Positive and Unlabeled Learning via Loss Decomposition and Centroid Estimation

Hong Shi, Shaojun Pan, Jian Yang and Chen Gong
School of Computer Science and Engineering, Nanjing University of Science and Technology
Jiangsu Key Laboratory of Image and Video Understanding for Social Security
chen.gong@njust.edu.cn

Abstract

Positive and Unlabeled learning (PU learning) aims to train a binary classifier based on only positive and unlabeled examples, where the unlabeled examples could be either positive or negative. The state-of-the-art algorithms usually cast PU learning as a cost-sensitive learning problem and impose distinct weights to different training examples via a manual or automatic way. However, such weight adjustment or estimation can be inaccurate and thus often lead to unsatisfactory performance. Therefore, this paper regards all unlabeled examples as negative, which means that some of the original positive data are mistakenly labeled as negative. By doing so, we convert PU learning into the risk minimization problem in the presence of false negative label noise, and propose a novel PU learning algorithm termed "Loss Decomposition and Centroid Estimation" (LDCE). By decomposing the hinge loss function into two parts, we show that only the second part is influenced by label noise, of which the adverse effect can be reduced by estimating the centroid of negative examples. We intensively validate our approach on synthetic dataset, UCI benchmark datasets and real-world datasets, and the experimental results firmly demonstrate the effectiveness of our approach when compared with other state-of-the-art PU learning methodologies.

1 Introduction

Traditional supervised machine learning methods usually assume that the negative training data are readily available, and a classifier can be established on both positive and negative training examples. However, in many cases the negative examples are missing, and manually labelling negative data is far more expensive than directly collecting the unlabeled data. To handle such situations, Positive and Unlabeled learning (PU learning) is proposed, of which the target is to accurately train a binary classifier by using positive data and unlabeled data. Here the unlabeled data might be positive or negative, however their groundtruth labels are unknown to the learning algorithm. Some preliminary researches on PU learning can be dated back to [Denis, 1998; De Comité *et al.*, 1999; Nigam *et al.*, 1998], which have shown that unlabeled data are helpful in building an accurate classifier. Recently, PU learning has attracted a great deal of attention due to its practical value and has been applied to solving various problems, such as information retrieval [Latulippe *et al.*, 2013], outlier detection [Scott and Blanchard, 2009], text classification [Liu *et al.*, 2003], and so on.

According to how the unlabeled data are handled, existing PU learning methods can be attributed to three main categories. The first category follows the two-step strategy which firstly identifies the reliable negative data from unlabeled data, and then invokes an ordinary classifier to perform traditional supervised learning. [Liu *et al.*, 2003; 2002; Li and Liu, 2003] are representative works belonging to this category. However, the identification of negative examples can be inaccurate, which may severely degrade the final model performance and lead to poor classification. Therefore, the methods belonging to the second category directly treat all unlabeled examples as negative and formulate PU learning as a cost-sensitive learning problem. By reweighting the training examples, the inaccurate data distribution carried by the observed training set can be calibrated to the potential correct one and thus the ideal data distribution can be approximated. For example, weighted Logistic regression [Lee and Liu, 2003] and weighted SVM [Elkan and Noto, 2008] adjust the data weights by imposing different regularization parameters on labeled and unlabeled examples. However, manually adjusting the regularization parameters could be rather empirical and is very likely to bring about unsatisfactory performance. To avoid tuning the parameters, several recent works focus on designing various unbiased risk estimators, which achieve the state-of-the-art performance. Specifically, [Du Plessis *et al.*, 2014] develops a non-convex ramp loss to amend the data bias caused by the missing of negative examples. To overcome the defect brought by the non-convexity, a convex unbiased loss is presented in [Du Plessis *et al.*, 2015], of which the key idea is to use a weighted ordinary convex loss function for unlabeled data and a weighted composite convex loss function for positive data. Similar to the second category, the last category also regards the unlabeled example as negative, however, with label noise [Yu *et al.*, 2017a; 2017b; Cheng *et al.*, 2017]. In other words, the potential positive examples in the unlabeled set are mislabeled as negative, and thus PU learning can be transformed into a noisy label learning problem. For example, biased SVM [Liu *et al.*, 2003] deploys two trade-off parameters C_+ and C_- to weight positive errors and negative errors during training, respectively. Nevertheless, this method only utilizes free parameters to roughly control the noise rate and does not build specific model to deal with the label noise, so its performance is heavily dependent on the selection of C_+ and C_-.

Considering that the third category that treats unlabeled set as noisy is more straightforward and more easily to implement than the first two categories, in this paper we formulate PU learning as a noisy label learning problem and propose a novel PU learning algorithm dubbed "Loss Decomposition and Centroid Estimation" (LDCE). To be specific, we take the labels of available positive training data as reliable and view the unlabeled examples as noisy negative data. Different from [Liu *et al.*, 2003], in LDCE we explicitly model the label noise in negative set (*i.e.* the original unlabeled set) and convert PU learning into a risk minimization problem in the presence of negative label noise. First of all, we adopt different loss functions for positive examples and noisy negative examples. Secondly, we decompose the empirical loss on negative data into two parts, where only the second part is affected by the noisy data. Furthermore, according to [Gao *et al.*, 2016], the risk minimization in the presence of label noise can be converted to the estimation of the centroid of the statistic labeled examples. Therefore, our problem is turned into the estimation of the centroid of noisy negative data. Thorough experiments on various synthetic and practical datasets demonstrate that the proposed approach is superior to the existing state-of-art methods on PU learning.

2 Problem Description

Suppose that we have n training examples $S_n = \{(\mathbf{x}_1, y_1),\ldots,(\mathbf{x}_k, y_k),(\mathbf{x}_{k+1}, y_{k+1}),\ldots,(\mathbf{x}_n, y_n)\}$ identically and independently drawn from some distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, where $\mathbf{x}_i$ denotes the input vector in feature space $\mathcal{X}$ and y_i is the corresponding class label in output space $\mathcal{Y} = \{+1, -1\}$. In S_n, the first k examples are positive (denoted as S_P), while the rest are unlabeled examples that might be positive or negative. Then the target of PU learning is to train a binary classifier $h\colon \mathcal{X} \times \mathcal{Y}$ on S_n, such that h can assign accurate label $sgn(h(\mathbf{x}))$ to the unseen test example $\mathbf{x}$.

As mentioned in Section 1, this paper treats all unlabeled examples as negative, in which the real positive examples are deemed as mislabeled. Therefore, we use $\widetilde{S_N} = \{(\mathbf{x}_{k+1}, \widetilde{y_{k+1}}), \ldots, (\mathbf{x}_n, \widetilde{y_n})\}$ to denote the corrupted negative set in which the notation " $\sim$ " means that the corresponding labels $y_i = -1 (i = k+1,\ldots,n)$ might be incorrect. Consequently, the entire training set for model training is formed by $\widetilde{S_n} = S_P \cup \widetilde{S_N}$, and the clean version of $\widetilde{S_N}$ is recorded by S_N, in which all examples are correctly labeled. As a result, PU learning here is converted to the problem of learning under label noise. Note that in the studied case no noisy labels appear in the positive set S_P while they only exist in $\widetilde{S_N}$. Besides, we use η to denote the prior of label noise rate in $\widetilde{S_N}$, which can be estimated via cross-validation [Natarajan *et al.*, 2013] or other advanced methods [Liu and Tao, 2016]. Based on above notations, we have the following fact

$$\Pr[\widetilde{y_i} = -1 | y_i = 1] = \eta, \;\; \Pr[\widetilde{y_i} = 1 | y_i = -1] = 0, \tag{1}$$

where $\Pr[\cdot]$ denotes probability. Given y_i as the true label corresponding to the observed corrupted label $\widetilde{y_i}$, we have

$$\begin{aligned} &\Pr[\widetilde{y_i} = 1] \\ &= \Pr[\widetilde{y_i} = 1 | y_i = 1]\Pr[y_i = 1] + \Pr[\widetilde{y_i} = 1 | y_i = -1]\Pr[y_i = -1] \\ &= \Pr[\widetilde{y_i} = 1 | y_i = 1]\Pr[y_i = 1] + 0 \times \Pr[\widetilde{y_i} = -1] \\ &= \Pr[\widetilde{y_i} = 1 | y_i = 1]\Pr[y_i = 1]. \end{aligned} \tag{2}$$

In (2), $\Pr[\widetilde{y_i} = 1]$ is the prior of observed positive class that equals to k/n, then we can get the prior probability of true positive example as

$$p = \Pr[y_i = 1] = \frac{\Pr[\widetilde{y_i} = 1]}{\Pr[\widetilde{y_i} = 1 | y_i = 1]} = \frac{k}{n(1-\eta)}, \tag{3}$$

which is denoted by p for simplicity. Different from other works that estimate the class prior by cross-validation, the positive class prior $\Pr[y_i = 1]$ in our work can be directly computed by (3).

3 Loss Decomposition

Suppose h is an obtained classifier and the loss function is $\ell\colon \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$, which penalizes the deviation from the predicted value $h(\mathbf{x})$ and the groundtruth label y. In our case, the risk of classifier h (*i.e.* $\mathcal{R}(h, S_n)$) on S_n is composed of two parts, where the first part is the loss on the clean positive examples and the second part is the loss on the corrupted negative examples, namely

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

Note that the first term $\mathcal{R}_P(h, S_P)$ in (4) can be easily computed as all the labels in S_P are correct. However, the real value of second term $\mathcal{R}_N(h, S_N)$ is not accessible as the groundtruth labels of the examples in S_N are unknown. Therefore, what follows is studying how to get the unbiased estimation of the second term.

In this paper, we use hinge loss as the loss ℓ, so according to [Patrini *et al.*, 2016], we can further decompose the hinge loss ℓ on the contaminated negative examples into two parts, which reaches

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

where z is variable and the equation 2 holds due to an arithmetic trick of $\max(a, b) = (a + b)/2 + |b - a|/2$. Since for any z, we have

$$|1 - z| \leqslant |z| + 1, |1 + z| \geqslant |z| - 1. \tag{6}$$

Eq. (5) can be further derived as

$$\ell(z) \leqslant \frac{1}{2}([1 - z]_+ + [1 + z]_+) + \frac{1}{2}(1 - z). \tag{7}$$

In this formulation, the term in the first bracket is an even function that is not affected by noise, and the second term $\frac{1}{2}(1-z)$ is an odd function that is affected by noise. Consequently, only the second term in (7) reflects the impact of label noise which will be further studied in Section 4. According to (7), the upper bound of $\mathcal{R}_N(h, S_N)$ is formulated as

$$\overline{\mathcal{R}}_N(h,S_N)=\frac{1}{n}\sum_{i=k+1}^{n}\frac{1}{2}([1-y_ih(\mathbf{x}_i)]_+ + [1+y_ih(\mathbf{x}_i)]_+) + \frac{1}{2}(1-y_ih(\mathbf{x}_i)). \tag{8}$$

Therefore, the upper bound $\overline{\mathcal{R}}_N$ is employed to replace the original $\mathcal{R}_N$ and it is simply denoted by $\mathcal{R}_N$ in the subsequent explanations with a little abuse of notation.

4 Analysis of Noisy Negative Examples

In this section, we analyze the classification risk on the corrupted negative examples $\widetilde{S_N}$. Section 3 has shown that $\mathcal{R}_N(h, S_N)$ can be divided into two parts, where the first part is label independent but the second part is influenced by the erroneous negative labels. Hence, we investigate the influence of noisy negative examples on the second term. Assume the linear classifier is $h_{\boldsymbol{w}}(\mathbf{x}_i) = \langle \boldsymbol{w}, \mathbf{x}_i\rangle$, where $\boldsymbol{w}$ is the model parameter, then according to (8), we have

$$\begin{aligned}\mathcal{R}_N(h,S_N)=&\frac{1}{n}\sum_{i=k+1}^{n}\frac{1}{2}([1-y_ih(\mathbf{x}_i)]_+ + [1+y_ih(\mathbf{x}_i)]_+)\\ &+\frac{1}{2}(1-y_ih(\mathbf{x}_i))\\ =&\frac{1}{n}\sum_{i=k+1}^{n}\frac{1}{2}([1-y_ih(\mathbf{x}_i)]_+ + [1+y_ih(\mathbf{x}_i)]_+)\\ &+\frac{1}{n}\sum_{i=k+1}^{n}\frac{1}{2}-\frac{1}{2}\langle\boldsymbol{w},\frac{1}{n}\sum_{i=k+1}^{n}y_i\mathbf{x}_i\rangle.\end{aligned} \tag{9}$$

Noise only exists in negative examples, hence the only thing we need to consider carefully is the corrupted negative examples. From the explanation of Eq. (7) in Section 3, we know that we only need to focus on the third term of Eq. (9) for dealing with label noise. Moreover, we also introduce the notion of negative example centroid that concerns the unlabeled examples in S_N with true labels and true distribution $\mathcal{D}$, namely $\mu(S_N) = \frac{1}{n-k}\sum_{i=k+1}^{n} y_i\mathbf{x}_i$ and $\mu(\mathcal{D}) = E_{(\mathbf{x},y)\sim\mathcal{D}[\mathbf{x},y]}$.

Similarly, we also define the unlabeled example centroid $\mu(\widetilde{S_N}) = \frac{1}{n-k}\sum_{i=k+1}^{n}\widetilde{y}_i\mathbf{x}_i$ and $\mu(\mathcal{D}_\eta) = E_{(\mathbf{x},\widetilde{y})\sim\mathcal{D}[\mathbf{x},\widetilde{y}]}$ on the corrupted negative set $\widetilde{S_N}$ and corrupted distribution $\mathcal{D}_\eta$, respectively. By substituting $\mu(S_N)$ into $\mathcal{R}_N(h, S_N)$ and ignoring the constant term, (9) is transformed to

$$\begin{aligned}\mathcal{R}_N(h,S_N)=&\frac{1}{2n}\sum_{i=k+1}^{n}([1-y_i\langle\boldsymbol{w},\mathbf{x}_i\rangle]_+ + [1+y_i\langle\boldsymbol{w},\mathbf{x}_i\rangle]_+)\\ &-\frac{n-k}{2n}\langle\boldsymbol{w},\mu(S_N)\rangle.\end{aligned} \tag{10}$$

Since we can only observe the corrupted $\widetilde{S_N}$ supported by $\mathcal{D}_\eta$ rather than the true S_N supported by $\mathcal{D}$. The key for computing (10) lies in the estimation of $\mu(S_N)$. To this end, we provide the following theorem:

Theorem 1. *Given η as the prior of label noise rate in $\widetilde{S_N}$ and p as the prior probability of true positive example defined in* (3)*, the means of true distribution $\mathcal{D}$ and the corrupted distribution $\mathcal{D}_\eta$ satisfy $\mu(\mathcal{D}_\eta) = (1-2p\eta)\mu(\mathcal{D})$. Similarly, the example centroid of the real S_N and the corrupted negative set $\widetilde{S_N}$ have the relationship $E_{\widetilde{y}_1,\ldots,\widetilde{y}_n}[\mu(\widetilde{S_N})] = (1-2p\eta)\mu(S_N)$.*

Proof. It is straightforward that

$$\begin{aligned}&E_{\widetilde{y}}[\widetilde{y}\mathbf{x}|(\mathbf{x},y)]\\ &= pE_{\widetilde{y}}[\widetilde{y}\mathbf{x}|(\mathbf{x},y)] + (1-p)E_{\widetilde{y}}[\widetilde{y}\mathbf{x}|(\mathbf{x},y)]\\ &= p(1-2\eta)y\mathbf{x} + (1-p)y\mathbf{x}\\ &= (1-2p\eta)y\mathbf{x}.\end{aligned} \tag{11}$$

From $E_{\widetilde{y}}[\mathbf{x}|(\mathbf{x},y)]$, we have

$$\begin{aligned}\mu(\mathcal{D}_\eta) &= E_{(\mathbf{x},\widetilde{y})\sim\mathcal{D}_\eta}[\widetilde{y}\mathbf{x}]\\ &= E_{(\mathbf{x},y)\sim\mathcal{D}}[E_{\widetilde{y}}[\widetilde{y}\mathbf{x}|(\mathbf{x},y)]]\\ &= E_{(\mathbf{x},y)\sim\mathcal{D}}[(1-2p\eta)y\mathbf{x}]\\ &= (1-2p\eta)\mu(\mathcal{D}),\end{aligned} \tag{12}$$

and

$$E[\mu(\widetilde{S_N})] = (1-2p\eta)\mu(S_N). \tag{13}$$

Theorem 1 informs us that $\mu(\widetilde{S_N})/(1-2p\eta)$ is an unbiased estimation of $\mu(S_N)$. Besides, [Gao *et al.*, 2016] shows that the random noise increases the covariance of $y\mathbf{x}$, and may result in heavy-tailed distributions. Hence, we derive the covariance matrix $\Sigma(\mu(\widetilde{S_N}))$ of negative instance centroid $\mu(\widetilde{S_N})$, which is given in the following theorem.

Theorem 2. *Given $\widetilde{S_N}$ as the corrupted negative example, the empirical covariance matrix $\hat{\Sigma}(\mu(\widetilde{S_N}))$ is*

$$\hat{\Sigma}[\mu(\widetilde{S_N})]=\sum_{i=k+1}^{n}\frac{\mathbf{x}_i^\top\mathbf{x}_i}{n-k}-\frac{1}{n-k}\sum_{i=k+1}^{n}\frac{\mathbf{x}_i^\top\widetilde{y}_i}{n-k}\sum_{i=k+1}^{n}\frac{\mathbf{x}_i\widetilde{y}_i}{n-k}. \tag{14}$$

Proof. The definition of covariance matrix is

$$\Sigma(\mu(\widetilde{S_N}))=E[[\mu(\widetilde{S_N})]^\top\mu(\widetilde{S_N})]-[E[\mu(\widetilde{S_N})]]^\top E[\mu(\widetilde{S_N})]. \tag{15}$$

Besides, since

$$\begin{aligned}&E[[\mu(\widetilde{S_N})]^\top\mu(\widetilde{S_N})]\\ &=E[[\frac{1}{n-k}\sum_{i=k+1}^{n}\widetilde{y}_i\mathbf{x}_i]^\top\frac{1}{n-k}\sum_{i=k+1}^{n}\widetilde{y}_i\mathbf{x}_i]\\ &=\frac{1}{(n-k)^2}(\sum_{i=k+1}^{n}E[\mathbf{x}_i^\top\mathbf{x}_i]+\sum_{i\neq j}E[\widetilde{y}_i\widetilde{y}_j\mathbf{x}_i^\top\mathbf{x}_j]),\end{aligned} \tag{16}$$

and

$$E[\widetilde{y}_i\widetilde{y}_j\mathbf{x}_i^\top\mathbf{x}_j]=E[\sum_{i=k+1}^{n}\frac{\mathbf{x}_i\widetilde{y}_i}{n-k}]^\top E[\sum_{i=k+1}^{n}\frac{\mathbf{x}_i\widetilde{y}_i}{n-k}]. \tag{17}$$

We may easily get the covariance matrix by substituting (16) and (17) into (15). As a good approximation of (15), we can define the empirical covariance as (14) [Gao *et al.*, 2016].

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

```
Input: The corrupted negative sample, $\widetilde{S_N}$; the number of groups $g$, $g \ge 1$;
Output: The median-of-means estimator, $\hat{\mu}(\widetilde{S_N})$;
Randomly divide $\widetilde{S_N}$ into $g$ groups $\{\widetilde{S_N}^{[1]}, \widetilde{S_N}^{[2]}, \dots, \widetilde{S_N}^{[g]}\}$ with almost equal size;
Calculate the standard empirical mean $\mu(\widetilde{S_N}^{[i]})$ for each $i \in [g]$ and each group $\widetilde{S_N}^{[i]}$;
Calculate $r_i = \mathrm{median}\{\mu(\widetilde{S_N}^{[i]}) - \mu(\widetilde{S_N}^{[j]})\}$ for each $i \in [g]$, and then set $i_* = \arg\min_{i\in[g]} r_i$;
return $\hat{\mu}(\widetilde{S_N})$.
```

5 The LDCE Algorithm

Section 4 shows that we can estimate the true unlabeled example centroid $\mu(S_N)$, which is significant to handle PU problem, by estimating the corrupted negative example centroid $\mu(\widetilde{S_N})$. In this paper, we adopt the recent generalized median-of-means estimator [Hsu and Sabato, 2014] to estimate $\mu(\widetilde{S_N})$. The basic idea is to randomly divide the corrupted negative set $\widetilde{S_N}$ into g groups with almost equal size, and then return the generalized median of sample means for each group under ℓ_2-norm metric. Algorithm 1 presents the detailed description about this process.

Due to the influence of noise on the covariance of $y\mathbf{x}$, that has been discussed in Section 4, here we impose a constraint on $\mu(\widetilde{S_N})$, which is

$$(\mu - \hat{\mu}(\widetilde{S_N}))^\top \hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))(\mu - \hat{\mu}(\widetilde{S_N})) \le \beta, \tag{18}$$

where $\hat{\mu}(\widetilde{S_N})$ is the output of Algorithm 1, $\hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))$ is shown in Eq. (15), and β can be estimated by cross-validation. Therefore, our PU learning model is formalized as

$$\begin{aligned}\min_{\boldsymbol{w},\mu}\quad & \frac{1}{n}\sum_{i=1}^{k}\ell(y_i\langle\boldsymbol{w},\mathbf{x}_i\rangle) + \frac{1}{2n}\sum_{i=k+1}^{n}\varphi(y_i,\langle\boldsymbol{w},\mathbf{x}_i\rangle)\\ & + \frac{c}{1-2p\eta}\langle\boldsymbol{w},\mu\rangle + \lambda\|\boldsymbol{w}\|^2\\ s.t.\quad & (\mu - \hat{\mu}(\widetilde{S_N}))^\top \hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))(\mu - \hat{\mu}(\widetilde{S_N})) \le \beta,\end{aligned} \tag{19}$$

where $\ell(y_i\langle\boldsymbol{w},\mathbf{x}_i\rangle) = \max(0, 1 - y_i\langle\boldsymbol{w},\mathbf{x}_i\rangle)$, $c = -(n-k)/2n$, $\varphi(y_i,\langle\boldsymbol{w},\mathbf{x}_i\rangle) = [1-y_i\langle\boldsymbol{w},\mathbf{x}_i\rangle]_+ + [1+y_i\langle\boldsymbol{w},\mathbf{x}_i\rangle]_+$, $p = k/n(1-\eta)$, and $\lambda\|\boldsymbol{w}\|^2$ is the regularization term to avoid overfitting.

In this work, we use the Alternative Convex Search method to solve the optimization problem (19). Specially, after fixing μ, we can simply use the gradient descent algorithm to solve the minimization problem on $\boldsymbol{w}$, which is

$$\begin{aligned}\min_{\boldsymbol{w}}\quad & \frac{1}{n}\sum_{i=1}^{k}\ell(y_i\langle\boldsymbol{w},\mathbf{x}_i\rangle) + \frac{1}{2n}\sum_{i=k+1}^{n}\varphi(y_i,\langle\boldsymbol{w},\mathbf{x}_i\rangle)\\ & + \frac{c}{1-2p\eta}\langle\boldsymbol{w},\mu\rangle + \lambda\|\boldsymbol{w}\|^2.\end{aligned} \tag{20}$$

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

```
Input: The corrupted sample $\widetilde{S}_n = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_k, y_k), (\mathbf{x}_{k+1}, \widetilde{y}_{k+1}), \dots, (\mathbf{x}_n, \widetilde{y}_n)\}$, the noisy parameter $\eta$, the regularization parameter $\lambda$, the approximation $\beta$;
Output: The optimal classifier parameter $\boldsymbol{w}$;
Call Algorithm 1 to give an estimation of $\hat{\mu} = \hat{\mu}(\widetilde{S_N})$;
Calculate $\hat{\Sigma} = \hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))$ by Eq. (14);
Initialize $t = 0$ and $\boldsymbol{w}_0$;
repeat
    Calculate $\mu = \hat{\mu} + \hat{\Sigma}^{-1}\boldsymbol{w}\sqrt{\beta/(\boldsymbol{w}^\top\hat{\Sigma}^{-1}\boldsymbol{w})}$;
    Use gradient descent method to solve
        $\boldsymbol{w}_t = \arg\min_{\boldsymbol{w}} \frac{1}{n}\sum_{i=1}^{k}\ell(y_i\langle\boldsymbol{w},\mathbf{x}_i\rangle) + \frac{1}{2n}\sum_{i=k+1}^{n}\varphi(y_i,\langle\boldsymbol{w},\mathbf{x}_i\rangle)$
        $+ \frac{c}{1-2p\eta}\langle\boldsymbol{w},\mu\rangle + \lambda\|\boldsymbol{w}\|^2$;
    $t = t + 1$;
until convergence;
return The converged $\boldsymbol{w}$.
```

For fixed $\boldsymbol{w}$, after ignoring some constant terms, the optimization problem regarding μ is

$$\begin{aligned}\min_{\mu}\quad & c\langle\boldsymbol{w},\mu\rangle\\ s.t.\quad & (\mu - \hat{\mu}(\widetilde{S_N}))^\top \hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))(\mu - \hat{\mu}(\widetilde{S_N})) \le \beta.\end{aligned} \tag{21}$$

To deal with this constrained optimization problem, we can introduce a Lagrange variable ρ, and thus

$$\begin{aligned}L(\mu,\beta) = {} & c\langle\boldsymbol{w},\mu\rangle - \rho(\mu - \hat{\mu}(\widetilde{S_N}))^\top \hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))(\mu - \hat{\mu}(\widetilde{S_N}))\\ & + \rho\beta.\end{aligned} \tag{22}$$

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

$$\mu = \frac{c}{2\rho}(\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1}\boldsymbol{w} + \hat{\mu}(\widetilde{S_N}). \tag{23}$$

By plugging Eq. (23) into Eq. (21), we have

$$\begin{aligned}\min_{\rho}\quad & \frac{c}{2\rho}\boldsymbol{w}^\top(\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1}\boldsymbol{w}\\ s.t.\quad & \frac{c^2}{4\rho^2}\boldsymbol{w}^\top(\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1}\boldsymbol{w} \le \beta,\end{aligned} \tag{24}$$

of which the 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. (23), we derive the solution of Eq. (23) as

$$\mu = \hat{\mu}(\widetilde{S_N}) + (\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1}\boldsymbol{w}\sqrt{\beta/(\boldsymbol{w}^\top(\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1}\boldsymbol{w})}. \tag{25}$$

Algorithm 2 shows the detailed procedure of our algorithm.

6 Experiments

In this section, we perform exhaustive experiments on one synthetic dataset, seven publicly available benchmark datasets and two real-world datasets. We compare our proposed method (dubbed as "LDCE") with state-of-the-art PU learning methods such as Weighted SVM (W-SVM) [Elkan and Noto, 2008], Unbiased PU (UPU) [Du Plessis *et al.*, 2015] and Non-Negative PU (NNPU) [Kiryo *et al.*, 2017]. Besides, LDCE is also compared with the traditional Linear SVM (L-SVM) for which the unlabeled examples are naively treated as negative.

6.1 Synthetic Dataset

Firstly, we create a two-dimensional dataset consisted of two Gaussians as shown in Figure 1 (a). The dataset contains 500 positive examples and 500 negative examples, and each class corresponds to a Gaussian. We make 20% ($\eta = 0.2$) of the original positive examples and all negative examples to be unlabeled (see Figure 1 (b)), and examine whether different PU learning methods can accurately find the proper decision boundary for separating positive and negative examples. The results of various compared methods are shown in Figure 1 (c)$\sim$(f), revealing that only our LDCE achieves 100% classification accuracy, which is higher than 99.8%, 99.7% and 94% obtained by W-SVM, UPU and L-SVM, respectively. Note that NNPU is not compared as this method does not output explicit decision function. Specifically, we see that many negative examples are classified as positive by the traditional L-SVM, so directly treating all unlabeled examples as negative is inappropriate for PU learning. Besides, although the performances of W-SVM and UPU are better than L-SVM, they fail to distinguish the ambiguous examples near the potentially correct decision boundary. Therefore, the superiority of LDCE to other existing models are demonstrated.

6.2 UCI Benchmark Dataset

To demonstrate the effectiveness of our proposed method, we also conducted extensive experiments on seven datasets from UCI machine learning repository [Merz and Murphy, 1998]. The size of training set n and the feature dimensionality d for each dataset is presented in Table 1. All data features are normalized to $[-1, 1]$ in advance. For each of the dataset illustrated in Table 1, we randomly pick 80% of the data for training and the rest 20% examples are used for testing. Then, we randomly select 20%, 30%, 40% (*i.e.* $\eta \in \{0.2, 0.3, 0.4\}$) positive training examples and combine them with original negative set to compose the unlabeled set. Note that the division of positive set and unlabeled set under each η is kept identical for every compared method. In our experiment, we conduct 5-fold cross validation on all comparators and their mean test accuracies over the five trials are reported in Table 1. Furthermore, we also apply the t-test with significant level 0.1 to statistically examine whether our LDCE is significantly better than other methods.

From the mean test accuracies reported in Table 1, we see that LDCE is consistently among the best two methods on the seven datasets. Apart from *mushroom* dataset, the performances of existing PU learning approaches can be significantly enhanced by LDCE as revealed by the t-test.

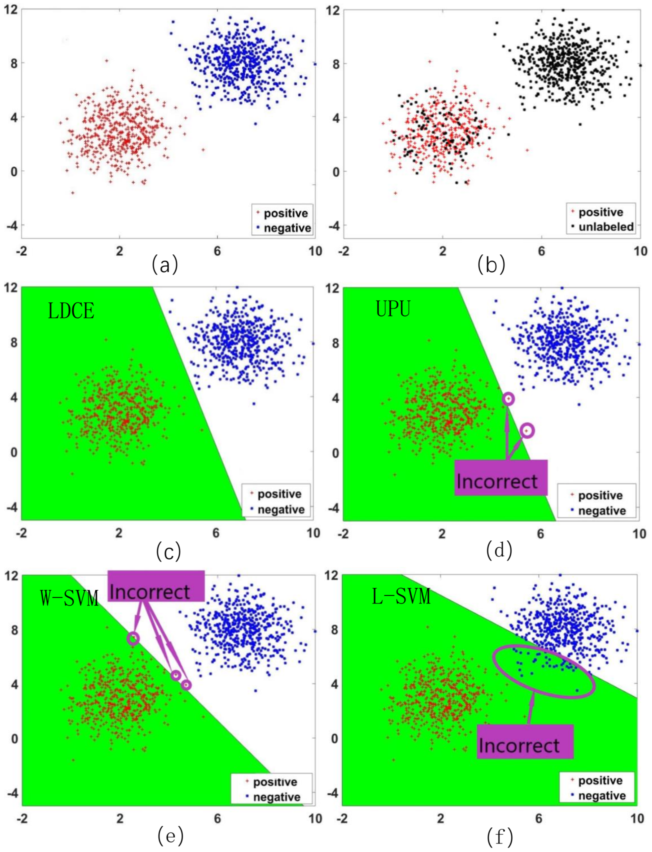

Figure 1: The performances of various methods on synthetic dataset: (a) shows the real positive and unlabeled examples, (b) shows the positive and unlabeled examples for model training, (c)$\sim$(f) display the decision boundaries generated by LDCE, UPU, W-SVM and L-SVM, respectively. The incorrectly classified examples are highlighted by purple circles.

6.3 Real-world Data

We also conduct the experiments on two real-world datasets to evaluate the ability of our LDCE in dealing with practical problem.

Handwritten Digit Recognition

The *USPS* [1] dataset was adopted to assess the ability of various methods in recognizing the handwritten digits. This dataset contains 9298 digit images belonging to 10 classes, *i.e.* the digits "0"-"9". The resolution of all images is 16×16, so the pixel-wise feature we adopted was 256 dimensions, in which every dimension represents the gray value of corresponding pixel. We choose the digit images of "0" as positive, and regard the rest of digit images as negative examples. Therefore, there are 1553 positive examples and 7745 negative examples, and such class imbalance will pose a great challenge for the compared methodologies. The way for generating the positive set is the same as the manipulations in Section 6.2.

Since this dataset is imbalanced, we apply two metrics,

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

Dataset	(n, d)	Num of P	Num of N	η	W-SVM	UPU	NNPU	L-SVM	LDCE
vote	(435, 16)	267	168	0.2	0.876 ± 0.04 $\surd$	0.888 ± 0.03	0.883 ± 0.03 $\surd$	0.775 ± 0.05 $\surd$	0.901 ± 0.03
				0.3	0.862 ± 0.05	0.885 ± 0.04	0.869 ± 0.02	0.834 ± 0.03 $\surd$	0.894 ± 0.04
				0.4	0.801 ± 0.09	0.802 ± 0.03 $\surd$	0.786 ± 0.03 $\surd$	0.747 ± 0.03 $\surd$	0.823 ± 0.05
balance	(625, 4)	288	337	0.2	0.853 ± 0.03 $\surd$	0.854 ± 0.03 $\surd$	0.838 ± 0.03 $\surd$	0.822 ± 0.04 $\surd$	0.910 ± 0.02
				0.3	0.773 ± 0.12	0.805 ± 0.03 $\surd$	0.760 ± 0.11	0.773 ± 0.12	0.827 ± 0.03
				0.4	0.629 ± 0.05 $\surd$	0.685 ± 0.02 $\surd$	0.680 ± 0.04 $\surd$	0.626 ± 0.05 $\surd$	0.721 ± 0.03
breast	(683, 10)	143	540	0.2	0.947 ± 0.01 $\surd$	0.952 ± 0.02 $\surd$	0.956 ± 0.01 $\surd$	0.933 ± 0.02 $\surd$	0.974 ± 0.02
				0.3	0.802 ± 0.02 $\surd$	0.795 ± 0.01 $\surd$	0.821 ± 0.03	0.798 ± 0.01 $\surd$	0.845 ± 0.03
				0.4	0.794 ± 0.01 $\surd$	0.791 ± 0.00 $\surd$	0.799 ± 0.02 $\surd$	0.789 ± 0.02 $\surd$	0.845 ± 0.04
australian	(690, 14)	370	383	0.2	0.810 ± 0.04 $\surd$	0.811 ± 0.04	0.838 ± 0.02	0.796 ± 0.05 $\surd$	0.849 ± 0.03
				0.3	0.828 ± 0.01 $\surd$	0.842 ± 0.01	0.838 ± 0.03 $\surd$	0.825 ± 0.02 $\surd$	0.852 ± 0.02
				0.4	0.832 ± 0.04	0.801 ± 0.02 $\surd$	0.839 ± 0.03 $\surd$	0.809 ± 0.04	0.851 ± 0.02
benknote	(1372, 4)	610	762	0.2	0.966 ± 0.01 $\surd$	0.958 ± 0.01 $\surd$	0.963 ± 0.01 $\surd$	0.864 ± 0.08 $\surd$	0.977 ± 0.01
				0.3	0.953 ± 0.02 $\surd$	0.952 ± 0.01 $\surd$	0.972 ± 0.01	0.775 ± 0.04 $\surd$	0.975 ± 0.02
				0.4	0.937 ± 0.01 $\surd$	0.929 ± 0.03 $\surd$	0.964 ± 0.01	0.808 ± 0.05 $\surd$	0.973 ± 0.01
mushroom	(8124, 112)	3916	4208	0.2	0.658 ± 0.09 $\surd$	0.764 ± 0.02 $\surd$	0.910 ± 0.04 $\times$	0.526 ± 0.05 $\surd$	0.849 ± 0.03
				0.3	0.688 ± 0.09 $\surd$	0.726 ± 0.01 $\surd$	0.864 ± 0.07 $\times$	0.504 ± 0.01 $\surd$	0.750 ± 0.01
				0.4	0.510 ± 0.01 $\surd$	0.656 ± 0.04	0.750 ± 0.05 $\times$	0.507 ± 0.01 $\surd$	0.658 ± 0.05
web	(11055, 31)	6157	4898	0.2	0.794 ± 0.03 $\surd$	0.785 ± 0.04 $\surd$	0.830 ± 0.02	0.630 ± 0.02 $\surd$	0.840 ± 0.02
				0.3	0.691 ± 0.06 $\surd$	0.729 ± 0.05 $\surd$	0.822 ± 0.02	0.652 ± 0.08 $\surd$	0.827 ± 0.02
				0.4	0.625 ± 0.03 $\surd$	0.732 ± 0.07 $\surd$	0.817 ± 0.01	0.612 ± 0.01 $\surd$	0.821 ± 0.01

Table 1: Comparison of mean test accuracies of various approaches on UCI datasets. $\surd$ ($\times$) denotes that our approach is significantly better (worse) than the corresponding method revealed by the paired t-test with significance level 0.1. The best two results on each dataset are indicated in red and blue, respectively.

i.e. test accuracy and F-measure, to evaluate the performance of all methods. The results of various methods are presented in Table 2 and Figure 2, respectively. From the Table 2, we see that our proposed LDCE preforms better than other methods under different noise rates (η), in terms of test accuracy. From Figure 2, we observe that the F-measure obtained by LDCE is also higher than other methods on the two real-world datasets, which demonstrate the effectiveness of our proposed approach in dealing with imbalance data.

Violent Behavior Detection

Recently, intelligent monitoring technique for detecting violent behavior has gained incresing of attention due to its great practical significance. In this section, we utilize the *HockeyFight* [2] dataset and apply our proposed approach and other PU methods to fight behavior detection. The *HockeyFight* dataset is made up of 1000 video clips collected in ice hockey competitions, of which 500 contain fight behavior and 500 are non-fight sequences. We classify the clips with fighting and without fighting by using various PU learning methods including L-SVM, W-SVM, UPU, NNPU and our LDCE. Similar to [Gong *et al.*, 2015], 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. Hence, every clip in the dataset was characterized by a 100-dimensional feature vector. Similar to Section 6.2, the proportions of training examples and test examples are also maintained as 80% and 20%, respectively, and the noise rate η also ranges from 0.2 to 0.4.

The test accuracies and F-measure values achieved by the compared methods are presented in Table 2 and Figure 2, respectively, which clearly indicate that our LDCE achieves the top-level performance among all the comparators. Particularly, it can be noted that the F-measure of LDCE is as high as 0.783 when the noise rate is 0.4. In contrast, the second best L-SVM only achieves the F-measure of 0.671. Therefore, our LDCE is still able to render very impressive results even though a large number of positive examples are "hidden" in the original unlabeled set.

Dataset	(n, d)	η	W-SVM	UPU	NNPU	L-SVM	LDCE
HockeyFight	(1000, 100)	0.2	0.841	0.845	0.845	0.825	**0.860**
		0.3	0.780	0.765	0.745	0.745	**0.791**
		0.4	0.681	0.742	0.735	0.514	**0.752**
USPS	(9298, 256)	0.2	0.925	0.929	0.916	0.896	**0.934**
		0.3	0.743	0.901	0.893	0.733	**0.911**
		0.4	0.812	0.892	0.866	0.768	**0.901**

Table 2: Comparison of test accuracies of various approaches on two real-world datasets including *HockeyFight* and *USPS*. The best result on each dataset is indicated in bold.

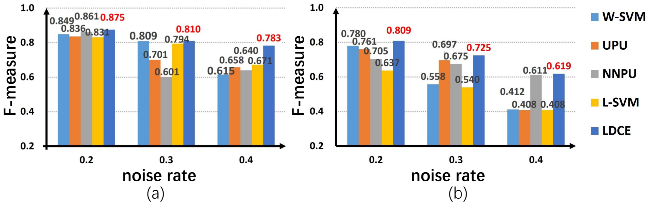

Figure 2: The F-measure of various PU learning methods on (a) *HockeyFight* and (b) *USPS*. The highest F-measure value under each noise rate η is marked by red.

7 Conclusion

This paper proposes a novel PU learning algorithm dubbed "Loss Decomposition and Centroid Estimation" (LDCE). By treating the unlabeled examples as negative with false negative label error, we convert PU learning to the noisy label learning problem, and use the loss decomposition technique to explicitly model the noisy labels. Based on loss decomposition, we shed light on that the unbiased estimation of labeled example centroid helps to reduce the adverse effect of noise. Thorough experimental results on both synthetic and practical datasets show that the proposed method is more effective than the state-of-the-art PU learning methods. In the future, it is worthwhile to extend our LDCE model to non-linear case by introducing the kernel trick.

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

Acknowledgments

This research is supported by NSF of China (No: 61602246, U1713208 and 61472187), NSF of Jiangsu Province (No: BK20171430), the Fundamental Research Funds for the Central Universities (No: 30918011319), the "Summit of the Six Top Talents" Program (No: DZXX-027), the 973 Program No.2014CB349303, and Program for Changjiang Scholars.

References

[Cheng *et al.*, 2017] Jiacheng Cheng, Tongliang Liu, Kotagiri Ramamohanarao, and Dacheng Tao. Learning with bounded instance-and label-dependent label noise. *arXiv preprint arXiv:1709.03768*, 2017.

[De Comité *et al.*, 1999] Francesco De Comité, François Denis, Rémi Gilleron, and Fabien Letouzey. Positive and unlabeled examples help learning. In *International Conference on Algorithmic Learning Theory*, pages 219–230, 1999.

[Denis, 1998] François Denis. PAC learning from positive statistical queries. In *International Conference on Algorithmic Learning Theory*, pages 112–126, 1998.

[Du Plessis *et al.*, 2014] Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *Advances in Neural Information Processing Systems*, pages 703–711, 2014.

[Du Plessis *et al.*, 2015] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *International Conference on Machine Learning*, pages 1386–1394, 2015.

[Elkan and Noto, 2008] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 213–220, 2008.

[Gao *et al.*, 2016] Wei Gao, Lu Wang, Yu-Feng Li, and Zhi-Hua Zhou. Risk minimization in the presence of label noise. In *AAAI Conference on Artificial Intelligence*, pages 1575–1581, 2016.

[Gong *et al.*, 2015] Chen Gong, Tongliang Liu, Dacheng Tao, Keren Fu, Enmei Tu, and Jie Yang. Deformed graph laplacian for semisupervised learning. *IEEE Transactions on Neural Networks and Learning Systems*, 26:2261–2274, 2015.

[Hsu and Sabato, 2014] Daniel Hsu and Sivan Sabato. Heavy-tailed regression with a generalized median-of-means. In *International Conference on Machine Learning*, pages 37–45, 2014.

[Kiryo *et al.*, 2017] Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *Advances in Neural Information Processing Systems*, pages 1674–1684, 2017.

[Latulippe *et al.*, 2013] Maxime Latulippe, Alexandre Drouin, Philippe Giguere, and François Laviolette. Accelerated robust point cloud registration in natural environments through positive and unlabeled learning. In *International Joint Conference on Artificial Intelligence*, pages 2480–2487, 2013.

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

[Li and Liu, 2003] Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *International Joint Conference on Artificial Intelligence*, volume 3, pages 587–592, 2003.

[Liu and Tao, 2016] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, pages 447–461, 2016.

[Liu *et al.*, 2002] Bing Liu, Wee Sun Lee, Philip S Yu, and Xiaoli Li. Partially supervised classification of text documents. In *International Conference on Machine Learning*, pages 387–394, 2002.

[Liu *et al.*, 2003] Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S Yu. Building text classifiers using positive and unlabeled examples. In *The IEEE International Conference on Data Mining*, volume 2, pages 179–186, 2003.

[Merz and Murphy, 1998] Christopher J Merz and Patrick M Murphy. UCI Repository of machine learning databases. 1998.

[Natarajan *et al.*, 2013] Nagarajan Natarajan, Inderjit S D-hillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, pages 1196–1204, 2013.

[Nigam *et al.*, 1998] Kamal Nigam, Andrew McCallum, Sebastian Thrun, Tom Mitchell, et al. Learning to classify text from labeled and unlabeled documents. In *AAAI Conference on Artificial Intelligence*, volume 792, 1998.

[Patrini *et al.*, 2016] Giorgio Patrini, Frank Nielsen, Richard Nock, and Marcello Carioni. Loss factorization, weakly supervised learning and label noise robustness. In *International Conference on Machine Learning*, pages 708–717, 2016.

[Scott and Blanchard, 2009] Clayton Scott and Gilles Blanchard. Novelty detection: Unlabeled data definitely help. In *Artificial Intelligence and Statistics*, pages 464–471, 2009.

[Yu *et al.*, 2017a] Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. *arXiv preprint arXiv:1711.09535*, 2017.

[Yu *et al.*, 2017b] Xiyu Yu, Tongliang Liu, Mingming Gong, Kun Zhang, and Dacheng Tao. Transfer learning with label noise. *arXiv preprint arXiv:1707.09724*, 2017.