Loss Decomposition and Centroid Estimation for Positive and Unlabeled Learning

Chen Gong, *Member, IEEE,* Hong Shi, Tongliang Liu, *Member, IEEE,*
Chuang Zhang, Jian Yang, *Member, IEEE,* Dacheng Tao, *Fellow, IEEE*

Abstract—This paper studies Positive and Unlabeled learning (PU learning), of which the target is to build a binary classifier where only positive data and unlabeled data are available for classifier training. To deal with the absence of negative training data, we first regard all unlabeled data as negative examples with false negative labels, and then convert PU learning into the risk minimization problem in the presence of such one-side label noise. Specifically, we propose a novel PU learning algorithm dubbed "Loss Decomposition and Centroid Estimation" (LDCE). By decomposing the loss function of corrupted negative examples into two parts, we show that only the second part is affected by the noisy labels. Thereby, we may estimate the centroid of corrupted negative set via an unbiased way to reduce the adverse impact of such label noise. Furthermore, we propose the "Kernelized LDCE" (KLDCE) by introducing the kernel trick, and show that KLDCE can be easily solved by combining Alternative Convex Search (ACS) and Sequential Minimal Optimization (SMO). Theoretically, we derive the generalization error bound which suggests that the generalization risk of our model converges to the empirical risk with the order of $\mathcal{O}(1/\sqrt{k}+1/\sqrt{n-k}+1/\sqrt{n})$ (n and k are the amounts of training data and positive data correspondingly). Experimentally, we conduct intensive experiments on synthetic dataset, UCI benchmark datasets and real-world datasets, and the results demonstrate that our approaches (LDCE and KLDCE) achieve the top-level performance when compared with both classic and state-of-the-art PU learning methods.

Index Terms—PU Learning, Loss Decomposition, Centroid Estimation, Kernel Extension, Generalization Bound.

1 Introduction

Different from traditional supervised learning that usually trains a classifier by harnessing both positive and negative examples, Positive and Unlabeled learning (PU learning) aims to find a suitable classifier simply based on positive and unlabeled data. Here each of the unlabeled data can be positive or negative, however its groundtruth label remains unknown during the training stage. An illustrative comparison of PU learning and traditional supervised learning is shown in Figure 1, from which we can see that PU learning is more challenging than the traditional binary supervised learning as the negative examples for training are not explicitly provided. Actually, the setting of PU learning is prevalent in many practical situations, where directly collecting negative examples is difficult or the collected negative examples are too diverse. Here we provide some examples:

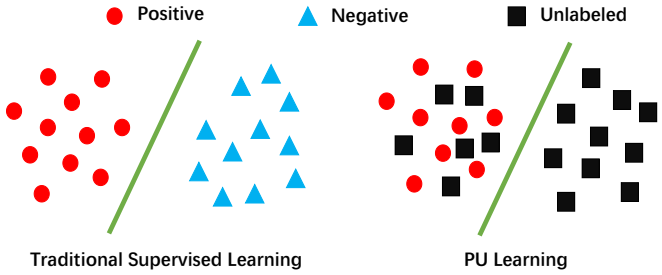

Fig. 1: The comparison of traditional supervised learning and PU learning. Traditional supervised learning trains a classifier from both positive and negative examples, while PU learning trains a classifier simply based on the positive and unlabeled data.

- Product recommendation [1]. The products once bought or favored by a certain customer are positive examples, and the remaining products that have not been browsed by the customer are deemed as unlabeled as some of them may be of interest to this customer. Therefore, PU learning can be adopted to identify the potential positive examples and then recommend them to the customer.
- Outlier detection [2]. The very few outliers identified by the primitive detector constitute the positive set, and the remaining data points are deemed as unlabeled because some

- *This work was supported by NSF of China (No: 61602246, 61973162, U1713208), NSF of Jiangsu Province (No: BK20171430), the Fundamental Research Funds for the Central Universities (No: 30918011319), the open project of State Key Laboratory of Integrated Services Networks (Xidian University, ID: ISN19-03), the "Summit of the Six Top Talents" Program (No: DZXX-027), the "Young Elite Scientists Sponsorship Program" by Jiangsu Province, the "Young Elite Scientists Sponsorship Program" by CAST (No: 2018QNRC001), the Program for Changjiang Scholars, the "111" Program AH92005, the ARC FL-170100117, the DP180103424 and DE190101473.*
- *C. Gong is with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, Nanjing, 210094, P.R. China, and the State Key Laboratory of Integrated Services Networks (Xidian University), Xi'an, 710071, P.R. China.*
  E-mail: {chen.gong}@njust.edu.cn
- *H. Shi, C. Zhang, and J. Yang are with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, 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: {shihong, c.zhang, csjyang}@njust.edu.cn
- *T. Liu and D. Tao are with the UBTECH Sydney Artificial Intelligence Centre and the School of Computer Science, in the Faculty of Engineering, at the University of Sydney, 6 Cleveland St, Darlington, NSW 2008, Australia.*
  E-mail: {tongliang.liu, dacheng.tao}@sydney.edu.au

outliers are probably hidden among them. Consequently, PU learning is an ideal tool to further detect the outliers in the unlabeled set.
- Remotely-sensed hyperspectral image classification [3]. We may treat the specific land-cover category that is studied as positive while the rest of the diverse land types are taken as unlabeled. By training a PU classifier, we can then effectively find the studied land type from a new hyperspectral image.
- Medical diagnosis [4]. Due to the scarcity of sick patients (*e.g.*, Alzheimer's disease, Parkinson's disease, etc.), we can only get very limited cases of illness. By treating them as positive and the unknown cases as unlabeled, one can employ PU learning to establish a binary classifier that is able to distinguish the patients and normal people.

To cope with the intensive practical demands as listed above, PU learning is proposed to handle the situations where only positive data and unlabeled data are available. PU learning has attracted a great deal of research attention in recent years, and a considerable amount of work has been done. The existing PU methods can be roughly divided into three categories. The first category focuses on extracting reliable negative examples from the unlabeled set, and then using these confident negative examples and the original positive examples to train a classifier, such as [5], [6]. The main shortcoming is that the extraction of the examples with reliable labels is not always accurate, which directly influences the performance of the finally obtained classifier. The second category formulates PU learning as a cost-sensitive learning problem by imposing different weights on positive data and unlabeled data. The representative works are [7], [8], [9], [10]. The methods belonging to this category usually employ the re-weighting technique to calibrate the inaccurate data distribution, which is caused by the missing of negative examples, to the potentially correct one. Nevertheless, the weights are usually determined based on the positive class prior, which could be rather empirical and is very likely to lead to the degraded classification results. The last category treats the unlabeled data as negative ones, and thus transferring PU learning problem into a one-side label noise learning problem [11], [12]. That is to say, some of the originally positive examples in the unlabeled set are regarded as mislabeled, while no negative examples are incorrectly labeled as positive. The typical methods include [13], [14]. However, these methods would fail when the unlabeled set contains a large amount of originally positive examples.

To overcome the drawbacks of the existing methods mentioned above and also inspired by the recent advances of label noise learning techniques [15], [16], [17], in this paper we follow the idea of the third category and formulate PU learning as a noisy label learning problem by regarding all unlabeled data as negative examples with label noise. Particularly, we propose an unbiased estimate of the true risk on PU datasets by utilizing the manipulations of loss decomposition [18] and centroid estimation [15]. As a result, we can explicitly model the label noise in negative set (*i.e.*, the original unlabeled set) and formulate PU learning as a risk minimization problem in the presence of false negative labels. To be specific, by decomposing the empirical loss (*e.g.*, hinge loss) on the corrupted negative set into two terms, we observe that only the second term is label-dependent and is affected by the noisy labels. Therefore, inspired by [15], the risk minimization in the presence of label noise can be converted to the estimation of the centroid of the labeled examples, and thus the adverse impact of noise can be eliminated. By the way, our problem is turned into the estimation of the centroid of noisy negative set. Consequently, we can obtain the unbiased risk of true PU dataset and propose a novel PU learning algorithm dubbed "Loss Decomposition and Centroid Estimation" (LDCE). To address non-linear cases, we further extend the linear LDCE model to a Kernelized LDCE (KLDCE) by introducing the kernel trick, which can be efficiently solved by combining the methods of Alternative Convex Search (ACS) and Sequential Minimal Optimization (SMO) [19]. Moreover, we theoretically analyze the generalization ability of our algorithm, showing that the empirical classification risk of an arbitrary classifier learned by our proposed algorithm will converge to its expected classification when the number of training examples goes to infinity. Intensive experimental results on synthetic and practical datasets demonstrate that the proposed algorithms are superior to the existing state-of-the-art PU learning methods.

This work is the extension of our previous conference paper [20]. Compared with [20], this paper has been improved with respect to the following aspects:

1) A kernelized model termed KLDCE is proposed to enable our method to tackle the non-linear cases.
2) An ACS and SMO based optimization algorithm is derived to efficiently solve the KLDCE model.
3) The generalization bound of our proposed algorithm is theoretically proved based on the Rademacher complexity analysis.
4) More experimental results are provided, and the algorithm behavior of our model is also empirically studied.

The outline of the paper is as follows. In Section 2, we revisit the existing typical PU methods. The problem setting is introduced in Section 3. In Section 4, we explain the loss decomposition technique, which helps to further analyze the effect of label noise in Section 5. The basic LDCE model as well as its extension to non-linear case is presented in Section 6. Section 7 proves the generalization bound of our model. The experimental results on different datasets and the model performance analyses are given in Section 8. Finally, Section 9 concludes the entire paper.

2 Related Work

As an important branch of weakly supervised learning [21], [22], the preliminary researches of PU Learning can be dated back to [23], [24], [25], which reveals that unlabeled data show a great impact on accurately training a binary classifier without the aid of negative examples.

Initially, Denis *et al.* [23] concentrate on the learning complexity and show that the function classes learnable under the statistical query model are also learnable from positive and unlabeled data. After that, a series of PU learning [26], [27] methods have been developed, which usually follow two popular settings according to how the positive and unlabeled data are generated. Specifically, let $\mathbf{X} \in \mathbb{R}^d$ (d is the dimensionality) be the input variable in the feature space $\mathcal{X}$ and $Y \in \mathbb{R}$ be the output variable in the label space $\mathcal{Y} = \{1, -1\}$, then the class-conditional density on positive data and the marginal density of $\mathbf{X}$ are respectively formulated as $P(\mathbf{X}|Y = 1)$ and $P(\mathbf{X})$, where $P(\cdot)$ denotes the probability throughout this paper. The first setting is called "case-control PU learning" [28] which follows a two-sample configuration, namely the input data $\mathbf{x}$ in positive set S_P and unlabeled set S_U are independently drawn as $S_P = \{\mathbf{x}_i\}_{i=1}^k \overset{i.i.d.}{\sim} P(\mathbf{X}|Y = 1)$ and

$S_U = \{\mathbf{x}_i\}_{i=k+1}^{n} \overset{i.i.d.}{\sim} P(\mathbf{X})$ with n and k denoting the sizes of $S = S_P \cup S_U$ and S_P accordingly. The second setting is called "censoring PU learning" [7], where only one sample $S = \{\mathbf{x}_i\}_{i=1}^{n}$ with size n is 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 p' (*i.e.*, $\mathbf{x}_i \in S_P$), and remains undiscovered with probability $1-p'$ (*i.e.*, $\mathbf{x}_i \in S_U$). If the groundtruth label of $\mathbf{x}_i$ is -1, its label will never be observed and it belongs to S_U with probability 1.

Under the case-control PU learning setting, Liu *et al.* [5] send "spy" examples from the positive set to the unlabeled set so that the conditional probability of an example belonging to positive class can be estimated. Then the probabilistic labels of the "spies" are used to detect the examples that are probably negative from the unlabeled set. Finally, the asymmetric cost formulation of the SVM algorithm with two trade-off parameters C_+ and C_- is deployed to respectively weight the positive errors and negative errors during training [14]. However, such detection of probable negative examples can be inaccurate. To address this defect, several recent works try to design various unbiased or biased risk estimators, which yield the top-level performance. Specifically, for reducing the data bias caused by the absence of negative examples, du Plessis *et al.* [8] develop a non-convex ramp loss, with which the learning under the PU setting would give the same classification boundary as in the ordinary supervised setting. To overcome the challenge brought by the non-convexity, a convex unbiased loss is introduced in [9], which uses a weighted ordinary convex loss function for unlabeled data and a weighted composite convex loss function for positive data. Considering that the empirical risk designed in [9] can be negative which causes the over-fitting problem, a biased non-negative risk estimator [29] is further proposed. Moreover, the recent work [10] studies a situation when a small number of biased negative examples can also be collected in addition to the positive and unlabeled data.

Under the censoring PU learning setting, many approaches cast PU learning as the learning problem with one-side label noise by treating the unlabeled data as noisy negative examples. That is to say, the label noises in the whole dataset are only caused by the false negative examples. For example, weighted Logistic regression [13] performs the standard logistic regression after weighting the examples to handle the noise of which the rate is greater than a half. Biased-PrTFIDF [30] employs a probabilistic learning algorithm PrTFIDF [31] and imposes an estimable weight on the conditional probability of an example to be positive to reduce the disturbance of label noise. Weighted SVM [7] assigns different weights to the class-specific losses via parameter tuning. Different from above methods in which the positive priors of all examples are identical, He *et al.* [32] claim that the prior of an example being positive is related to the data distribution, and favor to build an optimal Bayesian classifier to tackle this problem.

There are also some setting-free PU methods developed so far, such as [33], [34] based on cluster assumption, [35] based on multi-manifold assumption, and [36] based on label disambiguation.

In this paper, we follow the setting of censoring PU learning as this setting allows us to directly estimate the class prior $P(Y=1)$ which is crucial in our model. Besides, our idea that treats fractional unlabeled data as false negative also connects PU learning to class-conditional label noise learning as suggested in [16], [37]. Specifically, we algorithmically show that PU learning is a special case of class-conditional label noise learning when the label flipping probability from negative to positive is zero.

3 Problem Description

According to the setting of censoring PU learning explained in Section 2, we may suppose that a total of n training data $S = \{S_P; S_U\} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_k, y_k); (\mathbf{x}_{k+1}, y_{k+1}), \ldots, (\mathbf{x}_n, y_n)\}$ are identically and independently drawn from some distribution $\mathcal{D}$ defined on $\mathcal{X} \times \mathcal{Y}$, where the labels of the first k examples in S (*i.e.*, S_P) are observed as positive, and the labels of the rest $n-k$ data (*i.e.*, S_U) are unknown. In the training stage, our target is to obtain a suitable decision function $h\colon \mathcal{X} \to \mathbb{R}$ on S, such that the unobserved test example $\mathbf{x}$ can obtain the correct label $sgn(h(\mathbf{x}))$ assigned by h.

Similar to [13], this paper also formulates the censoring PU learning as a learning problem with one-side label noise where all unlabeled data are treated as negative with label noise while the positive examples in S_P have clean labels without any noise. Therefore, a noisy version of the original training sample S is obtained as $\widetilde{S} = \{S_P; \widetilde{S_N}\} = \{(\mathbf{x}_1, \widetilde{y}_1), \ldots, (\mathbf{x}_k, \widetilde{y}_k); (\mathbf{x}_{k+1}, \widetilde{y}_{k+1}), \ldots, (\mathbf{x}_n, \widetilde{y}_n)\}$ where the corrupted negative set $\widetilde{S_N}$ takes the same place as the original unlabeled set S_U. Obviously, $\{\widetilde{y}_i\}_{i=k+1}^{n} = -1$ for $(\mathbf{x}_i, \widetilde{y}_i) \in \widetilde{S_N}$, and $\widetilde{y}_i = y_i = 1$ for $(\mathbf{x}_i, y_i) \in S_P$ since S_P contains no false positive examples. Consequently, the noisy set $\widetilde{S}$ is employed for our model training. Additionally, let $\widetilde{Y}$ be the corrupted version of Y, and defining $\eta = P(\mathbf{X} \in S_U | Y=1)$, we get the fact that

$$P(\widetilde{Y}=-1|Y=1)=\eta, \;\; P(\widetilde{Y}=1|Y=-1)=0, \quad (1)$$

where η in our model is interpreted as the flipping probability of a positive example to be negative in $\widetilde{S_N}$ that can be tuned via cross-validation [16]. Alternatively, it can be estimated by some advanced methods such as [38], [39], [40] based on the theory of mixture proposition estimation, [41] based on the divergence penalization via ℓ_1-regularization, [42] based on the density estimation in a univariate space, or [43] based on the decision tree induction. Besides, we have

$$\begin{aligned} &P(\widetilde{Y}=1) \\ &=P(\widetilde{Y}=1|Y=1)P(Y=1)+P(\widetilde{Y}=1|Y=-1)P(Y=-1) \\ &=P(\widetilde{Y}=1|Y=1)P(Y=1)+0\times P(Y=-1) \\ &=P(\widetilde{Y}=1|Y=1)P(Y=1), \end{aligned} \quad (2)$$

where $P(\widetilde{Y}=1)$ is the prior of observed positive examples approximated by k/n. Consequently, the class prior probability of true positive examples in PU datasets defined as $p = P(Y=1)$ can be expressed as

$$p = P(Y=1) = \frac{P(\widetilde{Y}=1)}{P(\widetilde{Y}=1|Y=1)} = \frac{P(\widetilde{Y}=1)}{1-\eta}. \quad (3)$$

According to the above formulation, we can compute the positive class prior p directly.

4 Loss Decomposition

In our paper, we formulate PU learning as the risk minimization problem of learning from pure positive and corrupted negative examples. Suppose that we have a classifier h and the loss function is $\ell\colon \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ which evaluates the deviation between the

predicted value $h(\mathbf{x})$ and the groundtruth label y. In our case, the empirical risk of classifier h on S, *i.e.*, $\hat{\mathcal{R}}(h,S)$, is composed of two parts: the loss on the clean positive examples and the loss on the unlabeled examples, namely

$$\begin{aligned}\hat{\mathcal{R}}(h,S) &= \frac{1}{n}[\sum_{i=1}^{k}\ell(y_i,h(\mathbf{x}_i)) + \sum_{i=k+1}^{n}\ell(y_i,h(\mathbf{x}_i))] \\ &= \hat{\mathcal{R}}_P(h,S_P) + \hat{\mathcal{R}}_U(h,S_U).\end{aligned} \tag{4}$$

Due to the correctness of all the labels in S_P, the first term $\hat{\mathcal{R}}_P(h,S_P)$ in Eq. (4) can be easily computed. However, it is infeasible to get the real value of the second term $\hat{\mathcal{R}}_U(h,S_U)$ owing to the unknown groundtruth labels of the examples in S_U. Therefore, an important thing here is to study how to obtain the unbiased estimate of the second term.

This paper employs hinge loss $\ell(z)=[1-z]_+$ where $[\cdot]_+ = \max(\cdot,0)$ and $z=yh(\mathbf{x})$ is the functional margin, then according to [18], we can further decompose the hinge loss $\ell(z)$ on the negative examples into two parts, which yields

$$\begin{aligned}\ell(z) &= [1-z]_+ \\ &\overset{1}{=} \frac{1}{2}([1-z]_+ + [1+z]_+) + \frac{1}{2}([1-z]_+ - [1+z]_+) \\ &\overset{2}{=} \frac{1}{2}([1-z]_+ + [1+z]_+) + \frac{1}{4}(-2z + |1-z| - |1+z|),\end{aligned} \tag{5}$$

where the 2^{nd} equation holds due to $[a]_+ = a/2 + |a|/2$. Since for any z, there are $|1-z| \leqslant |z|+1$ and $|1+z| \geqslant |z|-1$, Eq. (5) can be further derived as

$$\ell(z) \leqslant \frac{1}{2}([1-z]_+ + [1+z]_+) + \frac{1}{2}(1-z). \tag{6}$$

In Eq. (6), the term in the first bracket of the right-hand side can eliminate the adverse impact of noise due to the property of even function. Here only the second term $\frac{1}{2}(1-z)$ is affected by label noise, which will be further studied in Section 5.

Remark 1: The prior works [9] and [18] have proven that a loss function $\ell(z)$ will lead to a convex PU learning model if it satisfies the linear-odd property $\ell(z)-\ell(-z)=-z$, so the traditional hinge loss is not preferred as it does not have such linear-odd property. In contrast, Eq. (6) shows that the conventional hinge loss, which is often utilized to achieve the max-margin effect, can also be deployed for convex PU learning as long as we minimize its upper bound which satisfies the linear-odd property. Other typical loss functions such as squared loss, logistic loss and double hinge loss mentioned in [9] can also be adopted here. However, considering the popularity for classification and simplicity for optimization, we still employ the hinge loss for our model which already achieves satisfactory performances as will be revealed in the experimental section.

Remark 2: It can be easily proved that the upper bound shown in Eq. (6) is tight regarding the original hinge loss $\ell(z)$. The margins between the right-hand side and left-hand side of Eq. (6) are 1, $\frac{1}{2}(z+1)$ and 0 when $z>1$, $-1\leq z\leq 1$ and $z<-1$, respectively, which suggest that the difference between the two sides of Eq. (6) is at most 1.

According to Eq. (6), the upper bound of $\hat{\mathcal{R}}_U(h,S_U)$ is formulated as

$$\begin{aligned}\overline{\mathcal{R}}_U(h,S_U) =& \frac{1}{n}\sum_{i=k+1}^{n}\frac{1}{2}([1-y_ih(\mathbf{x}_i)]_+ + [1+y_ih(\mathbf{x}_i)]_+) \\ &+ \frac{1}{2}(1-y_ih(\mathbf{x}_i)).\end{aligned} \tag{7}$$

Therefore, due to the tightness of this bound as explained in Remark 2 and that $\hat{\mathcal{R}}_U(h,S_U)$ is lower bounded by 0, we may replace the original $\hat{\mathcal{R}}_U(h,S_U)$ by its upper bound $\overline{\mathcal{R}}_U(h,S_U)$ in the empirical risk minimization framework. For notational simplicity, we slightly abuse the notation and directly use $\hat{\mathcal{R}}_U(h,S_U)$ to represent $\overline{\mathcal{R}}_U(h,S_U)$ in the following sections.

5 Analysis of Noisy Negative Examples

In this section, we give detailed analysis to the classification risk on the corrupted negative examples in $\widetilde{S_N}$ (*i.e.* the original S_U). Suppose that we have a linear classifier as $h_{\boldsymbol{w}}(\mathbf{x}_i)=\langle \boldsymbol{w},\mathbf{x}_i\rangle$ where $\boldsymbol{w}$ is the model parameter, then based on Eq. (7), we know that

$$\begin{aligned}\hat{\mathcal{R}}_U(h,S_U) =& \frac{1}{n}\sum_{i=k+1}^{n}\frac{1}{2}([1-y_ih(\mathbf{x}_i)]_+ + [1+y_ih(\mathbf{x}_i)]_+) \\ &+ \frac{1}{2}(1-y_ih(\mathbf{x}_i)) \\ =& \frac{1}{n}\sum_{i=k+1}^{n}\frac{1}{2}([1-y_ih(\mathbf{x}_i)]_+ + [1+y_ih(\mathbf{x}_i)]_+) \\ &+ \frac{n-k}{2n} - \frac{1}{2}\langle \boldsymbol{w}, \frac{1}{n}\sum_{i=k+1}^{n}y_i\mathbf{x}_i\rangle. \\ =& \frac{1}{2n}\sum_{i=k+1}^{n}([1-y_i\langle \boldsymbol{w},\mathbf{x}_i\rangle]_+ + [1+y_i\langle \boldsymbol{w},\mathbf{x}_i\rangle]_+) \\ &- \frac{n-k}{2n}\langle \boldsymbol{w}, \mu(S_U)\rangle + \text{Const}\end{aligned} \tag{8}$$

where "Const" is short for "Constant", and $\mu(S_U) = \frac{1}{n-k}\sum_{i=k+1}^{n}\mathbf{m}_i = \frac{1}{n-k}\sum_{i=k+1}^{n}y_i\mathbf{x}_i$ with $\mathbf{m}_i = y_i\mathbf{x}_i$ ($i=1,2,\cdots,n$) being a sample of the random variable $\mathbf{M}=Y\mathbf{X}$ drawn from the true distribution $\mathcal{D}$. Note that $\mu(S_U)$ here is statistically meaningful and is defined as the centroid of true unlabeled set which is related to the examples in S_U with true labels y_i and true distribution $\mathcal{D}$, and we can write $\mu(\mathcal{D}) = \mathbb{E}_{(\mathbf{X},Y)\sim\mathcal{D}}[\mathbf{X},Y]$.

According to the analysis of Eq. (6) in Section 4, we know that the first summation term of Eq. (8) is label independent, and only the second term containing $\mu(S_U)$ should be investigated to deal with label noise as it is related to the unknown true labels y_i. Therefore, by denoting the variable $\widetilde{\mathbf{M}}=\widetilde{Y}\mathbf{X}$ and its sample $\widetilde{\mathbf{m}}_i=\widetilde{y}_i\mathbf{x}_i$ ($i=1,2,\cdots,n$), we also define the centroid of corrupted negative set $\mu(\widetilde{S_N}) = \frac{1}{n-k}\sum_{i=k+1}^{n}\widetilde{\mathbf{m}}_i = \frac{1}{n-k}\sum_{i=k+1}^{n}\widetilde{y}_i\mathbf{x}_i$ and $\mu(\mathcal{D}_\eta) = \mathbb{E}_{(\mathbf{X},\widetilde{Y})\sim\mathcal{D}_\eta}[\mathbf{X},\widetilde{Y}]$ on the corrupted negative set $\widetilde{S_N}$ and corrupted distribution $\mathcal{D}_\eta$, respectively. Consequently, we have the following theorem which bridges the centroid of true unlabeled set $\mu(S_U)$ and that of corrupted negative set $\mu(\widetilde{S_N})$, namely:

Theorem 1. *Given η as the label flipping probability and p as the prior probability of true positive example defined in Eq.* (3)*, the relationship of means between the true distribution $\mathcal{D}$ and the corrupted distribution $\mathcal{D}_\eta$ can be expressed as $\mu(\mathcal{D}_\eta) = (1-2p\eta)\mu(\mathcal{D})$. Similarly, the example centroids of the true unlabeled set S_U and the corrupted negative set $\widetilde{S_N}$ have the relationship $\mathbb{E}_{\widetilde{y}_1,\ldots,\widetilde{y}_n}[\mu(\widetilde{S_N})] = (1-2p\eta)\mu(S_U)$.*

This theorem is proved in the **appendix**. Note that we can only observe the corrupted negative set $\widetilde{S_N}$ on the corrupted distribution $\mathcal{D}_\eta$ rather than the true corresponding set S_U on

true distribution $\mathcal{D}$, so it is fortunate to have Theorem 1 which reveals that $\mu(\widetilde{S_N})/(1-2p\eta)$ is an unbiased estimate of $\mu(S_U)$. However, the direct estimate of $\mu(S_U)$ via $\mu(\widetilde{S_N})$ might be inaccurate in our problem as the label noise in $\widetilde{S_N}$ will make the covariance of $\widetilde{\mathbf{M}}$ quite large which may lead to the heavy-tailed distribution. To show this point, we use $\Sigma(\mathcal{D})$ to denote the covariance matrix of the random variable $\mathbf{M}$ on the distribution $\mathcal{D}$, and employ $\Sigma(\mathcal{D}_\eta)$ to represent the covariance matrix of $\widetilde{\mathbf{M}}$ on the distribution $\mathcal{D}_\eta$, then we have the following theorem (proved in the **appendix**):

Theorem 2. *The covariance matrices of* $\mathbf{M}$ *on the true distribution* $\mathcal{D}$ *(i.e.* $\Sigma(\mathcal{D})$*) and* $\widetilde{\mathbf{M}}$ *on the corrupted distribution* $\mathcal{D}_\eta$ *(i.e.* $\Sigma(\mathcal{D}_\eta)$*) have the relationship*

$$\Sigma(\mathcal{D}_\eta) = \Sigma(\mathcal{D}) + 4p\eta(1-p\eta)[\mu(\mathcal{D})]^\top \mu(\mathcal{D}), \tag{9}$$

which suggests that the noisy labels with the nonnegative flipping probability η will make the covariance $\Sigma(\mathcal{D}_\eta)$ of $\widetilde{\mathbf{M}}$ larger than the covariance $\Sigma(\mathcal{D})$ of $\mathbf{M}$, which may lead to the deviated estimation of $\mu(S_U)$ from $\mu(\widetilde{S_N})$. Therefore, we further investigate the covariance matrix $\Sigma(\mu(\widetilde{S_N}))$ for the centroid $\mu(\widetilde{S_N})$ of corrupted negative set, of which the empirical version is given in the following theorem:

Theorem 3. *Given* $\widetilde{S_N}$ *as the corrupted negative set, the empirical covariance matrix* $\hat{\Sigma}(\mu(\widetilde{S_N}))$ *is formulated as*

$$\hat{\Sigma}[\mu(\widetilde{S_N})] = \sum_{i=k+1}^{n} \frac{\mathbf{x}_i^\top \mathbf{x}_i}{(n-k)^2} - \frac{1}{n-k}\sum_{i=k+1}^{n}\frac{\mathbf{x}_i^\top \widetilde{y}_i}{n-k}\sum_{i=k+1}^{n}\frac{\mathbf{x}_i \widetilde{y}_i}{n-k}. \tag{10}$$

Proof. The definition of covariance matrix is

$$\Sigma(\mu(\widetilde{S_N})) = \mathbb{E}[[\mu(\widetilde{S_N})]^\top \mu(\widetilde{S_N})] - [\mathbb{E}[\mu(\widetilde{S_N})]]^\top \mathbb{E}[\mu(\widetilde{S_N})]. \tag{11}$$

Besides, since

$$\begin{aligned}\mathbb{E}[[\mu(\widetilde{S_N})]^\top \mu(\widetilde{S_N})] &= \mathbb{E}[[\frac{1}{n-k}\sum_{i=k+1}^{n}\widetilde{y}_i\mathbf{x}_i]^\top \frac{1}{n-k}\sum_{i=k+1}^{n}\widetilde{y}_i\mathbf{x}_i] \\ &= \frac{1}{(n-k)^2}(\sum_{i=k+1}^{n}\mathbb{E}[\mathbf{x}_i^\top\mathbf{x}_i] + \sum_{i\neq j}\mathbb{E}[\widetilde{y}_i\widetilde{y}_j\mathbf{x}_i^\top\mathbf{x}_j]),\end{aligned} \tag{12}$$

and

$$\mathbb{E}[\widetilde{y}_i\widetilde{y}_j\mathbf{x}_i^\top\mathbf{x}_j] = \mathbb{E}[\sum_{i=k+1}^{n}\frac{\mathbf{x}_i\widetilde{y}_i}{n-k}]^\top\mathbb{E}[\sum_{i=k+1}^{n}\frac{\mathbf{x}_i\widetilde{y}_i}{n-k}], \tag{13}$$

we can easily get the covariance matrix by substituting Eqs. (12) and (13) into Eq. (11).

As demonstrated by [15], the empirical covariance as Eq. (10) is a good approximation of $\Sigma(\mu(\widetilde{S_N}))$, so it can be utilized to constrain the variation of $\mu(\widetilde{S_N})$ as will be shown in the subsequent section.

6 The Proposed Algorithm

In this section, we formally present the proposed algorithm based on the analyses in Sections 4 and 5.

Algorithm 1 Median-of-means estimator of corrupted negative mean

Input: The corrupted negative set $\widetilde{S_N}$; the number of groups g with $g \geq 1$;
Output: The median-of-means estimator $\hat{\mu}(\widetilde{S_N})$;
1: Randomly partition $\widetilde{S_N}$ into g groups $\{\widetilde{S_N}^{[1]}, \widetilde{S_N}^{[2]}, \ldots, \widetilde{S_N}^{[g]}\}$ with almost equal size;
2: Compute the empirical mean $\mu(\widetilde{S_N}^{[i]})$ for each $i \in [g]$;
3: Compute $r_i = \text{median}_j\{\left\|\mu(\widetilde{S_N}^{[i]}) - \mu(\widetilde{S_N}^{[j]})\right\|_2\}$ for each $i \in [g]$, and then set $i_* = \arg\min_{i\in[g]} r_i$;
4: **return** $\hat{\mu}(\widetilde{S_N}) = \hat{\mu}(\widetilde{S_N}^{[i_*]})$.

Algorithm 2 Loss Decomposition and Centroid Estimation (LDCE) algorithm for PU learning

Input: The corrupted sample $\widetilde{S} = \{(\mathbf{x}_1, \widetilde{y}_1), \ldots, (\mathbf{x}_k, \widetilde{y}_k), (\mathbf{x}_{k+1}, \widetilde{y}_{k+1}), \ldots, (\mathbf{x}_n, \widetilde{y}_n)\}$, the parameters η, λ and β;
Output: The optimal classifier parameter $\boldsymbol{w}$;
1: Call Algorithm 1 to give an initial estimation of $\hat{\mu} = \hat{\mu}(\widetilde{S_N})$;
2: Compute $\hat{\Sigma} = \hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))$ by Eq. (10);
3: Initialize $\boldsymbol{w}$ and set $t = 0$;
4: **repeat**
5: Calculate $\mu = \hat{\mu} + \hat{\Sigma}^{-1}\boldsymbol{w}\sqrt{\beta/(\boldsymbol{w}^\top\hat{\Sigma}^{-1}\boldsymbol{w})}$;
6: Use gradient descent method to solve

$$\begin{aligned}\boldsymbol{w}_t = \arg\min_{\boldsymbol{w}} &\frac{1}{n}\sum_{i=1}^{k}\ell(\widetilde{y}_i\langle\boldsymbol{w},\mathbf{x}_i\rangle) + \frac{1}{2n}\sum_{i=k+1}^{n}\varphi(\widetilde{y}_i\langle\boldsymbol{w},\mathbf{x}_i\rangle) \\ &+ \frac{c}{1-2p\eta}\langle\boldsymbol{w},\mu\rangle + \lambda\|\boldsymbol{w}\|^2;\end{aligned}$$

7: $t = t + 1$;
8: **until** convergence;
9: **return** The converged $\boldsymbol{w}$.

6.1 The Basic LDCE Algorithm

From Section 5, we know that in order to get the unbiased estimate of centroid $\mu(S_U)$ of unlabeled set, we need to obtain the centroid $\mu(\widetilde{S_N})$ of the corrupted negative set $\widetilde{S_N}$. Due to the influence of noise on the covariance of $\widetilde{\mathbf{M}}$ discussed in Section 5, in this paper we estimate $\mu(\widetilde{S_N})$ by adopting the generalized median-of-means estimator [44] rather than the traditional empirical version $\mu(\widetilde{S_N}) = \frac{1}{n-k}\sum_{i=k+1}^{n}\widetilde{y}_i\mathbf{x}_i$. The basic idea is that the corrupted negative set $\widetilde{S_N}$ is randomly broken up into g groups with almost equal size, and then return the generalized median of example means for each group under ℓ_2-norm metric. In Algorithm 1, we present the description of this process in detail.

To further reduce the adverse influence of label noise on computing $\mu(\widetilde{S_N})$, here we follow [15] and impose a constraint on $\mu(\widetilde{S_N})$ (μ for short in the following explanations), which is

$$(\mu - \hat{\mu}(\widetilde{S_N}))^\top\hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))(\mu - \hat{\mu}(\widetilde{S_N})) \leq \beta, \tag{14}$$

where $\hat{\mu}(\widetilde{S_N})$ is the output of Algorithm 1, $\hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))$ is computed by Eq. (10), and β can be tuned by cross-validation. Note that in our method, we do not directly take $\hat{\mu}(\widetilde{S_N})$ as the final estimate for μ, but regard μ as an unknown variable to be optimized. This operation has two main advantages: 1) the obtained optimal μ (*i.e.* $\mu^\star$) is allowed to vary around the preliminary estimation $\hat{\mu}(\widetilde{S_N})$ slightly within a range suggested by the constraint (14), so the unstable result caused by the large covariance of $\widetilde{\mathbf{M}}$ can be

suppressed; and 2) starting from the imprecise point $\hat{\mu}(\widetilde{S_N})$, the optimal $\mu^\star$ can be automatically learned during the optimization process. Consequently, our PU learning model is formalized as

$$\begin{aligned}\min_{\boldsymbol{w},\mu} \quad & \frac{1}{n}\sum_{i=1}^{k}\ell(y_i\langle \boldsymbol{w},\mathbf{x}_i\rangle)+\frac{1}{2n}\sum_{i=k+1}^{n}\varphi(y_i\langle \boldsymbol{w},\mathbf{x}_i\rangle)\\ & +\frac{c}{1-2p\eta}\langle \boldsymbol{w},\mu\rangle+\lambda\|\boldsymbol{w}\|^2,\\ \text{s.t.} \quad & (\mu-\hat{\mu}(\widetilde{S_N}))^\top\hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))(\mu-\hat{\mu}(\widetilde{S_N}))\le\beta,\end{aligned} \tag{15}$$

where $\ell(y_i\langle \boldsymbol{w},\mathbf{x}_i\rangle)=[1-y_i\langle \boldsymbol{w},\mathbf{x}_i\rangle]_+$, $c=-(n-k)/2n$, $\varphi(y_i\langle \boldsymbol{w},\mathbf{x}_i\rangle)=[1-y_i\langle \boldsymbol{w},\mathbf{x}_i\rangle]_+ + [1+y_i\langle \boldsymbol{w},\mathbf{x}_i\rangle]_+$, $p=k/n(1-\eta)$, and the regularization term $\lambda\|\boldsymbol{w}\|^2$ is employed to avoid overfitting. Due to that the labels in positive set are clean and $\varphi(\cdot)$ is an even function, Eq. (15) can be rewritten as

$$\begin{aligned}\min_{\boldsymbol{w},\mu} \quad & \frac{1}{n}\sum_{i=1}^{k}\ell(\widetilde{y}_i\langle \boldsymbol{w},\mathbf{x}_i\rangle)+\frac{1}{2n}\sum_{i=k+1}^{n}\varphi(\widetilde{y}_i\langle \boldsymbol{w},\mathbf{x}_i\rangle)\\ & +\frac{c}{1-2p\eta}\langle \boldsymbol{w},\mu\rangle+\lambda\|\boldsymbol{w}\|^2,\\ \text{s.t.} \quad & (\mu-\hat{\mu}(\widetilde{S_N}))^\top\hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))(\mu-\hat{\mu}(\widetilde{S_N}))\le\beta,\end{aligned} \tag{16}$$

This optimization problem Eq. (16) can be solved by the Alternative Convex Search (ACS) method. To be specific, after fixing μ, we can simply use the gradient descent algorithm to solve the minimization problem on $\boldsymbol{w}$, which is

$$\begin{aligned}\min_{\boldsymbol{w}} \quad & \frac{1}{n}\sum_{i=1}^{k}\ell(\widetilde{y}_i\langle \boldsymbol{w},\mathbf{x}_i\rangle)+\frac{1}{2n}\sum_{i=k+1}^{n}\varphi(\widetilde{y}_i\langle \boldsymbol{w},\mathbf{x}_i\rangle)\\ & +\frac{c}{1-2p\eta}\langle \boldsymbol{w},\mu\rangle+\lambda\|\boldsymbol{w}\|^2.\end{aligned} \tag{17}$$

When $\boldsymbol{w}$ is fixed, after ignoring some constant terms, the optimization problem regarding μ can be expressed as

$$\begin{aligned}\min_{\mu} \quad & c\langle \boldsymbol{w},\mu\rangle,\\ \text{s.t.} \quad & (\mu-\hat{\mu}(\widetilde{S_N}))^\top\hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))(\mu-\hat{\mu}(\widetilde{S_N}))\le\beta.\end{aligned} \tag{18}$$

To cope with this constrained optimization problem, this paper introduces a Lagrangian variable ρ, and then the following formulation can be obtained, namely

$$\begin{aligned}L(\mu,\beta)=&c\langle \boldsymbol{w},\mu\rangle-\rho(\mu-\hat{\mu}(\widetilde{S_N}))^\top\hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))(\mu-\hat{\mu}(\widetilde{S_N}))\\ &+\rho\beta.\end{aligned} \tag{19}$$

By setting $\frac{\partial L(\mu,\beta)}{\partial\mu}=0$, we can get

$$\mu=\frac{c}{2\rho}(\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1}\boldsymbol{w}+\hat{\mu}(\widetilde{S_N}). \tag{20}$$

By plugging Eq. (20) into Eq. (18), we have

$$\begin{aligned}\min_{\rho} \quad & \frac{c}{2\rho}\boldsymbol{w}^\top(\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1}\boldsymbol{w},\\ \text{s.t.} \quad & \frac{c^2}{4\rho^2}\boldsymbol{w}^\top(\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1}\boldsymbol{w}\le\beta,\end{aligned} \tag{21}$$

of which the closed-form solution is $\rho=-\frac{c}{2}\sqrt{\boldsymbol{w}^\top(\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1}\boldsymbol{w}/\beta}$. By further plugging it into Eq. (20), the solution of Eq. (18) is derived as

$$\mu=\hat{\mu}(\widetilde{S_N})+(\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1}\boldsymbol{w}\sqrt{\beta/(\boldsymbol{w}^\top(\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1}\boldsymbol{w})}. \tag{22}$$

The detailed procedure of above optimization process is presented in Algorithm 2.

6.2 The Kernelized LDCE Model

In this section, we extend the above LDCE model to non-linear case by introducing the kernel trick, and the resulting algorithm is then called "Kernelized LDCE" (KLDCE). Given the linear decision function as $f(\mathbf{x})=\langle \boldsymbol{w},\mathbf{x}\rangle+b=\boldsymbol{w}^\top\mathbf{x}+b$ where $\boldsymbol{w}$ is the parameter vector and b is the bias term, by employing the slack variables ξ_i and δ_i, Eq. (16) can then be reformulated as

$$\begin{aligned}\min_{\boldsymbol{w},\mu,\xi_i,\delta_i} \quad & \frac{1}{n}\sum_{i=1}^{k}\xi_i+\frac{1}{2n}\sum_{i=k+1}^{n}\xi_i+\frac{1}{2n}\sum_{i=k+1}^{n}\delta_i\\ & +C(\boldsymbol{w}^\top\mu+b)+\lambda\|\boldsymbol{w}\|^2,\\ \text{s.t.} \quad & (\mu-\hat{\mu}(\widetilde{S_N}))^\top\hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))(\mu-\hat{\mu}(\widetilde{S_N}))\le\beta,\\ & \widetilde{y}_i(\boldsymbol{w}^\top\mathbf{x}_i+b)\geqslant 1-\xi_i,\quad \xi_i\geqslant 0, i=1,2,\cdots,n,\\ & \widetilde{y}_i(\boldsymbol{w}^\top\mathbf{x}_i+b)\leqslant \delta_i-1,\quad \delta_i\geqslant 0, i=k+1,\cdots,n.\end{aligned} \tag{23}$$

After introducing the dual variables $\boldsymbol{\alpha}=(\alpha_1,\cdots,\alpha_n)$, $\boldsymbol{\gamma}=(\gamma_{k+1},\cdots,\gamma_n)$, and also the kernel function $G(\cdot)$, the dual problem can be expressed as

$$\begin{aligned}\max_{\boldsymbol{\alpha},\boldsymbol{\gamma},\mu} \quad & \sum_{i=1}^{n}\alpha_i+\sum_{i=k+1}^{n}\gamma_i-\frac{1}{4\lambda}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\widetilde{y}_i\widetilde{y}_jG(\mathbf{x}_i,\mathbf{x}_j)\\ & +\frac{1}{2\lambda}\sum_{i=1}^{n}\sum_{j=k+1}^{n}\alpha_i\gamma_j\widetilde{y}_i\widetilde{y}_jG(\mathbf{x}_i,\mathbf{x}_j)\\ & -\frac{1}{4\lambda}\sum_{i=k+1}^{n}\sum_{j=k+1}^{n}\gamma_i\gamma_j\widetilde{y}_i\widetilde{y}_jG(\mathbf{x}_i,\mathbf{x}_j)\\ & +\frac{C}{2\lambda}\sum_{i=1}^{n}\alpha_i\widetilde{y}_iG(\mathbf{x}_i,\mu)-\frac{C}{2\lambda}\sum_{i=k+1}^{n}\gamma_i\widetilde{y}_iG(\mathbf{x}_i,\mu)\\ & -\frac{C^2}{2\lambda}G(\mu,\mu)\\ \text{s.t.} \quad & (\mu-\hat{\mu}(\widetilde{S_N}))^\top\hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))(\mu-\hat{\mu}(\widetilde{S_N}))\le\beta,\\ & C-\sum_{i=1}^{n}\alpha_i\widetilde{y}_i+\sum_{i=k+1}^{n}\gamma_i\widetilde{y}_i=0,\\ & C_1\ge\alpha_i\ge 0,\quad C_2\ge\gamma_i\ge 0.\end{aligned} \tag{24}$$

It is worth pointing out that Eq. (24) is a quadratic programming problem regarding $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, and the subproblem regarding μ can also be easily solved via the same way as in Section 6.1, so Eq. (24) can be efficiently solved by employing the ACS and SMO methods. The detailed derivations for the dual form in Eq. (24) and its optimization process are put into the **appendix**. After solving Eq. (24), we can get the final classifier as

$$\begin{aligned}f(\mathbf{x})=&\frac{1}{2\lambda}\sum_{i=1}^{n}\alpha_i\widetilde{y}_iG(\mathbf{x},\mathbf{x}_i)-\frac{1}{2\lambda}\sum_{i=k+1}^{n}\gamma_i\widetilde{y}_iG(\mathbf{x},\mathbf{x}_i)\\ &-\frac{1}{2\lambda}CG(\mathbf{x},\mu^\star)+b,\end{aligned} \tag{25}$$

where $\mu^\star$ is the optimal value of μ as explained above.

7 Theoretical Analyses

In this section, we study the generalization ability of KLDCE algorithm, and show that the empirical risk of an arbitrary classifier learned by KLDCE algorithm will converge to its expected classification. The linear LDCE can be regarded as a special case

of KLDCE with linear kernel, so the theoretical analyses presented below are also applicable to LDCE.

Let $\phi : \mathcal{X} \to \mathcal{H}$ be a mapping of input features to a Hilbert space $\mathcal{H}$ with inner product $\langle \cdot, \cdot \rangle$, then a vector $\boldsymbol{w}_{\mathcal{H}} \in \mathcal{H}$ can be used to predict the label of $\mathbf{x}$ as $y = \langle \boldsymbol{w}_{\mathcal{H}}, \phi(\mathbf{x}) \rangle$. First of all, we define the expected classification error of a classifier $\boldsymbol{w}_{\mathcal{H}}$ by

$$\mathcal{R}(\boldsymbol{w}_{\mathcal{H}}) = \mathbb{E}\left[\mathbb{1}_{Y\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X})} \leq 0\right], \tag{26}$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function that represents the 0-1 loss function, and $(\mathbf{X}, Y)$ stands for a pair of random variables distributed from the unknown joint distribution $\mathcal{D}$ on the set $\mathcal{X} \times \mathcal{Y}$ which has been defined in Section 3.

Since there are only positive and unlabeled data available for training, we need to rewrite the expected risk. We set

$$\mathcal{R}_1(\boldsymbol{w}_{\mathcal{H}}) = \int P(\phi(\mathbf{X})|Y=1)\mathbb{1}_{\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X})\leq 0} d\phi(\mathbf{X}) \tag{27}$$

and

$$\mathcal{R}_{-1}(\boldsymbol{w}_{\mathcal{H}}) = \int P(\phi(\mathbf{X})|Y=-1)\mathbb{1}_{\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X})\leq 0} d\phi(\mathbf{X}), \tag{28}$$

which denote the false positive risk and false negative risk, respectively, and then we have

$$\mathcal{R}(\boldsymbol{w}_{\mathcal{H}}) = \mathbb{E}[\mathbb{1}_{Y\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X})\leq 0}] = p\mathcal{R}_1(\boldsymbol{w}_{\mathcal{H}}) + (1-p)\mathcal{R}_{-1}(\boldsymbol{w}_{\mathcal{H}}), \tag{29}$$

where p is the positive prior $P(Y=1)$. Moreover, we define the risk on unlabeled set as

$$\begin{aligned}
\mathcal{R}_U(\boldsymbol{w}_{\mathcal{H}}) &= \mathbb{E}[\mathbb{1}_{\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X})\geq 0}] \\
&= \int P(\phi(\mathbf{X}))\mathbb{1}_{\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X})\geq 0} d\phi(\mathbf{X}) \\
&= \int (P(\phi(\mathbf{X})|Y=1)P(Y=1) \\
&\quad + P(\phi(\mathbf{X})|Y=-1)P(Y=-1))\mathbb{1}_{\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X})\geq 0} d\phi(\mathbf{X}) \\
&= p(1-\mathcal{R}_1(\boldsymbol{w}_{\mathcal{H}})) + (1-p)\mathcal{R}_{-1}(\boldsymbol{w}_{\mathcal{H}}).
\end{aligned} \tag{30}$$

By combining Eqs. (29) and (30), we can obtain the expected risk on PU datasets, namely

$$\mathcal{R}(\boldsymbol{w}_{\mathcal{H}}) = 2p\mathcal{R}_1(\boldsymbol{w}_{\mathcal{H}}) + \mathcal{R}_U(\boldsymbol{w}_{\mathcal{H}}) - p. \tag{31}$$

Furthermore, the empirical margin error of the classifier $\boldsymbol{w}_{\mathcal{H}}$ can be defined as

$$\begin{aligned}
\hat{\mathcal{R}}^{\rho}(\boldsymbol{w}_{\mathcal{H}}) = &\frac{2p}{k}\sum_{i=1}^{k}\mathbb{1}_{\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X})\leq\rho} + \frac{1}{2(n-k)}\sum_{i=k+1}^{n}\mathbb{1}_{-\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X})\leq\rho} \\
&+ \frac{1}{2(n-k)}\sum_{i=k+1}^{n}\mathbb{1}_{-\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X})\geq-\rho} + \frac{c}{1-2p\eta}\langle \boldsymbol{w}_{\mathcal{H}}, \mu_{\phi}\rangle - p,
\end{aligned} \tag{32}$$

where ρ is the margin, and μ_{ϕ} is the centroid of $\phi(\mathbf{X})$ where $\mathbf{X} \in \mathcal{X}$.

Let $\hat{\boldsymbol{w}}_{\mathcal{H}}$ be any classifier output learned by KLDCE algorithm, we aim to derive the upper bound of the generalization error, namely

$$\mathcal{R}(\hat{\boldsymbol{w}}_{\mathcal{H}}) - \hat{\mathcal{R}}^{\rho}(\hat{\boldsymbol{w}}_{\mathcal{H}}). \tag{33}$$

Theorem 4. *Assume that $\forall \mathbf{x} \in \mathcal{X}, \|\phi(\mathbf{x})\| \leq B$. Let $\hat{\boldsymbol{w}}_{\mathcal{H}}$ be the classifier parameter learned by the KLDCE algorithm. For any $\delta > 0$, with probability at least $1-\delta$, we have*

$$\begin{aligned}
\mathcal{R}(\hat{\boldsymbol{w}}_{\mathcal{H}}) - \hat{\mathcal{R}}^{\rho}(\hat{\boldsymbol{w}}_{\mathcal{H}}) \leq &\frac{C^*B^2 + B\sqrt{C^{*2}B^2 + 4\lambda}}{\rho\lambda\sqrt{k}} \\
&+ \frac{C^*B^2 + B\sqrt{C^{*2}B^2 + 4\lambda}}{\rho\lambda\sqrt{n-k}} + \sqrt{\frac{\ln(1/\delta)}{2n}},
\end{aligned} \tag{34}$$

where $C^* = -\frac{c}{1-2p\eta} = \frac{(n-k)(1-\eta)}{n(1-\eta)-2k\eta}$.

Before providing the proof of this theorem, we present some necessary definitions and lemmas in advance.

Definition 1. *(Rademacher Complexity, [45]) Let $\theta = \{\theta_1, \ldots, \theta_n\}$ be a set of independent Rademacher variables which are uniformly sampled from $\{-1, 1\}$. Let $v_1, \ldots, v_n$ be an independent distributed sample set and $\mathcal{F}$ a function class, then the Rademacher complexity is defined as:*

$$\mathcal{R}_n(\mathcal{F}) = \mathbb{E}_{\theta}[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\theta_i f(v_i)]. \tag{35}$$

Lemma 1. *(Generalization bound, [45]) Let $\mathcal{F}$ be a $[0,1]$ valued function class on $\mathcal{X}$ and $f \in \mathcal{F}$. Given $\mathbf{X}_1, \cdots, \mathbf{X}_n \in \mathcal{X}$ are i.i.d. variables, then for any $\delta > 0$, with probability at least $1-\delta$, we have*

$$\sup_{f\in\mathcal{F}}\left(\mathbb{E}f(\mathbf{X}) - \frac{1}{n}\sum_{i=1}^{n}f(\mathbf{X}_i)\right) \leq 2\mathcal{R}_n(\mathcal{F}) + \sqrt{\frac{\ln(1/\delta)}{2n}}. \tag{36}$$

Lemma 2. *(Talagrand contraction Lemma, [45]) If $\ell\colon \mathbb{R} \to \mathbb{R}$ is Lipschitz continuous with constant Ω and satisfies $\ell(0) = 0$, then*

$$\mathcal{R}_n(\ell \circ \mathcal{F}) \leq \Omega\mathcal{R}_n(\mathcal{F}), \tag{37}$$

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

Now we present the formal proof for Theorem 4.

Proof. Firstly, we introduce the following functions that

$$\ell_1^{\rho}(x) = \begin{cases} 0 & if \quad x \geq \rho, \\ 1 - x/\rho & if \quad 0 \leq x \leq \rho, \\ 1 & otherwise, \end{cases} \tag{38}$$

$$\ell_2^{\rho}(x) = \begin{cases} 0 & if \quad x \leq -\rho, \\ 1 + x/\rho & if \quad -\rho \leq x \leq 0, \\ 1 & otherwise, \end{cases} \tag{39}$$

where ρ is the margin as defined in Eq. (32) and the above introduced functions Eqs. (38) and (39) are uniformly denoted as $\ell^{\rho}(x)$. Let

$$\begin{aligned}
\mathcal{R}(\ell^{\rho} \circ \boldsymbol{w}_{\mathcal{H}}) &= \mathbb{E}[\ell^{\rho}(Y\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X}))] \\
&= 2p\mathcal{R}_1(\ell^{\rho} \circ \boldsymbol{w}_{\mathcal{H}}) + \mathcal{R}_U(\ell^{\rho} \circ \boldsymbol{w}_{\mathcal{H}}) - p,
\end{aligned} \tag{40}$$

$$\begin{aligned}
\hat{\mathcal{R}}(\ell^{\rho} \circ \boldsymbol{w}_{\mathcal{H}}) = &\frac{2p}{k}\sum_{i=1}^{k}\ell_1^{\rho}(\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X}_i)) + \frac{1}{2(n-k)}\sum_{i=k+1}^{n}\ell_1^{\rho}(-\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X}_i)) \\
&+ \frac{1}{2(n-k)}\sum_{i=k+1}^{n}\ell_2^{\rho}(-\boldsymbol{w}_{\mathcal{H}}^{\top}\phi(\mathbf{X}_i)) + \frac{c}{1-2p\eta}\langle \boldsymbol{w}_{\mathcal{H}}, \mu_{\phi}\rangle - p,
\end{aligned} \tag{41}$$

where $\ell^\rho \circ \boldsymbol{w}_\mathcal{H}$ represents the composite function, and

$$\mathcal{R}_1(\ell^\rho \circ \boldsymbol{w}_\mathcal{H}) = \int P(\phi(\mathbf{X})|Y=1)\ell_1^\rho(\boldsymbol{w}_\mathcal{H}^\top\phi(\mathbf{X}))d\phi(\mathbf{X}), \tag{42}$$

and

$$\mathcal{R}_U(\ell^\rho \circ \boldsymbol{w}_\mathcal{H}) = \int P(\phi(\mathbf{X})|Y=1)\ell_1^\rho(-\boldsymbol{w}_\mathcal{H}^\top\phi(\mathbf{X}))d\phi(\mathbf{X}). \tag{43}$$

It can be easily verified that $\mathcal{R}(\boldsymbol{w}_\mathcal{H}) \leq \mathcal{R}(\ell^\rho \circ \boldsymbol{w}_\mathcal{H})$ and $\hat{\mathcal{R}}^\rho(\boldsymbol{w}_\mathcal{H}) \geq \hat{\mathcal{R}}(\ell^\rho \circ \boldsymbol{w}_\mathcal{H})$, which imply that

$$\mathcal{R}(\boldsymbol{w}_\mathcal{H}) - \hat{\mathcal{R}}^\rho(\boldsymbol{w}_\mathcal{H}) \leq \mathcal{R}(\ell^\rho \circ \boldsymbol{w}_\mathcal{H}) - \hat{\mathcal{R}}(\ell^\rho \circ \boldsymbol{w}_\mathcal{H}). \tag{44}$$

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

$$\mathcal{R}(\boldsymbol{w}_\mathcal{H}) - \hat{\mathcal{R}}^\rho(\boldsymbol{w}_\mathcal{H}) \leq \sup_{\boldsymbol{w}_\mathcal{H}\in\mathcal{W}}(\mathcal{R}(\ell^\rho \circ \boldsymbol{w}_\mathcal{H}) - \hat{\mathcal{R}}(\ell^\rho \circ \boldsymbol{w}_\mathcal{H})). \tag{45}$$

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

$$\begin{aligned}&\sup_{\boldsymbol{w}_\mathcal{H}\in\mathcal{W}}(\mathcal{R}(\ell^\rho \circ \boldsymbol{w}_\mathcal{H}) - \hat{\mathcal{R}}(\ell^\rho \circ \boldsymbol{w}_\mathcal{H}))\\ &\leq 2\mathcal{R}_P(\ell^\rho \circ \boldsymbol{w}_\mathcal{H}) + 2\mathcal{R}_U(\ell^\rho \circ \boldsymbol{w}_\mathcal{H}) + \sqrt{\frac{\ln(1/\delta)}{2n}},\end{aligned} \tag{46}$$

where

$$\mathcal{R}_P(\ell^\rho \circ \boldsymbol{w}_\mathcal{H}) = \mathbb{E}[\sup_{\boldsymbol{w}_\mathcal{H}\in\mathcal{W}}\frac{1}{k}\sum_{i=1}^{k}\theta_i\ell_1^\rho(\boldsymbol{w}_\mathcal{H}^\top\phi(\mathbf{X}))] \tag{47}$$

and

$$\mathcal{R}_U(\ell^\rho \circ \boldsymbol{w}_\mathcal{H}) = \mathbb{E}[\sup_{\boldsymbol{w}_\mathcal{H}\in\mathcal{W}}\frac{1}{n-k}\sum_{i=k+1}^{n}\theta_i\ell_1^\rho(-\boldsymbol{w}_\mathcal{H}^\top\phi(\mathbf{X}))]. \tag{48}$$

To obtain the upper bounds of the Rademacher complexities in Eqs. (47) and (48), we firstly derive an upper bound for the classifier in $\mathcal{W}$. Specifically, due to the optimality of any $\boldsymbol{w}_\mathcal{H} \in \mathcal{W}$, we have

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{k}[1-y_i\langle\boldsymbol{w}_\mathcal{H},\phi(\mathbf{x}_i)\rangle]_+ + \frac{1}{2n}\sum_{i=k+1}^{n}[1-y_i\langle\boldsymbol{w}_\mathcal{H},\phi(\mathbf{x}_i)\rangle]_+\\ &+ \frac{1}{2n}\sum_{i=k+1}^{n}[1+y_i\langle\boldsymbol{w}_\mathcal{H},\phi(\mathbf{x}_i)\rangle]_+ + \frac{c}{1-2p\eta}\langle\boldsymbol{w}_\mathcal{H},\mu_\phi\rangle + \lambda\|\boldsymbol{w}_\mathcal{H}\|^2\\ &\leq \frac{1}{n}\sum_{i=1}^{k}[1-y_i\langle\mathbf{0},\phi(\mathbf{x}_i)\rangle]_+ + \frac{1}{2n}\sum_{i=k+1}^{n}[1-y_i\langle\mathbf{0},\phi(\mathbf{x}_i)\rangle]_+\\ &+ \frac{1}{2n}\sum_{i=k+1}^{n}[1+y_i\langle\mathbf{0},\phi(\mathbf{x}_i)\rangle]_+ + \frac{c}{1-2p\eta}\langle\mathbf{0},\mu_\phi\rangle + \lambda\|\mathbf{0}\|^2\\ &= 1.\end{aligned} \tag{49}$$

This inequality implies that

$$\lambda\|\boldsymbol{w}_\mathcal{H}\|^2 \leq 1 - \frac{c}{1-2p\eta}\langle\boldsymbol{w}_\mathcal{H},\mu_\phi\rangle. \tag{50}$$

Due to $c = -\frac{(n-k)}{2n}$ and $p = k/n(1-\eta)$, where η denotes the false negative rate $P(\widetilde{Y}=-1|Y=1)$, we have

$$-\frac{c}{1-2p\eta} = \frac{(n-k)(1-\eta)}{n(1-\eta)-2k\eta}, \tag{51}$$

which is further marked as C^*. Thus, Eq. (50) can be converted to

$$\lambda\|\boldsymbol{w}_\mathcal{H}\|^2 \leq 1 + C^*\langle\boldsymbol{w}_\mathcal{H},\mu_\phi\rangle. \tag{52}$$

Similarly, since μ_ϕ is the optimal solution for model, it is very close to the true centroid of the mapped corrupted negative set (*i.e.* unlabeled set), which is $\mu_\phi(\phi(\widetilde{S_N})) = \frac{1}{n-k}\sum_{i=k+1}^{n}\widetilde{y_i}\phi(\mathbf{x}_i)$, so we have

$$\begin{aligned}1 + C^*\langle\boldsymbol{w}_\mathcal{H},\mu_\phi\rangle &\leq 1 + C^*\langle\boldsymbol{w}_\mathcal{H},\mu_\phi(\widetilde{S_N})\rangle\\ &\leq 1 + C^*\|\boldsymbol{w}_\mathcal{H}\|\|\mu_\phi(\phi(\widetilde{S_N}))\|.\end{aligned} \tag{53}$$

The following thing is to derive the upper bound of $\|\mu_\phi(\phi(\widetilde{S_N}))\|$. As $\|\phi(\mathbf{x})\| \leq B$, we have

$$\|\mu_\phi(\phi(\widetilde{S_N}))\| \leq \frac{1}{n-k}\sum_{i=k+1}^{n}\|\widetilde{y_i}\|\|\phi(\mathbf{x}_i)\| \leq B. \tag{54}$$

By combining Eqs. (52) and (53), we can obtain

$$\lambda\|\boldsymbol{w}_\mathcal{H}\|^2 - C^*B\|\boldsymbol{w}_\mathcal{H}\| - 1 \leq 0. \tag{55}$$

Therefore, we know $\|\boldsymbol{w}_\mathcal{H}\| \leq \frac{C^*B+\sqrt{C^{*2}B^2+4\lambda}}{2\lambda}$ by solving the above inequality.

Now, we are going to achieve the upper bounds of $\mathcal{R}_P(\ell^\rho \circ \boldsymbol{w}_\mathcal{H})$ and $\mathcal{R}_U(\ell^\rho \circ \boldsymbol{w}_\mathcal{H})$, respectively. Specifically, since the function $\ell^\rho(x)$ is $1/\rho$-Lipschtiz, by using Lemma 2, we have

$$\begin{aligned}\mathcal{R}_P(\ell^\rho \circ \mathcal{W}) &\leq \frac{1}{\rho}\mathcal{R}_P(\mathcal{W})\\ &= \frac{1}{\rho}\mathbb{E}[\sup_{\boldsymbol{w}_\mathcal{H}\in\mathcal{W}}\frac{1}{k}\sum_{i=1}^{k}\theta_i\boldsymbol{w}_\mathcal{H}^\top\mathbf{X}_i]\\ &= \frac{1}{\rho}\mathbb{E}[\sup_{\boldsymbol{w}_\mathcal{H}\in\mathcal{W}}\langle\boldsymbol{w}_\mathcal{H},\frac{1}{k}\sum_{i=1}^{k}\theta_i\phi(\mathbf{X}_i)\rangle]\\ &\overset{1}{\leq} \frac{1}{\rho k}\mathbb{E}[\sup_{\boldsymbol{w}_\mathcal{H}\in\mathcal{W}}\|\boldsymbol{w}_\mathcal{H}\|\|\sum_{i=1}^{k}\theta_i\phi(\mathbf{X}_i)\|]\\ &\leq \frac{C^*B+\sqrt{C^{*2}B^2+4\lambda}}{2\lambda\rho k}\mathbb{E}[\|\sum_{i=1}^{k}\theta_i\phi(\mathbf{X}_i)\|]\\ &\overset{2}{\leq} \frac{C^*B+\sqrt{C^{*2}B^2+4\lambda}}{2\lambda\rho k}\sqrt{\sum_{i=1}^{k}\mathbb{E}[\|\phi(\mathbf{X}_i)\|^2]}\\ &\leq \frac{C^*B^2+B\sqrt{C^{*2}B^2+4\lambda}}{2\lambda\rho\sqrt{k}},\end{aligned} \tag{56}$$

where the 1^{st} inequality holds because of the Cauchy-Schwarz inequality, and the 2^{nd} inequality holds because of the Jensen's inequality. Similarly, we have

$$\mathcal{R}_U(\ell^\rho \circ \mathcal{W}) \leq \frac{C^*B^2+B\sqrt{C^{*2}B^2+4\lambda}}{2\lambda\rho\sqrt{n-k}}. \tag{57}$$

By combining inequalities Eqs. (45), (46), (55) and (56), we can get the result of Theorem 4. □

As mentioned in Section 3, the parameter η is usually estimated from data in practical situations, so it might be slightly deviated from its real value. Although the real value of η in Theorem 4 is treated as known, the derived generalization bound is also applicable even if η is estimated. Specifically, by denoting the right-hand side of Eq. (34) as a function of η, *i.e.* $\psi(\eta)$, we have

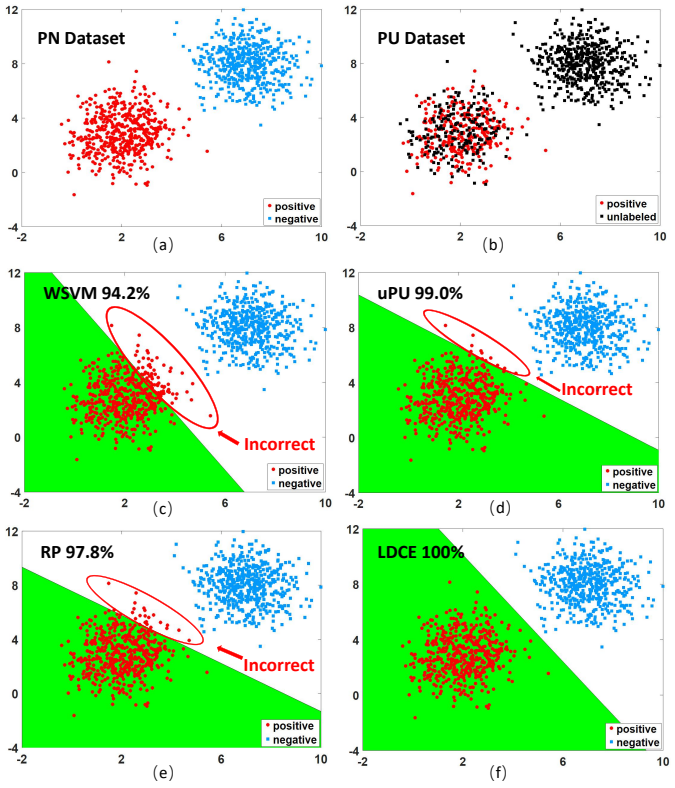

Fig. 2: The performances of various methods on synthetic dataset, where red, blue and black dots denote positive, negative and unlabeled data points, respectively. (a) shows the real positive and negative examples, (b) shows the positive and unlabeled data for model training, (c)−(f) display the decision boundaries and classification accuracies generated by WSVM [7], uPU [9], RP [47], and LDCE, respectively. The incorrectly classified examples are highlighted by red circles.

that $\mathcal{R}(\hat{\boldsymbol{w}}_{\mathcal{H}}) - \hat{\mathcal{R}}^{\rho}(\hat{\boldsymbol{w}}_{\mathcal{H}}) \leq \psi(\eta)$ with probability $1-\delta$. Furthermore, the false negative rate η can be estimated by methods for mixture proportion estimation such as [40], [46]. According to the Theorem 13 in [40], we know that under a mild condition of data distribution (*i.e.*, Definition 8 in [40]), with probability $1-4\delta'$, $\eta \leq \hat{\eta} + \mathcal{O}(\sqrt{\log(1/\delta')}(\min(k, n-k))^{-1/2})$ where $\hat{\eta}$ is an empirical estimate for η. Therefore, under the same assumption, with probability $1-\delta-4\delta'$, we conclude that $\mathcal{R}(\hat{\boldsymbol{w}}_{\mathcal{H}}) - \hat{\mathcal{R}}^{\rho}(\hat{\boldsymbol{w}}_{\mathcal{H}}) \leq \psi\big(\hat{\eta} + \mathcal{O}(\sqrt{\log(1/\delta')}(\min(k, n-k))^{-1/2})\big)$. This shows that for the proposed method, the estimated $\hat{\boldsymbol{w}}_{\mathcal{H}}$ and $\hat{\eta}$ will converge to the optimal ones when we have sufficient positive and unlabeled data. The impact of the estimated $\hat{\eta}$ on the final classification accuracy will also be empirically studied in Section 8.6.

8 Experiments

To demonstrate the effectiveness of our proposed LDCE and KLDCE, in this section, we perform exhaustive experiments on one synthetic dataset, nine publicly available benchmark datasets and three real-world datasets. Some state-of-the-art methods, such as Weighted SVM (WSVM) [7], unbiased PU (uPU) [9], non-negative PU (nnPU) [29], and Rank Pruning (RP) [47] are employed as the competing methods.

8.1 Synthetic Dataset

To start with, we create a two-dimensional binary dataset composed of two clusters of data points generated from two Gaussian distributions as shown in Figure 2 (a). In this dataset, there are 500 positive examples and 500 negative examples, and each Gaussian constitutes a class. We randomly choose 40% ($\eta = 0.4$) of the original positive examples and then combine them with all negative examples to form the unlabeled set (see Figure 2 (b)). We further run various PU learning methods on the dataset and investigate whether these methods can yield accurate decision boundary for separating true positive and negative examples. Since this synthetic dataset is linearly separable, here we only compare our proposed linear LDCE with other existing PU methods. Moreover, nnPU is also not compared as this method does not output explicit linear decision function. As shown in Figure 2 (c)∼(f), the results of various methods reveal that only our LDCE achieves 100% classification accuracy, which is higher than 94.2%, 99.0%, and 97.8% obtained by WSVM, uPU and RP, respectively. Specifically, uPU, WSVM and RP fail to distinguish the ambiguous examples near the potentially correct decision boundary, and many positive examples are classified erroneously as negative by them. Based on these results, it is clearly demonstrated that LDCE is superior to other existing PU methods on this synthetic dataset.

8.2 UCI Benchmark Dataset

To further demonstrate the effectiveness of our proposed methods, next we conduct extensive experiments on nine benchmark datasets from UCI machine learning repository [48], which include *Vote*, *Balance*, *Breast*, *Australian*, *Banknote*, *Mushroom*, *PhishingWebsites*, *Connect-4* and *Skin*. The brief information of these datasets are presented in Table 1, which indicates the number of examples n, the feature dimensionality d, the number of positive examples n_+, and the number of negative examples n_- for each dataset. For multiclass dataset *Connect-4*, we regard the first class of it as positive while the rest of the classes are regarded as negative. Moreover, all data features are normalized to $[-1, 1]$ in advance.

For each dataset illustrated in Table 1, we partition it into five subsets with almost equal size, which facilitates the subsequent five-fold cross validation. In each training round, we use 80% of the original examples for training while the rest 20% examples are employed for testing. Note that the unlabeled set is composed of the original negative set as well as 20%, 30% or 40% (*i.e.*, $\eta \in \{0.2, 0.3, 0.4\}$) of positive training examples that are randomly selected. Under each η, the formation of training set is kept identical for every compared method to ensure fair comparison. In LDCE and KLDCE, we choose the regularization parameter λ from $\{2^{-4}, \ldots, 2^{4}\}$, and select the parameter β from $\{0.1, 0.2, \ldots, 0.9\}$ via cross validation according to [15]. Moreover, for KLDCE, we select the bandwidth σ of kernel function from $\{2^0, 2^1, 2^2, 2^3\}$. In uPU, the regularization parameter λ is selected from $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$. For nnPU, the parameter β for preventing the overfitting is tuned to the default value 0 on all UCI benchmark datasets. Besides, when implementing uPU and nnPU, we add the positive data back to the unlabeled set to satisfy the sampling requirement of these two methods.

Table 2 summarizes the test results of compared methods averaged over the five trials. Furthermore, we also apply the paired t-test with significance level 0.05 to investigate whether our approach is significantly better than other methods. From the results on all datasets except *Mushroom*, we observe that our two proposed methods LDCE and KLDCE are superior to other PU

TABLE 1: The characteristics of nine UCI datasets.

Dataset	n	d	n_+	n_-
Vote	435	16	267	168
Balance	625	4	288	337
Breast	683	10	143	540
Australian	690	14	370	383
Banknote	1372	4	610	762
Mushroom	8124	112	3916	4208
PhishingWebsites	11055	30	6157	4898
Connect-4	67557	42	44473	23084
Skin	245057	3	50859	194198

methods in most cases. For *Mushroom* data, the performances of nnPU and RP are better than LDCE, but it is still not as good as KLDCE. Table 2 reveals that our approach KLDCE is consistently the best method among all compared methods except on *Banknote* dataset. Instead, our linear LDCE model achieves the top level performance on *Banknote*. Therefore, the results reveal that introducing kernel function can generally help to obtain an improved non-linear classifier.

8.3 Real-world Data

To further evaluate the ability of LDCE and KLDCE in handling complex practical problem, we also conduct the experiments on three real-world datasets, including *USPS*[1], *HockeyFight*[2] and *NBA*[3] datasets. For the experiments on each dataset, we also investigate the test accuracies of our two approaches (LDCE and KLDCE) and other state-of-the-art PU methods.

8.3.1 Handwritten Digit Recognition

The *USPS* dataset is used to assess the ability of various PU learning methods in recognizing the handwritten digits. In this dataset, there are 9298 digit images belonging to 10 classes, *i.e.*, the digits "0"-"9". We adopted the 256-dimensional pixel-wise feature, in which every dimension represents the gray value of corresponding pixel, and the resolution of all images is 16×16. In this paper, the digit images of "0" are regarded as positive, while the rest of the digit images are viewed as negative examples. Therefore, this dataset has 1553 positive examples and 7745 negative examples, and such class imbalance will pose a great challenge for the compared methodologies. Furthermore, we also adopt the same experimental setting as described in Section 8.2 to form the positive set and unlabeled set.

For KLDCE, under $\eta = 0.2$, we set $\lambda = 5, \beta = 0.7$ and $\sigma = 5$ to get the best performance. Similarly, to obtain satisfactory results, we set $\lambda = 5, \beta = 0.9$ and $\sigma = 5$ when $\eta = 0.3$ and $\eta = 0.4$. In nnPU, the parameters are set as $\gamma = 1$ and $\beta = 0.2$ for $\eta = 0.2$. In the situations under $\eta = 0.3$ and $\eta = 0.4$, the parameters of nnPU are optimally tuned as $\gamma = 0.2$ and $\beta = 0.2$. In uPU, the parameter β is adaptively determined as 0.021, 0.003 and 0.167 when $\eta = 0.2$, $\eta = 0.3$ and $\eta = 0.4$, respectively.

The obtained test accuracies of various methods are reported in Table 3, from which we observe that our proposed approaches LDCE and KLDCE are always the two best methods among all methods under different η. In particular, the proposed KLDCE achieves very high accuracy of 91.9% even when 40% positive examples are mixed to the unlabeled set, which is a very impressive result. The results shown in Table 3 also demonstrate the effectiveness of our proposed approach in dealing with imbalanced data.

1. http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html
2. http://visilab.etsii.uclm.es/personas/oscar/FightDetection/index.html
3. https://data.world/exercises/logistic-regression-exercise-1

Fig. 3: Examples of fighting and non-fighting video frames in *HockeyFight* dataset.

8.3.2 Violent Behavior Detection

Nowadays, there is a surge of research interests in detecting the violent behavior in intelligent surveillance system due to the great practical value. In this section, we apply the proposed approach and other PU methods on the *HockeyFight* dataset for fight behavior detection. The *HockeyFight* dataset is composed of 1000 video clips collected in ice hockey competitions, of which 500 contain fight behavior and 500 are non-fight sequences. Some examples of this dataset are shown in Figure 3.

We try to classify the video clips as fighting and non-fighting by using various PU learning methods including WSVM, uPU, nnPU, RP, our LDCE and KLDCE. Similar to [49], after adopting the space-time interest point (STIP) and motion SIFT (MoSIFT) as action descriptors, each video clip of the dataset can be represented as a histogram over 100 visual words by further using the Bag-of-Words (BoW) quantization. Therefore, every clip belonging to the dataset was characterized by a 100-dimensional feature vector. Similar to the setting in Section 8.2, the proportions of training examples and test examples are also respectively maintained as 80% and 20%, and the dataset partitions are kept identical for all compared methods.

For the experiments on this dataset, the key parameters in LDCE or KLDCE are set to $\lambda = 9, \beta = 0.3$ and $\sigma = 2$ when $\eta = 0.2$ and 0.3. When η is equal to 0.4, the regularization parameter λ is adjusted to 7, and the parameter β and σ are set to the same value as those for $\eta = 0.2$ and $\eta = 0.3$. In nnPU, the parameters are set as $\gamma = 1$ and $\beta = 0$.

Table 3 presents the test accuracies rendered by the compared methods, which clearly validate the top-level performance of our two proposed methods (LDCE and KLDCE) among all the approaches in this dataset. It can be observed that our LDCE and KLDCE are able to generate very impressive results even though a large number of positive examples are "hidden" in the original unlabeled set (*e.g.* $\eta = 0.4$), which outperforms other methods with noticeable margins.

8.3.3 Prediction of Career Longevity for NBA Rookies

Based on the statistics of NBA rookies, which players are potentially able to have 5 years or more career longevity? To study this problem, we adopt the *NBA* dataset in which the real statistics of NBA players are provided. The *NBA* dataset is composed of 1340 examples (*i.e.*, players) and each example has 22 different attributes. The attributes include "Games Played", "Minutes Played", "Points Per Game", "Field Goal Made", "Three

TABLE 2: Comparison of the mean test accuracies of various approaches on UCI datasets. The black "$\surd$" ("$\times$") denotes that our KLDCE approach is significantly better (worse) than the corresponding existing methods revealed by the paired t-test with significance level 0.05. Similarly, the magenta "$\surd$" ("$\times$") denotes that our LDCE approach is significantly better (worse) than the corresponding existing methods revealed by the paired t-test. The best and second best results on each dataset are indicated in red and blue, respectively.

Dataset	η	WSVM [7]	uPU [9]	nnPU [29]	RP [47]	LDCE	KLDCE
Vote	0.2	0.876 ± 0.04 $\surd$ $\surd$	0.899 ± 0.02	0.896 ± 0.02	0.897 ± 0.05	0.901 ± 0.03	0.906 ± 0.03
	0.3	0.862 ± 0.05 $\surd$ $\surd$	0.893 ± 0.02	0.883 ± 0.03 $\surd$ $\surd$	0.890 ± 0.02	0.894 ± 0.04	0.901 ± 0.03
	0.4	0.801 ± 0.09 $\surd$ $\surd$	0.821 ± 0.03 $\surd$	0.808 ± 0.05 $\surd$ $\surd$	0.821 ± 0.02 $\surd$	0.823 ± 0.05	0.892 ± 0.04
Balance	0.2	0.853 ± 0.03 $\surd$ $\surd$	0.853 ± 0.04 $\surd$ $\surd$	0.850 ± 0.04 $\surd$ $\surd$	0.858 ± 0.03 $\surd$ $\surd$	0.910 ± 0.02	0.922 ± 0.01
	0.3	0.773 ± 0.12 $\surd$ $\surd$	0.819 ± 0.12 $\surd$	0.787 ± 0.06 $\surd$ $\surd$	0.773 ± 0.13 $\surd$ $\surd$	0.827 ± 0.03	0.882 ± 0.03
	0.4	0.629 ± 0.05 $\surd$ $\surd$	0.711 ± 0.04 $\surd$	0.716 ± 0.07 $\surd$	0.713 ± 0.03 $\surd$	0.721 ± 0.03	0.842 ± 0.02
Breast	0.2	0.947 ± 0.01 $\surd$ $\surd$	0.962 ± 0.02 $\surd$	0.956 ± 0.03 $\surd$ $\surd$	0.934 ± 0.04 $\surd$ $\surd$	0.974 ± 0.02	0.985 ± 0.01
	0.3	0.802 ± 0.02 $\surd$	0.836 ± 0.02 $\surd$ $\surd$	0.842 ± 0.01	0.839 ± 0.02 $\surd$	0.845 ± 0.03	0.866 ± 0.01
	0.4	0.794 ± 0.01 $\surd$ $\surd$	0.825 ± 0.02 $\surd$ $\surd$	0.837 ± 0.02 $\surd$	0.815 ± 0.03 $\surd$ $\surd$	0.845 ± 0.04	0.860 ± 0.03
Australian	0.2	0.810 ± 0.04 $\surd$ $\surd$	0.840 ± 0.03 $\surd$	0.856 ± 0.02 $\surd$	0.858 ± 0.01 $\surd$	0.849 ± 0.03	0.871 ± 0.02
	0.3	0.828 ± 0.01 $\surd$ $\surd$	0.835 ± 0.03 $\surd$ $\surd$	0.853 ± 0.02 $\surd$	0.858 ± 0.03	0.852 ± 0.02	0.865 ± 0.02
	0.4	0.832 ± 0.04	0.828 ± 0.04 $\surd$ $\surd$	0.848 ± 0.03 $\surd$	0.849 ± 0.03 $\surd$	0.851 ± 0.02	0.863 ± 0.02
Banknote	0.2	0.966 ± 0.01 $\surd$ $\surd$	0.969 ± 0.01 $\surd$ $\surd$	0.971 ± 0.02	0.963 ± 0.01 $\surd$ $\surd$	0.977 ± 0.01	0.975 ± 0.01
	0.3	0.953 ± 0.02 $\surd$ $\surd$	0.960 ± 0.01 $\surd$ $\surd$	0.970 ± 0.06	0.963 ± 0.01 $\surd$ $\surd$	0.975 ± 0.02	0.972 ± 0.02
	0.4	0.937 ± 0.01 $\surd$ $\surd$	0.958 ± 0.00 $\surd$ $\surd$	0.965 ± 0.01 $\surd$	0.959 ± 0.01 $\surd$	0.973 ± 0.01	0.968 ± 0.02
Mushroom	0.2	0.658 ± 0.09 $\surd$ $\surd$	0.841 ± 0.01 $\surd$	0.918 ± 0.00 $\times$	0.912 ± 0.01 $\surd$ $\times$	0.849 ± 0.03	0.921 ± 0.02
	0.3	0.688 ± 0.09 $\surd$ $\surd$	0.787 ± 0.01 $\surd$ $\times$	0.865 ± 0.01 $\times$	0.849 ± 0.01 $\surd$ $\times$	0.750 ± 0.01	0.866 ± 0.02
	0.4	0.643 ± 0.01 $\surd$ $\surd$	0.728 ± 0.01 $\surd$ $\times$	0.782 ± 0.01 $\surd$ $\times$	0.801 ± 0.01 $\surd$ $\times$	0.658 ± 0.05	0.832 ± 0.02
PhishingWebsites	0.2	0.794 ± 0.03 $\surd$ $\surd$	0.819 ± 0.01 $\surd$ $\surd$	0.834 ± 0.01 $\surd$ $\surd$	0.835 ± 0.01 $\surd$	0.840 ± 0.02	0.842 ± 0.02
	0.3	0.691 ± 0.06 $\surd$ $\surd$	0.806 ± 0.01 $\surd$ $\surd$	0.825 ± 0.00 $\surd$	0.827 ± 0.01	0.827 ± 0.02	0.832 ± 0.02
	0.4	0.625 ± 0.03 $\surd$ $\surd$	0.782 ± 0.01 $\surd$ $\surd$	0.819 ± 0.01 $\surd$	0.790 ± 0.01 $\surd$ $\surd$	0.821 ± 0.01	0.830 ± 0.02
Connect-4	0.2	0.664 ± 0.02 $\surd$ $\surd$	0.687 ± 0.03 $\surd$ $\surd$	0.725 ± 0.01 $\surd$ $\surd$	0.700 ± 0.00 $\surd$ $\surd$	0.740 ± 0.03	0.813 ± 0.01
	0.3	0.558 ± 0.03 $\surd$ $\surd$	0.651 ± 0.01 $\surd$ $\surd$	0.691 ± 0.01 $\surd$	0.671 ± 0.00 $\surd$ $\surd$	0.703 ± 0.02	0.781 ± 0.03
	0.4	0.563 ± 0.03 $\surd$ $\surd$	0.637 ± 0.00 $\surd$ $\surd$	0.681 ± 0.00 $\surd$	0.643 ± 0.00 $\surd$ $\surd$	0.695 ± 0.02	0.742 ± 0.02
Skin	0.2	0.791 ± 0.05 $\surd$ $\surd$	0.833 ± 0.02 $\surd$ $\surd$	0.843 ± 0.01 $\surd$ $\surd$	0.864 ± 0.01 $\surd$ $\surd$	0.886 ± 0.02	0.950 ± 0.01
	0.3	0.789 ± 0.03 $\surd$ $\surd$	0.826 ± 0.02 $\surd$ $\surd$	0.838 ± 0.00 $\surd$ $\surd$	0.859 ± 0.02 $\surd$ $\surd$	0.875 ± 0.02	0.946 ± 0.01
	0.4	0.789 ± 0.02 $\surd$ $\surd$	0.816 ± 0.02 $\surd$ $\surd$	0.818 ± 0.00 $\surd$ $\surd$	0.859 ± 0.01 $\surd$	0.864 ± 0.03	0.939 ± 0.02

TABLE 3: Comparison of test accuracies of various approaches on three real-world datasets including *USPS*, *HockeyFight* and *NBA*. The best two results on each dataset are indicated in red and blue, respectively.

Dataset	(n, d)	η	WSVM [7]	uPU [9]	nnPU [29]	RP [47]	LDCE	KLDCE
USPS	(9298, 256)	0.2	0.925	0.931	0.921	0.931	0.934	0.971
		0.3	0.743	0.907	0.902	0.908	0.911	0.925
		0.4	0.812	0.897	0.892	0.892	0.901	0.919
HockeyFight	(1000, 100)	0.2	0.841	0.847	0.851	0.858	0.860	0.881
		0.3	0.780	0.789	0.782	0.788	0.791	0.815
		0.4	0.681	0.748	0.745	0.750	0.752	0.785
NBA	(1340, 20)	0.2	0.621	0.672	0.672	0.670	0.724	0.681
		0.3	0.721	0.714	0.706	0.688	0.729	0.725
		0.4	0.654	0.686	0.677	0.687	0.714	0.690

Point Made", and so on. The label is 1 if career length of the corresponding player is longer than 5 years, otherwise the label is -1. Similar to previous experiments, here we also transform this problem to PU problem with $\eta = \{0.2, 0.3, 0.4\}$. The training set is made up of 80% examples of the entire "*NBA*" dataset while the rest examples are used for testing.

The results of the test accuracies for various PU methods are described in Table 3. We see that our two methods LDCE and KLDCE achieve comparable performance on this dataset, and they are much better than other state-of-the-art PU methods no matter how many positive examples are integrated to the unlabeled set.

8.4 Parametric Sensitivity

In this subsection, we will discuss the parametric sensitivity of our model in detail. Here we only present the results of KLDCE, as LDCE has similar behavior with KLDCE. Specifically, three real-world datasets *USPS*, *HockeyFight* and *NBA* are adopted for our study. In our KLDCE model (*i.e.* Eq. (23)), there are two trade-off parameters β and λ that should be pre-tuned manually. In our experiments, we examine the test accuracy by varying one of β and λ, and meanwhile fixing the other one to a constant value. In the experiments, the parameter β is changed from 0.1 to 0.9 and the parameter λ ranges from 2^{-4} to 2^4. The results on three real-world datasets with η ranging from 0.2 to 0.4 are shown in Figure 4. From this figure, we observe that the modification of these two parameters will not heavily influence the output of our method, therefore they can be easily tuned for practical implementations.

8.5 The Necessity of Constraint (14)

In Section 5, we theoretically explained that the imposed constraint (14) is critical for our method to accurately estimate the centroid of unlabeled set μ. In this part, we empirically justify the importance of (14) by comparing the performances of our method (including LDCE and KLDCE) when the constraint (14)

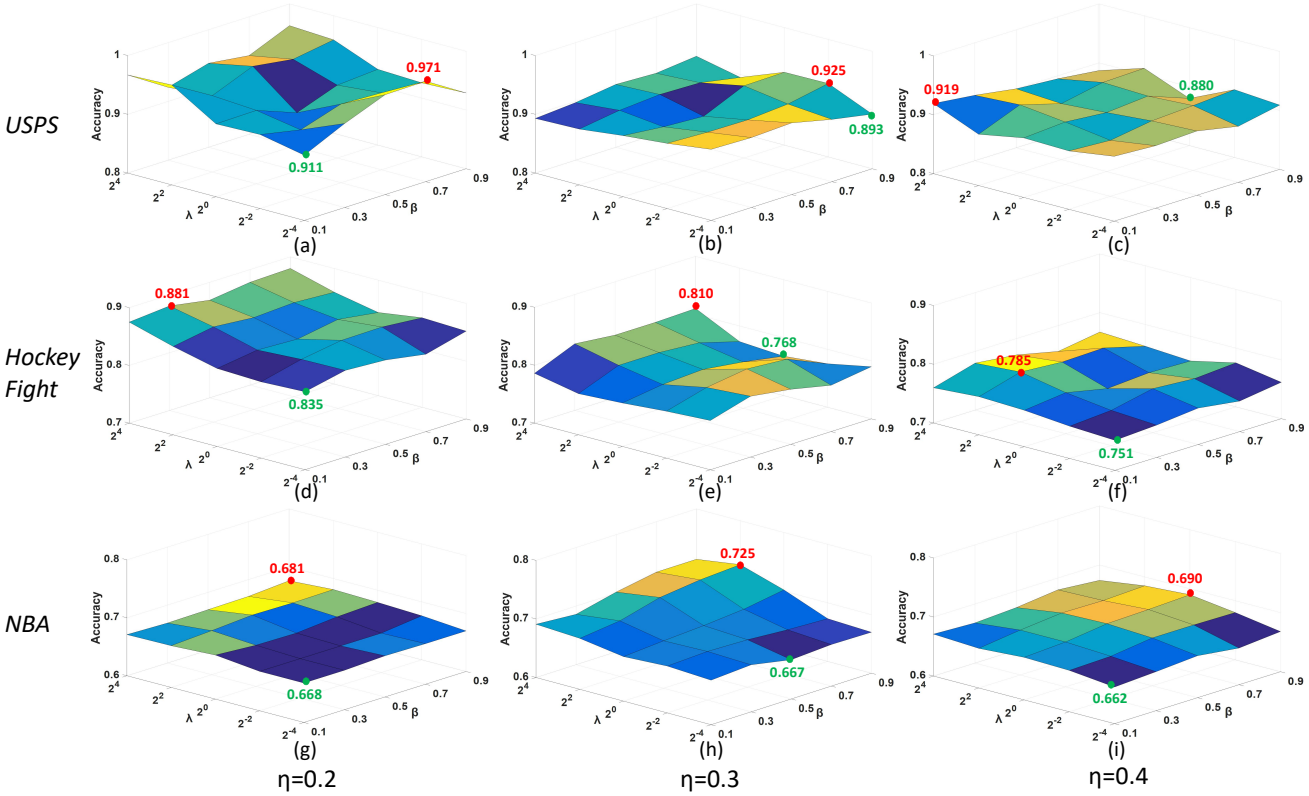

Fig. 4: The parametric sensitivity of β and λ of our KLDCE method when the false negative rate η changes from 0.2 to 0.4. (a)-(c), (d)-(f) and (g)-(i) present the results on *USPS*, *HockeyFight* and *NBA* datasets, respectively. The highest accuracy and lowest accuracy in each subfigure are indicated by red number and green number accordingly.

TABLE 4: Comparison of test accuracies generated by LDCE and KLDCE when the constraint (14) is present and absent. The abbreviation "con." is short for "constraint".

Dataset	η	LDCE (no con.)	LDCE	KLDCE (no con.)	KLDCE
USPS	0.2	0.908	0.934	0.917	0.971
	0.3	0.831	0.911	0.876	0.925
	0.4	0.825	0.901	0.841	0.919
HockeyFight	0.2	0.753	0.860	0.823	0.881
	0.3	0.721	0.791	0.759	0.815
	0.4	0.698	0.752	0.735	0.785
NBA	0.2	0.647	0.724	0.673	0.681
	0.3	0.632	0.729	0.651	0.725
	0.4	0.621	0.714	0.621	0.690

is present and absent. The above three real-world datasets *USPS*, *HockeyFight* and *NBA* are employed, and the experimental results are illustrated in Table 4. It clearly shows that the performance of either LDCE or KLDCE drops dramatically when the constraint is removed, therefore the importance of imposing this constraint for learning a suitable μ is validated.

8.6 The Impact of Inexact η

As mentioned previously, the flipping probability η defined in Eq. (1) might be unknown under practical applications, so it should be estimated via some off-the-shelf methodologies. However, such estimation can be inaccurate, therefore here we investigate how the classification accuracy of our method is influenced by the inaccurate $\hat{\eta}$. The three real-world datasets *USPS*, *HockeyFight* and *NBA* with the actual value $\eta = 0.3$ are employed, and we observe the test accuracies of our method (including LDCE and KLDCE) as well as other compared approaches under the grid $\hat{\eta} = \{0.6\eta, 0.8\eta, \eta, 1.2\eta, 1.4\eta\}$. WSVM is not tested here as this method does not need to estimate the parameter η. The results are presented in Figure 5, which suggests that all approaches are influenced by the inaccurate estimate of η. However, their performances will not severely decrease when the estimated $\hat{\eta}$ is slightly deviated from the real value η.

9 Conclusion

In this paper, we propose a novel PU learning algorithm dubbed "Loss Decomposition and Centroid Estimation" (LDCE), of which the target is to accurately train a binary classifier by only leveraging positive data and unlabeled data. Firstly, we treat all unlabeled data as negative with false negative labeling error and convert PU Learning to the learning problem with noisy labels. Secondly, we utilize the techniques of loss decomposition and centroid estimation to handle this problem. Based on loss decomposition, we shed light on that the unbiased estimate of unlabeled data centroid helps to reduce the adverse effect of noise. Furthermore, we extend the basic linear LDCE algorithm to the kernerlized version to deal with non-linear cases, and the induced algorithm is named "Kernerlized LDCE" (KLDCE). We theoretically show that KLDCE has a good generalization ability. Finally, the experimental results on both synthetic and practical datasets reveal that the proposed methods (LDCE and KLDCE) are more effective than the state-of-the-art PU learning methods.

Since our KLDCE involves the computation of kernel which is usually slow, in the future we may design an acceleration

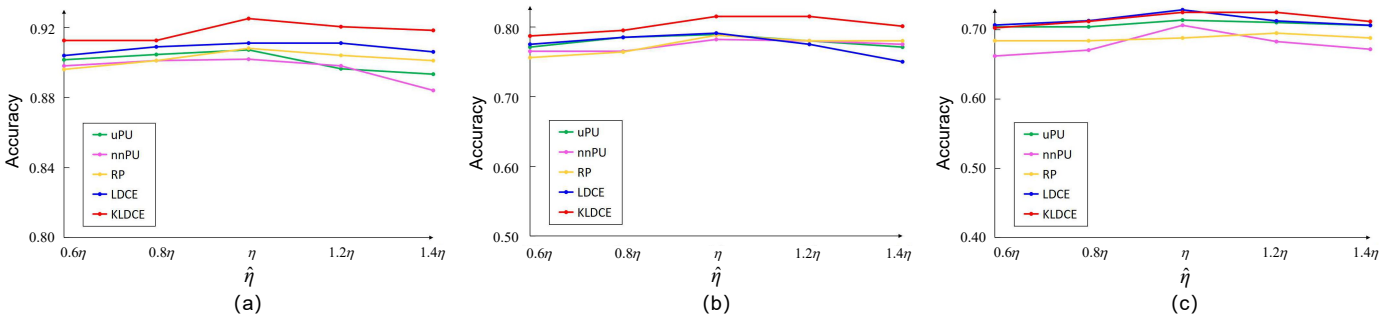

Fig. 5: The performances of uPU, nnPU, RP, LDCE and KLDCE under $\hat{\eta} = \{0.6\eta, 0.8\eta, \eta, 1.2\eta, 1.4\eta\}$ where the actual value of η is 0.3. The subfigures (a), (b) and (c) present the results on *USPS*, *HockeyFight* and *NBA* datasets, respectively.

strategy like [50] to make our method scalable to large-scale datasets. Besides, considering that the positive data in S_P are often not uniformly sampled from the positive distribution in practical situations, we may further investigate the case of biased positive data sampling where only a fraction of definite positive examples are initially annotated.

REFERENCES

[1] C.-J. Hsieh, N. Natarajan, and I. S. Dhillon, "PU learning for matrix completion," in *International Conference on Machine Learning*, 2015, pp. 2445–2453.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] X. Yu, T. Liu, M. Gong, K. Zhang, and D. Tao, "Transfer learning with label noise," *arXiv preprint arXiv:1707.09724*, 2017.

[12] J. Cheng, T. Liu, K. Ramamohanarao, and D. Tao, "Learning with bounded instance-and label-dependent label noise," *arXiv preprint arXiv:1709.03768*, 2017.

[13] 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.

[14] 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.

[15] 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.

[16] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Advances in Neural Information Processing Systems*, 2013, pp. 1196–1204.

[17] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, and X. Gao, "Learning from weak and noisy labels for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 486–500, 2017.

[18] G. Patrini, F. Nielsen, R. Nock, and M. Carioni, "Loss factorization, weakly supervised learning and label noise robustness," in *International Conference on Machine Learning*, 2016, pp. 708–717.

[19] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.

[20] 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.

[21] X. Xu, W. Li, D. Xu, and I. W. Tsang, "Co-labeling for multi-view weakly labeled learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1113–1125, 2016.

[22] E. Sangineto, M. Nabi, D. Culibrk, and N. Sebe, "Self-paced deep learning for weakly supervised object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 3, pp. 712–725, 2019.

[23] F. Denis, "PAC learning from positive statistical queries," in *International Conference on Algorithmic Learning Theory*, 1998, pp. 112–126.

[24] F. De Comité, F. Denis, R. Gilleron, and F. Letouzey, "Positive and unlabeled examples help learning," in *International Conference on Algorithmic Learning Theory*, 1999, pp. 219–230.

[25] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Learning to classify text from labeled and unlabeled documents," in *AAAI Conference on Artificial Intelligence*, 1998, pp. 792–799.

[26] J. Bekker and J. Davis, "Learning from positive and unlabeled data: A survey," *arXiv preprint arXiv:1811.04820*, 2018.

[27] G. Li, "A survey on postive and unlabelled learning," 2013.

[28] 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.

[29] 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.

[30] D. Zhang and W. S. Lee, "A simple probabilistic approach to learning from positive and unlabeled examples," in *Proceedings of the 5th Annual UK Workshop on Computational Intelligence*, 2005, pp. 83–87.

[31] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization," Tech. Rep., 1996.

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

[33] H. Yu, J. Han, and K. C.-C. Chang, "PEBL: positive example based learning for web page classification using SVM," in *Proceedings of The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2002, pp. 239–248.

[34] 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.

[35] 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.

[36] 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. 1–7.

[37] 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.

[38] C. Scott, G. Blanchard, and G. Handy, "Classification with asymmetric label noise: Consistency and maximal denoising," in *Annual Conference on Learning Theory*, 2013, pp. 489–511.

[39] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 447–461, 2016.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] D. Hsu and S. Sabato, "Heavy-tailed regression with a generalized median-of-means," in *International Conference on Machine Learning*, 2014, pp. 37–45.

[45] 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.

[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] C. G. Northcutt, T. Wu, and I. L. Chuang, "Learning with confident examples: Rank pruning for robust classification with noisy labels," *arXiv preprint arXiv:1705.01936*, 2017.

[48] C. J. Merz and P. M. Murphy, "UCI Repository of machine learning databases," 1998.

[49] 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.

[50] 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.

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 60 technical papers at prominent journals and conferences such as IEEE T-PAMI, IEEE T-NNLS, IEEE T-IP, ACM T-IST, NeurIPS, CVPR, AAAI, IJCAI, ICDM, etc. He received the "Excellent Doctorial Dissertation" awarded by Chinese Association for Artificial Intelligence. He was also enrolled by the "Young Elite Scientists Sponsorship Program" by CAST.

Hong Shi received the B.E. degree in compute science and technology from Liaoning University, Shenyang, China. She is currently pursuing the master degree with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. Her current research interests include machine learning and learning-based vision problems.

Tongliang Liu is currently a Lecturer with the School of Computer Science and the Faculty of Engineering, and a core member in the UBTECH Sydney AI Centre, at The University of Sydney. His research interests include machine learning, computer vision, and data mining. He has authored and co-authored 60+ research papers including IEEE T-PAMI, T-NNLS, T-IP, ICML, NeurIPS, AAAI, IJCAI, CVPR, ECCV, KDD, and ICME, with best paper awards, e.g. the 2019 ICME Best Paper Award. He is a recipient of DECRA from Australian Research Council.

Chuang Zhang received the B.E. degree in software engineering from Anhui university, Hefei, China. He is currently pursuing the master degree with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interest is weakly supervised learning.

Jian Yang received the PhD degree from Nanjing University of Science and Technology (NUST) in 2002. Now, he is a Chang-Jiang professor in NUST. He is the author of more than 200 scientific papers in pattern recognition and computer vision. His papers have been cited more than 17000 times in the Google Scholar. Currently, he is/was an associate editor of Pattern Recognition, Pattern Recognition Letters, IEEE Trans. Neural Networks and Learning Systems, and Neurocomputing. His research interests include pattern recognition, computer vision and machine learning. He is a Fellow of IAPR.

Dacheng Tao (F'15) is Professor of Computer Science and ARC Laureate Fellow in the School of Computer Science and the Faculty of Engineering, and the Inaugural Director of the UBTECH Sydney Artificial Intelligence Centre, at The University of Sydney. His research results in artificial intelligence have expounded in one monograph and 200+ publications at prestigious journals and prominent conferences, such as IEEE T-PAMI, IJCV, JMLR, AAAI, IJCAI, NIPS, ICML, CVPR, ICCV, ECCV, ICDM, and KDD, with several best paper awards. He received the 2018 IEEE ICDM Research Contributions Award and the 2015 Australian Scopus-Eureka prize. He is a Fellow of the IEEE and the Australian Academy of Science.