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 learn-
ing problem. For example, biased SVM [Liu et al., 2003] deploy 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 statistical labeled examples. Therefore, our problem is turned into the estimation of the centroid of noisy negative data. Through 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 \( S_n = \{(x_1, y_1), ..., (x_k, y_k), (x_{k+1}, y_{k+1}), ..., (x_n, y_n)\} \) identically and independently drawn from some distribution \( D \) over \( \mathcal{X} \times \mathcal{Y} \), where \( x_i \) 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 learn a binary classifier \( h: \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{S}_n \), such that \( h \) can assign accurate label \( sign(h(x)) \) to the unseen test example \( x \).

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 \( \overline{S_N} = \{(x_{k+1}, y_{k+1}), ..., (x_n, y_n)\} \) to denote the corrupted negative set in which the notation “\( \sim \)” means that the corresponding labels \( y_i = -1 (i = k + 1, ..., n) \) might be incorrect. Consequently, the entire training set for model training is formed by \( \overline{S_n} = S_P \cup \overline{S_N} \), and the clean version of \( \overline{S_N} \) is recorded by \( \overline{S_N} \), 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 \( \overline{S_N} \). Besides, we use \( \eta \) to denote the prior of label noise rate in \( \overline{S_N} \), 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

\[
\Pr[\hat{y}_i = -1|y_i = 1] = \eta, \quad \Pr[\hat{y}_i = 1|y_i = -1] = 0, \tag{1}
\]

where \( \Pr[\cdot] \) denotes probability. Given \( y_i \) as the true label corresponding to the observed corrupted label \( \hat{y}_i \), we have

\[
\begin{align*}
\Pr[\hat{y}_i = 1] &= \Pr[\hat{y}_i = 1|y_i = 1]\Pr[y_i = 1] + \Pr[\hat{y}_i = 1|y_i = -1]\Pr[y_i = -1] \\
&= \Pr[\hat{y}_i = 1|y_i = 1]\Pr[y_i = 1] + 0 \times \Pr[\hat{y}_i = -1] \\
&= \Pr[\hat{y}_i = 1|y_i = 1]\Pr[y_i = 1]. \tag{2}
\end{align*}
\]

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

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

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 \( \ell: \mathcal{Y} \times \mathbb{R} \rightarrow \mathbb{R} \), which penalizes the deviation from the predicted value \( h(x) \) 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

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

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 \( \ell \), so according to [Patrini et al., 2016], we can further decompose the hinge loss \( \ell \) on the contaminated negative examples into two parts, which reaches

\[
\begin{align*}
\ell(z) &= \begin{cases} 
1 - z & \text{if } z < 1, \\
0 & \text{else.}
\end{cases} \\
\ell(z) &= \begin{cases} 
\frac{1}{2}([1 - z]_+ + [1 + z]_+) + \frac{1}{2}([1 - z]_+ - [1 + z]_+) & \text{if } |z| < 1, \\
\frac{1}{2}([1 - z]_+ + [1 + z]_+) + \frac{1}{4}(-2z + [1 - z]_+ - [1 + z]_+) & \text{otherwise},
\end{cases} \tag{5}
\end{align*}
\]

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

\[
|1 - z| \leq |z| + 1, |1 + z| \geq |z| - 1. \tag{6}
\]

Eq. (5) can be further derived as

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

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

\[
\mathcal{R}_N(h, S_N) = \frac{1}{n} \sum_{k=1}^{n} \frac{1}{2} \left[ (1-y_i h(x_i))^+ + [1+y_i h(x_i)]^+ \right] + \frac{1}{2} (1 - y_i h(x_i)).
\]

(8)

Therefore, the upper bound \(\mathcal{R}_N\) 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 \(\tilde{S}_N\). 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 \(h_w(x_i) = \langle w, x_i \rangle\), where \(w\) is the model parameter, then according to (8), we have

\[
\mathcal{R}_N(h, S_N) = \frac{1}{n} \sum_{k=1}^{n} \frac{1}{2} \left[ (1-y_i h(x_i))^+ + [1+y_i h(x_i)]^+ \right] + \frac{1}{2} (1 - y_i h(x_i))
\]

\[
= \frac{1}{n} \sum_{k=1}^{n} \frac{1}{2} \left[ (1-y_i h(x_i))^+ + [1+y_i h(x_i)]^+ \right] + \frac{1}{2} \left[ \langle w, \frac{1}{n} \sum_{i=1}^{n} y_i x_i \rangle \right].
\]

(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 \(D\), namely \(\mu(S_N) = \frac{1}{n} \sum_{i=1}^{n} y_i x_i\) and \(\mu(D) = E_{(x,y) \sim D} [\langle x, y \rangle]\).

Similarly, we also define the unlabeled example centroid \(\mu(S_N) = \frac{1}{n} \sum_{i=1}^{n} y_i x_i\) and \(\mu(D) = E_{(x,y) \sim \tilde{D}} [\langle x, y \rangle]\) on the corrupted negative set \(\tilde{S}_N\) and corrupted distribution \(\tilde{D}\), respectively. By substituting \(\mu(S_N)\) into \(\mathcal{R}_N(h, S_N)\) and ignoring the constant term, (9) is transformed to

\[
\mathcal{R}_N(h, S_N) = \frac{1}{n} \sum_{i=1}^{n} \left[ (1-y_i \langle w, x_i \rangle)^+ + [1+y_i \langle w, x_i \rangle]^+ \right] = \frac{1}{2n} \langle w, \mu(S_N) \rangle.
\]

(10)

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

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

**Proof.** It is straightforward that

\[
\begin{align*}
E_{\tilde{y}}[y|x, y] &= p E_{\tilde{y}}[y|x, y] + (1-p) E_{\tilde{y}}[y|x, y] \\
&= p(1-2\eta)y + (1-p)y \\
&= (1-2p)\eta y.
\end{align*}
\]

From \(E_{\tilde{y}}[x|\{x, y\}]\), we have

\[
\begin{align*}
\mu(D) &= E_{(x,y) \sim D} [\langle x, y \rangle] \\
&= E_{(x,y) \sim \tilde{D}} [\langle x, y \rangle] \\
&= E_{(x,y) \sim \tilde{D}} [(1 - 2pn)p]y \\
&= (1-2pn)p \mu(D),
\end{align*}
\]

and

\[
E[\mu(\tilde{S}_N)] = (1-2pn)\mu(S_N).
\]

(13)

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

**Theorem 2.** Given \(\tilde{S}_N\) as the corrupted negative example, the empirical covariance matrix \(\Sigma(\mu(S_N))\) is

\[
\Sigma[\mu(\tilde{S}_N)] = \frac{1}{n-k} \sum_{i=k+1}^{n} x_i^T x_i - \frac{1}{n-k} \sum_{i=k+1}^{n} x_i^T y_i \sum_{i=k+1}^{n} y_i x_i - \frac{n}{n-k}.
\]

(14)

**Proof.** The definition of covariance matrix is

\[
\Sigma(\mu(\tilde{S}_N)) = E[\mu(\tilde{S}_N)]^T \mu(\tilde{S}_N) - E[\mu(\tilde{S}_N)]^T E[\mu(\tilde{S}_N)].
\]

(15)

Besides, since

\[
E[\mu(\tilde{S}_N)]^T \mu(\tilde{S}_N)
\]

\[
= E[\frac{1}{n-k} \sum_{i=k+1}^{n} y_i x_i]^T \frac{1}{n-k} \sum_{i=k+1}^{n} y_i x_i
\]

\[
= \frac{1}{(n-k)^2} \left( \sum_{i=k+1}^{n} E[x_i^T x_i] + \sum_{i \neq j} E[y_i y_j x_i^T x_j] \right),
\]

and

\[
E[y_i y_j x_i^T x_j] = E[\sum_{i=k+1}^{n} x_i^T x_i y_i y_j] E[\sum_{i=k+1}^{n} x_i y_i y_j].
\]

(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 \(\Sigma(\mu(S_N))\). [Gao et al., 2016].
Algorithm 1 Median-of-means estimator of corrupted negative mean

**Input:** The corrupted negative sample, \( \tilde{S}_N \); the number of groups \( g \geq 1 \);

**Output:** The median-of-means estimator, \( \hat{\mu}(\tilde{S}_N) \);

1. Randomly divide \( \tilde{S}_N \) into \( g \) groups \( \{ \tilde{S}_N^{[1]}, \tilde{S}_N^{[2]}, \ldots, \tilde{S}_N^{[g]} \} \) with almost equal size;
2. Calculate the standard empirical mean \( \mu(\tilde{S}_N^{[i]}) \) for each \( i \in [g] \) and each group \( \tilde{S}_N^{[i]} \);
3. Calculate \( r_i = \text{median}\{\mu(\tilde{S}_N^{[i]}) - \mu(\tilde{S}_N^{[g]})\} \) for each \( i \in [g] \), and then set \( i^* = \arg\min_{i \in [g]} r_i \);
4. return \( \hat{\mu}(\tilde{S}_N) \).

5 The LDCE Algorithm

Section 4 shows that we can estimate the true unlabeled example centroid \( \mu(\tilde{S}_N) \), which is significant to handle PU problem, by estimating the corrupted negative example centroid \( \hat{\mu}(\tilde{S}_N) \). In this paper, we adopt the recent generalized median-of-means estimator [Hsu and Sabato, 2014] to estimate \( \mu(\tilde{S}_N) \). The basic idea is to randomly divide the corrupted negative set \( \tilde{S}_N \) into \( g \) groups with almost equal size, and then return the generalized median of sample means for each group under \( \ell_2 \)-norm metric. Algorithm 1 presents the detailed description about this process.

Due to the influence of noise on the covariance of \( yx \), that has been discussed in Section 4, here we impose a constraint on \( \mu(\tilde{S}_N) \), which is

\[
(\mu - \hat{\mu}(\tilde{S}_N))^\top \Sigma(\hat{\mu}(\tilde{S}_N))(\mu - \hat{\mu}(\tilde{S}_N)) \leq \beta, \tag{18}
\]

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

\[
\min_{\mu, \omega} \frac{1}{n} \sum_{i=1}^{k} \ell(y_i(\omega, x_i)) + \frac{1}{2n} \sum_{i=k+1}^{n} \varphi(y_i, (\omega, x_i)) + \frac{c}{1-2\rho}(\omega, \mu) + \lambda \|\omega\|^2 \tag{19}
\]

subject to

\[
(\mu - \hat{\mu}(\tilde{S}_N))^\top \Sigma(\hat{\mu}(\tilde{S}_N))(\mu - \hat{\mu}(\tilde{S}_N)) \leq \beta, \tag{20}
\]

where \( \ell(y_i(\omega, x_i)) = \max(0, 1 - y_i(\omega, x_i)), c = -\frac{n-k}{2n}, \varphi(y_i, (\omega, x_i)) = [1 - y_i(\omega, x_i)]_+ + [1 + y_i(\omega, x_i)]_+, \rho = k/n(1 - \eta), \) and \( \lambda \|\omega\|^2 \) 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 \( \mu \), we can simply use the gradient descent algorithm to solve the minimization problem on \( \omega \), which is

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

**Input:** The corrupted sample \( \tilde{S}_n = \{(x_1, y_1), \ldots, (x_n, y_n), (x_{k+1}, y_{k+1}), \ldots, (x_n, y_n)\} \), the noisy parameter \( \eta \), the regularization parameter \( \lambda \), the approximation \( \beta \);

**Output:** The optimal classifier parameter \( \omega \);

1. Call Algorithm 1 to give an estimation of \( \hat{\mu}(\tilde{S}_N) \);
2. Calculate \( \hat{\Sigma} = \hat{\Sigma}(\hat{\mu}(\tilde{S}_N)) \) by Eq. (14);
3. Initialize \( t = 0 \) and \( \omega_0 \);
4. repeat
5. Calculate \( \mu = \hat{\mu} + \hat{\Sigma}^{-1} \omega \sqrt{\beta/(\omega^\top \hat{\Sigma}^{-1} \omega)} \)
6. Use gradient descent method to solve

\[
\omega_t = \arg\min_{\omega} \frac{1}{n} \sum_{i=1}^{k} \ell(y_i(\omega, x_i)) + \frac{1}{2n} \sum_{i=k+1}^{n} \varphi(y_i, (\omega, x_i)) + \frac{c}{1-2\rho}(\omega, \mu) + \lambda \|\omega\|^2;
\]

7. \( t = t + 1; \)
8. until convergence;
9. return The converged \( \omega \).

For fixed \( \omega \), after ignoring some constant terms, the optimization problem regarding \( \mu \)

\[
\min_{\mu} c(\omega, \mu) \tag{21}
\]

subject to

\[
(\mu - \hat{\mu}(\tilde{S}_N))^\top \Sigma(\hat{\mu}(\tilde{S}_N))(\mu - \hat{\mu}(\tilde{S}_N)) \leq \beta. \tag{22}
\]

To deal with this constrained optimization problem, we can introduce a Lagrange variable \( \rho \), and thus

\[
L(\mu, \beta) = c(\omega, \mu) - \rho (\mu - \hat{\mu}(\tilde{S}_N))^\top \Sigma(\hat{\mu}(\tilde{S}_N))(\mu - \hat{\mu}(\tilde{S}_N)) + \rho \beta. \tag{23}
\]

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

\[
\mu = \frac{c}{2\rho}(\hat{\Sigma}(\hat{\mu}(\tilde{S}_N)))^{-1} \omega + \hat{\mu}(\tilde{S}_N). \tag{24}
\]

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

\[
\min_{\rho} \frac{c}{2\rho} \omega^\top (\hat{\Sigma}(\hat{\mu}(\tilde{S}_N)))^{-1} \omega \tag{25}
\]

subject to

\[
\frac{c^2}{4\rho^2} \omega^\top (\hat{\Sigma}(\hat{\mu}(\tilde{S}_N)))^{-1} \omega \leq \beta,
\]

of which the solution is \( \rho = -\frac{1}{2} \sqrt{\omega^\top (\hat{\Sigma}(\hat{\mu}(\tilde{S}_N)))^{-1} \omega/\beta} \).

By further plugging it into Eq. (23), we derive the solution of Eq. (23) as

\[
\mu = \hat{\mu}(\tilde{S}_N) + (\hat{\Sigma}(\hat{\mu}(\tilde{S}_N)))^{-1} \omega \sqrt{\beta/(\omega^\top (\hat{\Sigma}(\hat{\mu}(\tilde{S}_N)))^{-1} \omega)}. \tag{25}
\]

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 pro-
posed 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% (η = 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)~(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. η ∈ {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.

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, 1http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html
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 increasing attention due to its great practical significance. In this section, we utilize the HockeyFight 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 at 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.

### 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.
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