $h_i(h_i-1)$ is always no larger than 0, we see that to make all the diagonal elements in $\mathbf{D}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_U}$ to be negative, the quadratic term regarding h_i, *i.e.*, $\left[(h_i - \frac{1}{\eta_i})^2 + \frac{\eta_i-1}{\eta_i^2}\right]$, should be positive. That is to say, h_i has to satisfy $h_i < (1-\sqrt{1-\eta_i})/\eta_i$, $\forall i = k+1, \cdots, n$. By denoting the function $f_{\eta}(\eta_i) = (1-\sqrt{1-\eta_i})/\eta_i$, we see that $f_{\eta}(\eta_i)$ is monotonically increasing when $\eta_i \in (0,1)$, and thus $f_{\eta}(\eta_i) > \lim_{\eta_i\to 0}(1-\sqrt{1-\eta_i})/\eta_i = 0.5$.

On the other hand, by noting that $0 < \eta_i < 1$, $0 \leq h_i \leq 1$, and recalling that $\bar{h}_i > 2h_i - 1$, we have $h_i\eta_i > 2h_i - 1$, which leads to $0 < \frac{h_i(1-\eta_i)}{1-h_i\eta_i} < 0.5$, and this indicates that

$$\begin{aligned}&\frac{h_i(1-\eta_i)}{1-h_i\eta_i} \\ =&\frac{h_i(1-\eta_i)}{1-[h_i\eta_i + (1-h_i)\cdot 0]} \\ =&\frac{P(y_i=1|\mathbf{x}_i)(1-P(s_i=1|y_i=1,\mathbf{x}_i))}{1-\big(P(y_i=1|\mathbf{x}_i)P(s_i=1|y_i=1,\mathbf{x}_i)+P(y_i=0|\mathbf{x}_i)P(s_i=1|y_i=0,\mathbf{x}_i)\big)} \\ =&\frac{P(y_i=1|\mathbf{x}_i)(1-P(s_i=1|y_i=1,\mathbf{x}_i))}{1-\big(P(s_i=1,y_i=1|\mathbf{x}_i)+P(s_i=1,y_i=0|\mathbf{x}_i)\big)} \\ =&\frac{P(y_i=1|\mathbf{x}_i)(1-P(s_i=1|y_i=1,\mathbf{x}_i))}{1-P(s_i=1|\mathbf{x}_i)} \\ =&\frac{P(y_i=1|\mathbf{x}_i)P(s_i=0|y_i=1,\mathbf{x}_i)}{P(s_i=0|\mathbf{x}_i)} \\ =&P(y_i=1|s_i=0,\mathbf{x}_i) \\ <&0.5.\end{aligned} \tag{40}$$

Consequently, we know that $h_i < (1-\sqrt{1-\eta_i})/\eta_i$ holds true when $\mathbf{x}_i \in S_U$. Therefore, the Hessian matrix $\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}^{S_U}$ regarding unlabeled set is negative semi-definite. As a result, the Hessian matrix $\mathbf{H}_{\mathcal{L},\boldsymbol{\theta}_1}$ in Eq. (35) is negative semi-definite as the sum of two negative semi-definite matrices is still negative semi-definite. Therefore, the entire log-likelihood function $\mathcal{L}$ is concave regarding $\boldsymbol{\theta}_1$.

Furthermore, by observing the equivalent log-likelihood function Eq. (19), we see that h_i and η_i are symmetrical and exchangeable, so the concavity of Eq. (19) regarding $\boldsymbol{\theta}_2$ can be similarly proved by invoking $\eta(\mathbf{x}_i;\boldsymbol{\theta}_2) = P(s_i = 1|y_i = 1, \mathbf{x}_i) < 0.5$ when $i = k+1, \cdots, n$. As a result, Theorem 3 is proved and the optimal solution $\boldsymbol{\theta}^*$ will be locally unique.

8.2 Proof of Lemma 6

Note that $h(\mathbf{x}_i)$ is the output of softmax on $h'(\mathbf{x}_i)$. Besides, we have $\bar{h}_1(\mathbf{x}_i) = \eta(\mathbf{x}_i)h_1(\mathbf{x}_i)$ and $\bar{h}_0(\mathbf{x}_i) = 1 - \bar{h}_1(\mathbf{x}_i) = 1 - \eta(\mathbf{x}_i) + \eta(\mathbf{x}_i)h_0(\mathbf{x}_i)$. Then we get the relationship $h'(\mathbf{x}_i) \xrightarrow{\text{softmax}} h(\mathbf{x}_i) \xrightarrow{\eta(\mathbf{x}_i)\circ h(\mathbf{x}_i)} \bar{h}(\mathbf{x}_i)$. Therefore, we have the following derivative results:

$$\frac{\partial l(h(\mathbf{x}_i), s_i)}{\partial \bar{h}(\mathbf{x}_i)} = \begin{pmatrix} \frac{\partial l(h(\mathbf{x}_i),s_i)}{\partial \bar{h}_0(\mathbf{x}_i)} \\ \frac{\partial l(h(\mathbf{x}_i),s_i)}{\partial \bar{h}_1(\mathbf{x}_i)} \end{pmatrix} = \begin{pmatrix} -(1-s_i)/\bar{h}_0(\mathbf{x}_i) \\ -s_i/\bar{h}_1(\mathbf{x}_i) \end{pmatrix}, \tag{41}$$

$$\frac{\partial \bar{h}(\mathbf{x}_i)}{\partial h(\mathbf{x}_i)} = \begin{pmatrix} \frac{\partial \bar{h}_0(\mathbf{x}_i)}{\partial h_0(\mathbf{x}_i)} & \frac{\partial \bar{h}_1(\mathbf{x}_i)}{\partial h_0(\mathbf{x}_i)} \\ \frac{\partial \bar{h}_0(\mathbf{x}_i)}{\partial h_1(\mathbf{x}_i)} & \frac{\partial \bar{h}_1(\mathbf{x}_i)}{\partial h_1(\mathbf{x}_i)} \end{pmatrix} = \begin{pmatrix} \eta(\mathbf{x}_i) & 0 \\ 0 & \eta(\mathbf{x}_i) \end{pmatrix}, \tag{42}$$

$$\begin{aligned}\frac{\partial h(\mathbf{x}_i)}{\partial h'(\mathbf{x}_i)} &= \begin{pmatrix} \frac{\partial h_0(\mathbf{x}_i)}{\partial h'_0(\mathbf{x}_i)} & \frac{\partial h_1(\mathbf{x}_i)}{\partial h'_0(\mathbf{x}_i)} \\ \frac{\partial h_0(\mathbf{x}_i)}{\partial h'_1(\mathbf{x}_i)} & \frac{\partial h_1(\mathbf{x}_i)}{\partial h'_1(\mathbf{x}_i)} \end{pmatrix} \\ &= \begin{pmatrix} h_0(\mathbf{x}_i)h_1(\mathbf{x}_i) & -h_0(\mathbf{x}_i)h_1(\mathbf{x}_i) \\ -h_0(\mathbf{x}_i)h_1(\mathbf{x}_i) & h_0(\mathbf{x}_i)h_1(\mathbf{x}_i) \end{pmatrix},\end{aligned} \tag{43}$$

where we use the facts that $h_1(\mathbf{x}_i) = 1 - h_0(\mathbf{x}_i)$ and the derivative of the softmax function is $\frac{\partial h_m(\mathbf{x}_i)}{\partial h'_j(\mathbf{x}_i)} = -h_m(\mathbf{x}_i)h_j(\mathbf{x}_i)$ for $m \neq j$.

For $s_i = 1$, according to the chain rule and Eqs. (41), (42), (43), we have

$$\begin{aligned}\frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h_1(\mathbf{x}_i)} &= \frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial \bar{h}_1(\mathbf{x}_i)}\frac{\partial \bar{h}_1(\mathbf{x}_i)}{\partial h_1(\mathbf{x}_i)} + \frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial \bar{h}_0(\mathbf{x}_i)}\frac{\partial \bar{h}_0(\mathbf{x}_i)}{\partial h_1(\mathbf{x}_i)} \\ &= -\eta(\mathbf{x}_i)/\bar{h}_1(\mathbf{x}_i)\end{aligned} \tag{44}$$

and

$$\begin{aligned}\frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h_0(\mathbf{x}_i)} &= \frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial \bar{h}_1(\mathbf{x}_i)}\frac{\partial \bar{h}_1(\mathbf{x}_i)}{\partial h_0(\mathbf{x}_i)} + \frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial \bar{h}_0(\mathbf{x}_i)}\frac{\partial \bar{h}_0(\mathbf{x}_i)}{\partial h_0(\mathbf{x}_i)} \\ &= 0.\end{aligned} \tag{45}$$

Therefore, we know that

$$\begin{aligned}\frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h'_1(\mathbf{x}_i)} &= \frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h_1(\mathbf{x}_i)}\frac{\partial h_1(\mathbf{x}_i)}{\partial h'_1(\mathbf{x}_i)} + \frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h_0(\mathbf{x}_i)}\frac{\partial h_0(\mathbf{x}_i)}{\partial h'_1(\mathbf{x}_i)} \\ &= -h_0(\mathbf{x}_i).\end{aligned} \tag{46}$$

Since $h_0(\mathbf{x}_i)$ is within $[0, 1]$, $\left|\frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h'_j(\mathbf{x}_i)}\right| < 1$ holds when $s_i = 1$.

For $s_i = 0$, we similarly have

$$\frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h_1(\mathbf{x}_i)} = 0 \tag{47}$$

and

$$\frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h_0(\mathbf{x}_i)} = -\eta(\mathbf{x}_i)/\bar{h}_0(\mathbf{x}_i). \tag{48}$$

Therefore, we know that

$$\begin{aligned}\frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h'_0(\mathbf{x}_i)} &= \frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h_1(\mathbf{x}_i)}\frac{\partial h_1(\mathbf{x}_i)}{\partial h'_0(\mathbf{x}_i)} + \frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h_0(\mathbf{x}_i)}\frac{\partial h_0(\mathbf{x}_i)}{\partial h'_0(\mathbf{x}_i)} \\ &= -\eta(\mathbf{x}_i)h_0(\mathbf{x}_i)h_1(\mathbf{x}_i)/\bar{h}_0(\mathbf{x}_i).\end{aligned} \tag{49}$$

Since $h_0(\mathbf{x}_i)$ and $\eta(\mathbf{x}_i)$ are within $[0, 1]$, we have $\bar{h}_0(\mathbf{x}_i) = 1 - \eta(\mathbf{x}_i) + \eta(\mathbf{x}_i)h_0(\mathbf{x}_i) = (1 - \eta(\mathbf{x}_i))(1 - h_0(\mathbf{x}_i)) + h_0(\mathbf{x}_i) \geq h_0(\mathbf{x}_i)$. By further noting that $h_1(\mathbf{x}_i)$ and $\bar{h}_0(\mathbf{x}_i)$ in Eq. (49) are also within $[0, 1]$, we conclude that $\left|\frac{\partial \ell(\bar{h}(\mathbf{x}_i), s_i)}{\partial h'_j(\mathbf{x}_i)}\right| < 1$ holds when $s_i = 0$. By taking both the cases of $s_i = 0$ and $s_i = 1$ into consideration, Lemma 6 is proved.

8.3 Proof of Lemma 7

To prove Lemma 7, we need the following existing result:

Lemma 11. *(Talagrand contraction Lemma, [48]) If $\ell : \mathbb{R} \to \mathbb{R}$ is Ω-Lipschitz continuous and satisfies $\ell(0) = 0$, then*

$$\hat{\mathcal{R}}_S(\ell \circ \mathcal{H}) \leq \Omega \hat{\mathcal{R}}_S(\mathcal{H}). \tag{50}$$

Therefore, Lemma 7 can be proved according to the following derivations:

$$\begin{aligned}\hat{\mathcal{R}}_S(\ell \circ \mathcal{H}) &= \mathbb{E}_r[\sup_{\bar{h} \in \mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} r_i \ell(\bar{h}(\mathbf{x}_i), s_i)] \\ &\overset{1}{\leq} \mathbb{E}_r[\sup_{h} \frac{1}{n}\sum_{i=1}^{n} r_i h(\mathbf{x}_i)] \\ &= \mathbb{E}_r[\sup_{\arg\max\{h'_0, h'_1\}} \frac{1}{n}\sum_{i=1}^{n} r_i h(\mathbf{x}_i)] \\ &= \mathbb{E}_r[\sup_{h'_j \in \mathcal{H}'} \frac{1}{n}\sum_{i=1}^{n} r_i h'_j(\mathbf{x}_i)] \\ &= \mathbb{E}_r[\sup_{h' \in \mathcal{H}'} \frac{1}{n}\sum_{i=1}^{n} r_i h'(\mathbf{x}_i)],\end{aligned} \tag{51}$$

where the 1^{st} inequality is according to Lemma 11.

References

[1] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in *International Conference on Machine Learning*, 2002, pp. 387–394.

[2] X.-L. Li and B. Liu, "Learning from positive and unlabeled examples with different data distributions," in *European Conference on Machine Learning*, 2005, pp. 218–229.

[3] M. Du Plessis, G. Niu, and M. Sugiyama, "Convex formulation for learning from positive and unlabeled data," in *International Conference on Machine Learning*, 2015, pp. 1386–1394.

[4] R. Kiryo, G. Niu, M. C. du Plessis, and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," in *Advances in Neural Information Processing Systems*, 2017, pp. 1674–1684.

[5] H. Shi, S. Pan, J. Yang, and C. Gong, "Positive and unlabeled learning via loss decomposition and centroid estimation." in *International Joint Conferences on Artificial Intelligence*, 2018, pp. 2689–2695.

[6] C. Gong, H. Shi, T. Liu, C. Zhang, J. Yang, and D. Tao, "Loss decomposition and centroid estimation for positive and unlabeled learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[7] C. Gong, T. Liu, J. Yang, and D. Tao, "Large-margin label-calibrated support vector machines for positive and unlabeled learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2019.

[8] C. Zhang, D. Ren, T. Liu, J. Yang, and C. Gong, "Positive and unlabeled learning with label disambiguation," in *International Joint Conference on Artificial Intelligence*, 2019, pp. 4250–4256.

[9] E. Sansone, F. G. De Natale, and Z.-H. Zhou, "Efficient training for positive unlabeled learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[10] J. Zhang, Z. Wang, J. Yuan, and Y.-P. Tan, "Positive and unlabeled learning for anomaly detection with multi-features," in *Proceedings of the 25th ACM International Conference on Multimedia*. ACM, 2017, pp. 854–862.

[11] J. Zhang, Z. Wang, J. Meng, Y. Tan, and J. Yuan, "Boosting positive and unlabeled learning for anomaly detection with multi-features," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1332–1344, 2019.

[12] P. Yang, X.-L. Li, J.-P. Mei, C.-K. Kwoh, and S.-K. Ng, "Positive-unlabeled learning for disease gene identification," *Bioinformatics*, vol. 28, no. 20, pp. 2640–2647, 2012.

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

[14] C. Gong, H. Shi, J. Yang, J. Yang, and J. Yang, "Multi-manifold positive and unlabeled learning for visual analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[15] G. Ward, T. Hastie, S. Barry, J. Elith, and J. R. Leathwick, "Presence-only data and the em algorithm," *Biometrics*, vol. 65, no. 2, pp. 554–563, 2009.

[16] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proceedings of The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 213–220.

[17] A. Menon, B. C. Rooyen, C. S. Ong, and B. Williamson, “Learning from corrupted binary labels via class-probability estimation,” in *International Conference on Machine Learning*, 2015, pp. 125–134.

[18] W. S. Lee and B. Liu, “Learning with positive and unlabeled examples using weighted logistic regression,” in *International Conference on Machine Learning*, vol. 3, 2003, pp. 448–455.

[19] A. K. Menon, B. van Rooyen, and N. Natarajan, “Learning from binary labels with instance-dependent noise,” *Machine Learning*, vol. 107, no. 8-10, pp. 1561–1595, 2018.

[20] A. Liu and B. Ziebart, “Robust classification under sample selection bias,” in *Advances in Neural Information Processing Systems*, 2014, pp. 37–45.

[21] J. Bekker, P. Robberecht, and J. Davis, “Beyond the selected completely at random assumption for learning from positive and unlabeled data,” in *ECML-PKDD*, 2019, pp. 1–16.

[22] F. He, T. Liu, G. I. Webb, and D. Tao, “Instance-dependent PU learning by bayesian optimal relabeling,” *arXiv preprint arXiv:1808.02180*, 2018.

[23] J. Cheng, T. Liu, K. Ramamohanarao, and D. Tao, “Learning with bounded instance-and label-dependent label noise,” in *International Conference on Machine Learning*, 2020, pp. 1–11.

[24] J. Bekker and J. Davis, “Estimating the class prior in positive and unlabeled data through decision tree induction,” in *AAAI Conference on Artificial Intelligence*, 2018, pp. 2712–2719.

[25] M. Plessis, G. Niu, and M. Sugiyama, “Class-prior estimation for learning from positive and unlabeled data,” in *Asian Conference on Machine Learning*, 2015, pp. 221–236.

[26] C. Gong, J. Yang, J. You, and M. Sugiyama, “Centroid estimation with guaranteed efficiency: A general framework for weakly supervised learning.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[27] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu, “Building text classifiers using positive and unlabeled examples,” in *IEEE International Conference on Data Mining*, vol. 2, 2003, pp. 179–186.

[28] X. Li and B. Liu, “Learning to classify texts using positive and unlabeled data,” in *International Joint Conference on Artificial Intelligence*, vol. 3, 2003, pp. 587–592.

[29] J. J. Rocchio, “Relevance feedback in information retrieval,” *The SMART retrieval system: experiments in automatic document processing*, pp. 313–323, 1971.

[30] M. Du Plessis, G. Niu, and M. Sugiyama, “Analysis of learning from positive and unlabeled data,” in *Advances in Neural Information Processing Systems*, 2014, pp. 703–711.

[31] M. Hou, B. Chaib-Draa, C. Li, and Q. Zhao, “Generative adversarial positive-unlabeled learning,” in *International Joint Conference on Artificial Intelligence*, 2018, pp. 2255–2261.

[32] S. Jain, M. White, and P. Radivojac, “Estimating the class prior and posterior from noisy positives and unlabeled data,” in *Advances in Neural Information Processing Systems*, 2016, pp. 2693–2701.

[33] W. Gao, L. Wang, Y.-F. Li, and Z.-H. Zhou, “Risk minimization in the presence of label noise,” in *AAAI Conference on Artificial Intelligence*, 2016, pp. 1575–1581.

[34] T. Li, C.-C. Wang, Y. Ma, P. Ortal, Q. Zhao, B. Stenger, and Y. Hirate, “Learning classifiers on positive and unlabeled data with policy gradient,” in *International Conference on Data Mining*, 2019, pp. 1–10.

[35] T. Gong, G. Wang, J. Ye, Z. Xu, and M. Lin, “Margin based pu learning,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[36] J. Bekker and J. Davis, “Learning from positive and unlabeled data: a survey,” *Machine Learning*, vol. 109, no. 4, pp. 719–760, 2020.

[37] G. Li, “A survey on postive and unlabelled learning,” 2013.

[38] M. Kato, T. Teshima, and J. Honda, “Learning from positive and unlabeled data with a selection bias,” in *International Conference on Learning Representation*, 2019, pp. 1–12.

[39] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2015.

[40] G. Jacob and B.-R. Ehud, “Training deep neural-networks using a noise adaptation layer,” in *International Conference on Learning Representations*.

[41] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, “Making deep neural networks robust to label noise: A loss correction approach,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1944–1952.

[42] X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, and M. Sugiyama, “Are anchor points really indispensable in label-noise learning?” in *Advances in Neural Information Processing Systems*, 2019, pp. 6835–6846.

[43] X. Yu, T. Liu, M. Gong, and D. Tao, “Learning with biased complementary labels,” in *European Conference on Computer Vision*, 2018, pp. 68–83.

[44] T. Mäkeläinen, K. Schmidt, and G. P. Styan, “On the existence and uniqueness of the maximum likelihood estimate of a vector-valued parameter in fixed-size samples,” *The Annals of Statistics*, pp. 758–767, 1981.

[45] K. Bogdan and M. Bogdan, “On existence of maximum likelihood estimators in exponential families,” *Statistics: A Journal of Theoretical and Applied Statistics*, vol. 34, no. 2, pp. 137–149, 2000.

[46] C. Scott, “A rate of convergence for mixture proportion estimation, with application to learning from noisy labels,” in *International Conference on Artificial Intelligence and Statistics*, 2015, pp. 838–846.

[47] H. Ramaswamy, C. Scott, and A. Tewari, “Mixture proportion estimation via kernel embeddings of distributions,” in *International Conference on Machine Learning*, 2016, pp. 2052–2060.

[48] P. L. Bartlett and S. Mendelson, “Rademacher and Gaussian complexities: Risk bounds and structural results,” *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 463–482, 2002.

[49] N. Golowich, A. Rakhlin, and O. Shamir, “Size-independent sample complexity of neural networks,” in *Annual Conference On Learning Theory*, 2018, pp. 297–299.

[50] B. Settles, “Active learning literature survey,” University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[51] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, “Deformed graph Laplacian for semisupervised learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, pp. 2261–2274, 2015.

[52] F. Zhou, T. Li, H. Zhou, H. Zhu, and J. Ye, “Graph-based semi-supervised learning with non-ignorable non-response,” in *Advances in Neural Information Processing Systems*, 2019, pp. 7013–7023.

[53] Y.-G. Hsieh, G. Niu, and M. Sugiyama, “Classification from positive, unlabeled and biased negative data,” in *International Conference on Machine Learning*, 2019, pp. 1–10.

Chen Gong received his B.E. degree from East China University of Science and Technology in 2010, and the doctoral degree from the University of Technology Sydney in 2017. Currently, he is a professor with Nanjing University of Science and Technology. His research interests mainly include machine learning and data mining. He has published more than 100 technical papers at prominent journals and conferences such as JMLR, IEEE T-PAMI, IEEE T-NNLS, IEEE T-IP, CVPR, ICML, NeurIPS, AAAI, IJCAI, ICDM, etc. He also serves as the reviewer for more than 20 international journals such as AIJ, IEEE T-PAMI, IEEE T-NNLS, IEEE T-IP, and also the (S)PC member of several top-tier conferences such as ICML, NeurIPS, ICLR, CVPR, ICCV, AAAI, IJCAI, etc. He received the “Excellent Doctorial Dissertation” awarded by Chinese Association for Artificial Intelligence, and was enrolled by the “Young Elite Scientists Sponsorship Program” of CAST. He was also the recipient of “Wu Wen-Jun AI Excellent Youth Scholar Award”.

Qizhou Wang received the B.E. degree from Nanjing University of Science and Technology in 2019, and he will pursue his Ph.D. degree in Computer Science at Hong Kong Baptist University, advised by Bo Han, Chen Gong, and Tongliang Liu. His research interests include weakly-supervised learning and data mining. He has published several technical papers at prominent conferences, such as AAAI.

Tongliang Liu is currently a Lecturer (Assistant Professor) and director of the Trustworthy Machine Learning Lab with School of Computer Science at the University of Sydney. He is also a Visiting Scientist at RIKEN AIP. He is broadly interested in the fields of trustworthy machine learning and its interdisciplinary applications, with a particular emphasis on learning with noisy labels, transfer learning, adversarial learning, unsupervised learning, and statistical deep learning theory. He has published papers on various top conferences and journals, such as NeurIPS, ICML, ICLR, CVPR, ECCV, KDD, IJCAI, AAAI, IEEE TPAMI, IEEE TNNLS, IEEE TIP, and IEEE TMM. He received the ICME 2019 best paper award and nominated as the distinguish paper award candidate for IJCAI 2017. He is a recipient of Discovery Early Career Researcher Award (DECRA) from Australian Research Council (ARC); the Cardiovascular Initiative Catalyst Award by the Cardiovascular Initiative; and was named in the Early Achievers Leadboard of Engineering and Computer Science by The Australian in 2020.

Bo Han is currently an Assistant Professor of Computer Science at Hong Kong Baptist University, affiliated with HKBU AI Research Cluster. He received his Ph.D. degree in Computer Science from University of Technology Sydney (2015-2019), advised by Ivor W. Tsang and Ling Chen. His current research interests lie in machine learning, deep learning and artificial intelligence. He has published more than 20 technical papers at prominent journals and conferences such as IEEE T-NNLS, MLJ, ICML, NeurIPS, etc.

Jane You received the B.Eng. degree in electronics engineering from Xi'an Jiaotong University, Xi'an, China, in 1986, and the Ph.D. degree in computer science from La Trobe University, Melbourne, VIC, Australia, in 1992. She was a Lecturer with the University of South Australia, Adelaide SA, Australia, and a Senior Lecturer with Griffith University, Nathan, QLD, Australia, from 1993 to 2002. She is currently a Full Professor with The Hong Kong Polytechnic University, Hong Kong. Her current research interests include image processing, pattern recognition, medical imaging, biometrics computing, multimedia systems, and data mining.

Jian Yang received the PhD degree from Nanjing University of Science and Technology (NUST), on the subject of pattern recognition and intelligence systems in 2002. In 2003, he was a Postdoctoral researcher at the University of Zaragoza. From 2004 to 2006, he was a Postdoctoral Fellow at Biometrics Centre of Hong Kong Polytechnic University. From 2006 to 2007, he was a Postdoctoral Fellow at Department of Computer Science of New Jersey Institute of Technology. Now, he is a Chang-Jiang professor in the School of Computer Science and Technology of NUST. He is the author of more than 200 scientific papers in pattern recognition and computer vision. His papers have been cited over 20000 times in the Scholar Google. His research interests include pattern recognition, computer vision and machine learning. Currently, he is/was an associate editor of Pattern Recognition, Pattern Recognition Letters, IEEE Trans. Neural Networks and Learning Systems, and Neurocomputing. He is a Fellow of IAPR.

Dacheng Tao (F'15) is currently the director of the JD Explore Academy and VP at JD.com. He mainly applies statistics and mathematics to artificial intelligence and data science, and his research is detailed in one monograph and over 200 publications in prestigious journals and proceedings at leading conferences. He received the 2015 Australian Scopus-Eureka Prize, the 2018 IEEE ICDM Research Contributions Award, and the 2021 IEEE Computer Society McCluskey Technical Achievement Award. He is a fellow of the Australian Academy of Science, AAAS, ACM and IEEE.

Instance-Dependent Positive and Unlabeled Learning with Labeling Bias Estimation (Supplementary Material)

Chen Gong, *Member, IEEE,* Qizhou Wang, Tongliang Liu, *Member, IEEE,*
Bo Han, Jane You, Jian Yang, *Member, IEEE,* Dacheng Tao, *Fellow, IEEE*

1 Additional Experiments on Synthetic Datasets

In Section 6.1 of our main paper, we present the experimental results of our LBE-LF and the other compared methods under perfectly-separable case with sampling Strategies 1 and 2. Here we provide more experimental results of the compared approaches on this dataset to further study their classification ability. We mainly investigate the performance of LBE-LF in this section, as LBE-MLP may cause overfitting as illustrated in Fig. 4 of the main paper.

Firstly, we study the performances of various methods when the positive data and negative data are not perfectly-separable. To this end, we increase the variance of the two Gaussian clusters from 1 in our main paper to 1.7 here, so that the data points of the two classes are "overlapped" with each other (see Fig. 1(a) and Fig. 2(a)). The Strategies 1 and 2 are also adopted here, and the other experimental configurations are kept identical to those in the main paper. The generated PU datasets are respectively displayed in Fig. 1(c) for Strategy 1 and Fig. 2(c) for Strategy 2.

The classification results of LBE-LF and the compared methods including uPU, nnPU, LDCE, PUSB and PWE are shown in Figs. 1(e)$\sim$(j) for Strategy 1 and in Figs. 2(e)$\sim$(j) for Strategy 2. We see that the classification accuracies of all investigated methods drop when compared with the results illustrated in Figs. 2 and 3 of our main paper. This is due to that the inseparable case involves more outliers than the separable case, which misleads the training process and makes the ideal decision boundary more difficult to identify. However, our LBE can still yield very impressive results, and the accuracy is as high as 97.7% for both Strategy 1 and Strategy 2. Moreover, we plot the real labeling probability of $\eta(\mathbf{x})$ generated by Strategy 1 (Fig. 1(b)) and Strategy 2 (Fig. 2(b)), and also show the estimated $\eta(\mathbf{x})$ by our LBE (Fig. 1(d) and Fig. 2(d)). It can be observed that the estimated $\eta(\mathbf{x})$ is very close to the true $\eta(\mathbf{x})$, therefore we learn that the labeling probability $\eta(\mathbf{x})$ can still be accurately estimated by LBE-LF even though the two classes are not perfectly separable.

Secondly, we consider a more challenging sampling strategy when the assumption of "invariance of order" that is widely employed by existing works [1], [2] is severely violated. To be specific, the sampling strategy is $\eta(\mathbf{x}) = \left[\left(1+\exp(-\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]^\top \mathbf{x})\right)^{-1}\right]^\kappa$ with $\kappa = 10$, and the labeling probability is plotted in Fig. 3(b). We see that under this sampling strategy, the positive data point $\mathbf{x} = (x_1, x_2)^\top$ with large x_2 is more likely to be labeled. Therefore, such $\eta(\mathbf{x})$ is inconsistent with the assumption of "invariance of order", as an example $\mathbf{x}$ with a large $P(y=1|\mathbf{x})$ does not necessarily have a high probability of $P(s=1|y=1,\mathbf{x})$.

- *This work was supported by NSF of China (Nos: 61973162, U1713208, 62006202), the Fundamental Research Funds for the Central Universities (No: 30920032202), CCF-Tencent Open Fund (No: RAGR20200101), the "Young Elite Scientists Sponsorship Program" by CAST (No: 2018QNRC001), Hong Kong Scholars Program (No: XJ2019036), "111 Program" (No: B13022), the RGC Early Career Scheme (No. 22200720), HKBU CSD Departmental Incentive Grant, the Hong Kong Polytechnic University grants (Nos: YZ3K, UAJP/UAGK, and ZVRH), and the ARC (No: DE190101473).*

- *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 Department of Computing, Hong Kong Polytechnic University, Hong Kong, P.R. China.*
  E-mail: chen.gong@njust.edu.cn
- *Q. Wang and B. Han are with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, P.R. China.*
  E-mail: qizhouwang.nanjing@gmail.com; bhanml@comp.hkbu.edu.hk
- *T. Liu is with the Trustworthy Machine Learning Lab, the University of Sydney, 6 Cleveland St, Darlington, NSW 2008, Australia.*
  E-mail: tongliang.liu@sydney.edu.au
- *J. You is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, P.R. China.*
  E-mail: jane.you@polyu.edu.hk
- *J. Yang is with the PCA Lab, the Jiangsu Key Laboratory of Image and Video Understanding for Social Security, and the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, P.R. China.*
  E-mail: csjyang@njust.edu.cn
- *D. Tao is with JD Explore Academy at JD.com, Beijing, P.R. China.*
  E-mail: dacheng.tao@jd.com
- *Corresponding authors: J. Yang and J. You.*

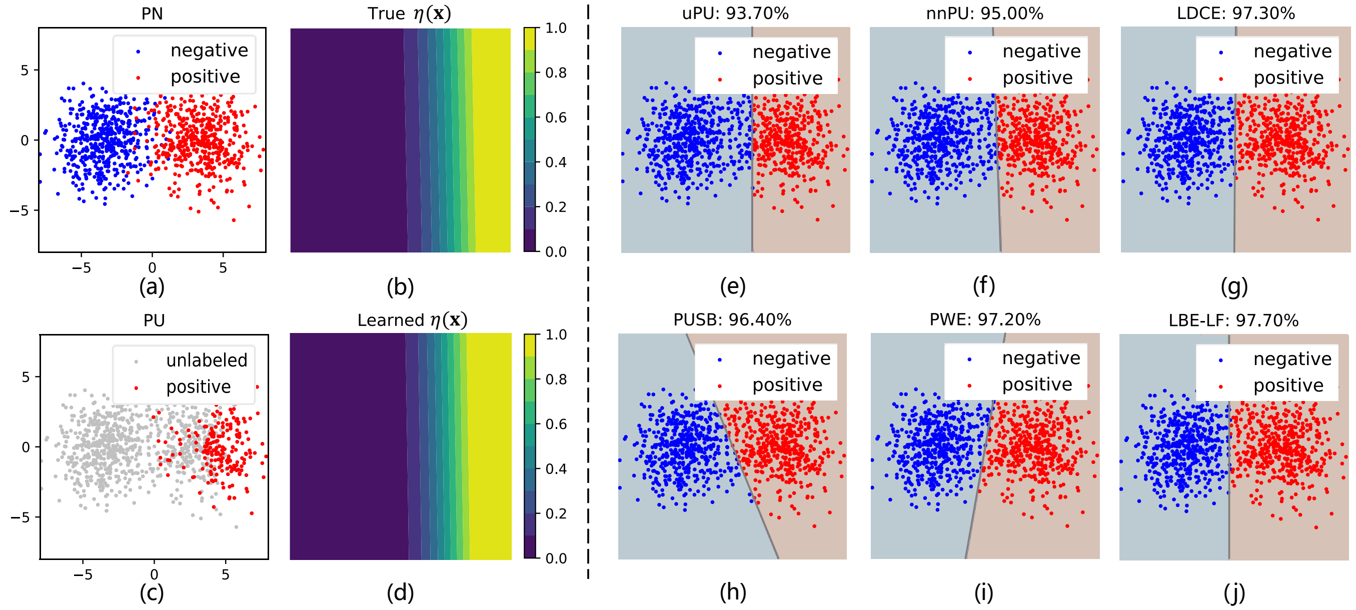

Fig. 1: The performances of various methods on the synthetic dataset under Strategy 1. The inseparable case is particularly studied. (a) shows the real positive and negative data; (c) shows the unlabeled and biased positive data for model training; (b) and (d) present the true $\eta(\mathbf{x})$ and the estimated $\eta(\mathbf{x})$ of LBE; (e)∼(j) display the classification results generated by uPU, nnPU, LDCE, PUSB, PWE, and LBE-LF. The classification accuracy of every method is presented above the corresponding subfigure.

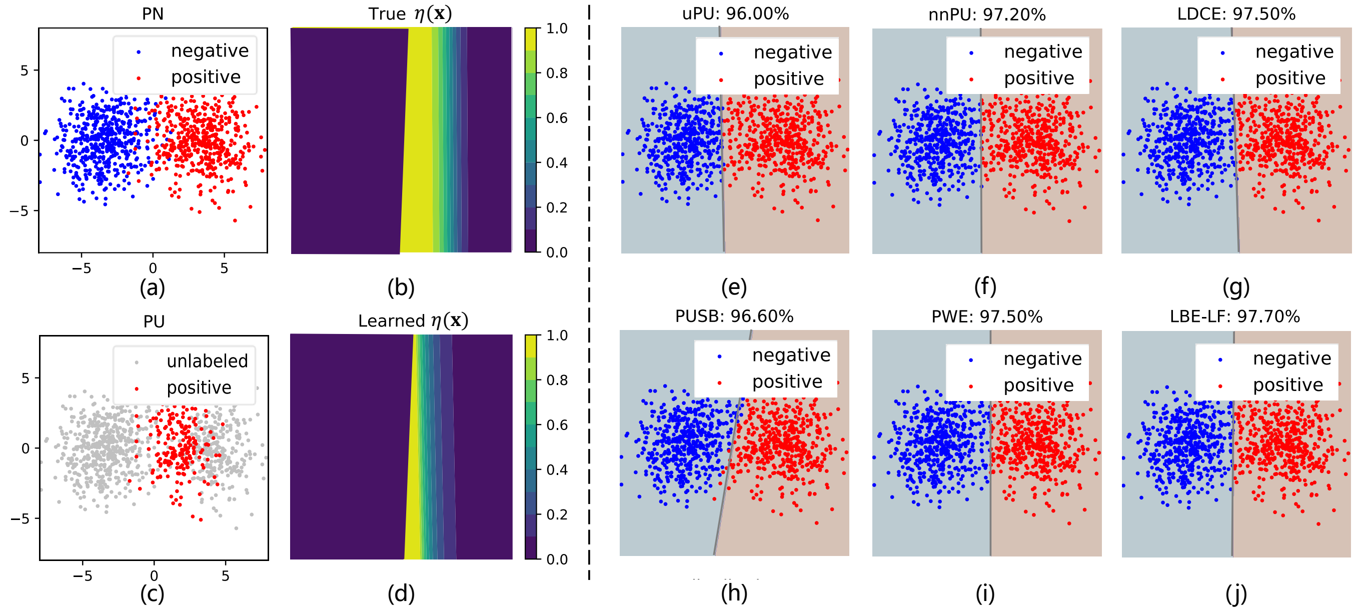

Fig. 2: The performances of various methods on the synthetic dataset under Strategy 2. The inseparable case is particularly studied. (a) shows the real positive and negative data; (c) shows the unlabeled and biased positive data for model training; (b) and (d) present the true $\eta(\mathbf{x})$ and the estimated $\eta(\mathbf{x})$ of LBE; (e)∼(j) display the classification results generated by uPU, nnPU, LDCE, PUSB, PWE, and LBE-LF. The classification accuracy of every method is presented above the corresponding subfigure.

The experimental results generated by various compared methods are presented in Figs. 3(e)∼(j). We see that the decision boundaries yielded by the compared methods are largely influenced by the biasedly sampled positive data. Especially, PUSB, which is based on the assumption of "invariance of order", shows imperfect decision boundary and thus the obtained classification accuracy is only 94.20%. Comparatively, our LBE-LF still yields reasonable decision boundary and the classification accuracy is as high as 99.50%. The reason of our method in achieving impressive classification result is that our method does not need the assumption of "invariance of order", so it can still accurately estimate $\eta(\mathbf{x})$ (see Fig. 3(d)), which is helpful for generating a good binary classifier.

2 Experiments under Constant η

Although our LBE method is devised for instance-dependent PU learning, namely the labeling probability $\eta(\mathbf{x})$ varies across different data points, we show that it is still applicable to instance-independent PU learning where $\eta(\mathbf{x}) = \eta$ is a constant for all $\mathbf{x}$. To be

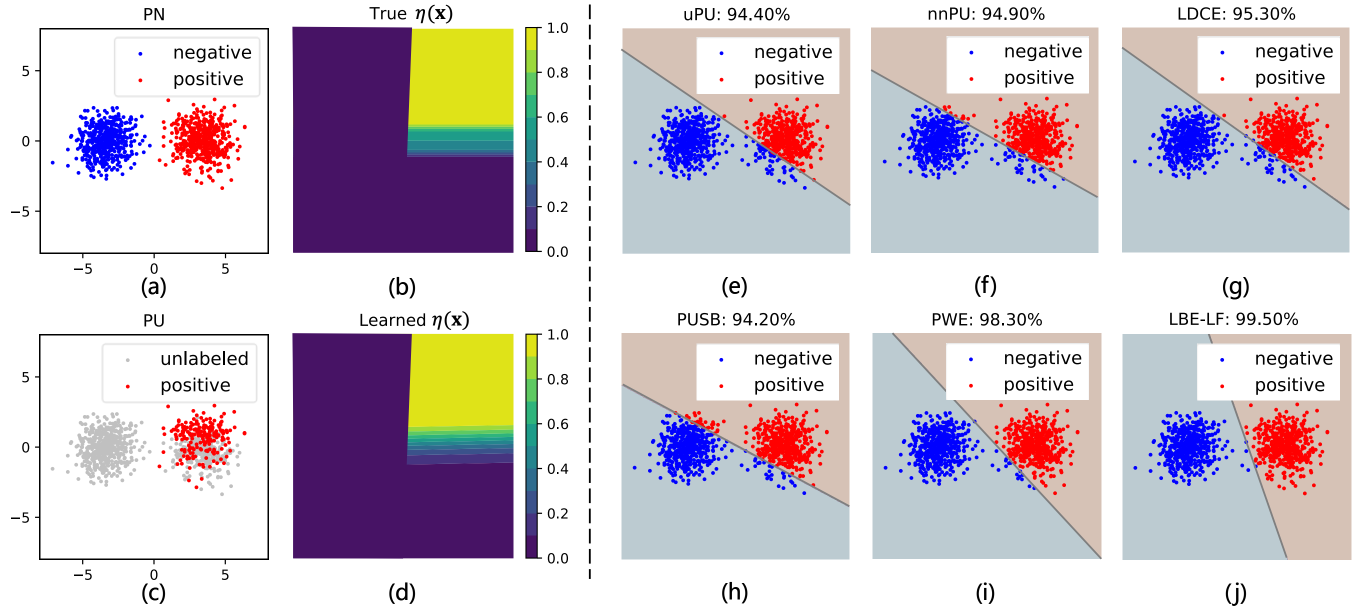

Fig. 3: The performances of various methods on synthetic dataset when the sampling strategy violates the assumption of "invariance of order". (a) shows the real positive and negative data; (c) shows the unlabeled and biased positive data for model training; (b) and (d) present the true $\eta(\mathbf{x})$ and the estimated $\eta(\mathbf{x})$ of LBE; (e)∼(j) display the classification results generated by uPU, nnPU, LDCE, PUSB, PWE, and LBE-LF. The classification accuracy of every method is presented above the corresponding subfigure.

TABLE 1: Experiments on two UCI datasets under the random sampling strategy with different η. The mean test accuracies and the estimated η (mean±std) for our LBE-LF and LBE-MLP are reported.

Dataset	true η	LBE-LF		LBE-MLP	
		accuracy	estimated η	accuracy	estimated η
banknote	0.8	0.9884 ± 0.0050	0.8254 ± 0.0019	0.9993 ± 0.0016	0.8125 ± 0.0049
	0.7	0.9727 ± 0.0015	0.7429 ± 0.0087	0.9793 ± 0.0020	0.6879 ± 0.0024
	0.6	0.9796 ± 0.0041	0.6751 ± 0.0018	0.9723 ± 0.0027	0.5922 ± 0.0011
breast	0.8	0.9704 ± 0.0031	0.8099 ± 0.0081	0.9606 ± 0.0083	0.7959 ± 0.0044
	0.7	0.9674 ± 0.0089	0.6925 ± 0.0087	0.9593 ± 0.0090	0.7012 ± 0.0062
	0.6	0.9731 ± 0.0027	0.6354 ± 0.0025	0.9801 ± 0.0003	0.5786 ± 0.0067

specific, we slightly change Eq. (9) in the main paper to $\eta = P(s = 1|y = 1) = (1 + \exp(-\theta_2))^{-1}$ such that η is irrelevant to $\mathbf{x}$ for both LBE-LF and LBE-MLP. Then we run our algorithm on two UCI benchmark datasets *banknote* and *breast* to see: 1) whether our LBE can still precisely estimate the constant value of η; and 2) the average classification accuracy over five-fold cross validation. The experimental setting is identical to that in the main paper, and on each dataset we focus on the performances of LBE-LF and LBE-MLP when true η are $\{0.6.0.7, 0.8\}$.

The experimental results are displayed in Table 1, from which we observe that the estimated η by our LBE is very close to the true η on the two datasets. As a sequel, the test accuracies are quite high which are all above 95%. Therefore, we see that our LBE is also effective on instance-independent PU learning with constant labeling probability η.

References

[1] M. Kato, T. Teshima, and J. Honda, "Learning from positive and unlabeled data with a selection bias," in *International Conference on Learning Representation*, 2019, pp. 1–12.

[2] F. He, T. Liu, G. I. Webb, and D. Tao, "Instance-dependent pu learning by bayesian optimal relabeling," *arXiv preprint arXiv:1808.02180*, 2018.