Multi-Positive and Unlabeled Learning

Yixing Xu[†], **Chang Xu**[‡], **Chao Xu**[†], **Dacheng Tao**[‡]
[†]Key Laboratory of Machine Perception (MOE), Cooperative Medianet Innovation Center,
School of Electronics Engineering and Computer Science, PKU, Beijing 100871, China
[‡]UBTech Sydney AI Institute, The School of Information Technologies,
The University of Sydney, J12, 1 Cleveland St, Darlington, NSW 2008, Australia
xuyixing@pku.edu.cn, c.xu@sydney.edu.au
xuchao@cis.pku.edu.cn, dacheng.tao@sydney.edu.au

Abstract

The positive and unlabeled (PU) learning problem focuses on learning a classifier from positive and unlabeled data. Some methods have been developed to solve the PU learning problem. However, they are often limited in practical applications, since only binary classes are involved and cannot easily be adapted to multi-class data. Here we propose a one-step method that directly enables multi-class model to be trained using the given input multi-class data and that predicts the label based on the model decision. Specifically, we construct different convex loss functions for labeled and unlabeled data to learn a discriminant function F. The theoretical analysis on the generalization error bound shows that it is no worse than $k\sqrt{k}$ times of the fully supervised multi-class classification methods when the size of the data in k classes is of the same order. Finally, our experimental results demonstrate the significance and effectiveness of the proposed algorithm in synthetic and real-world datasets.

1 Introduction

Training examples are labeled as positive or negative in the conventional binary classification problem. In contrast, the positive and unlabeled (PU) learning problem aims to learn a classifier from positive and unlabeled data, where unlabeled data contain both positive and negative examples. Many real-world applications can be generalized as PU learning problems. For example, when distinguishing urban areas and non-urban areas using remote-sensing data [Li *et al.*, 2011], urban examples can easily be labeled whereas non-urban examples are too diverse to be fully labeled. In gene association studies [Yang *et al.*, 2014], confirmed causative genes of various human diseases are considered positive while all other unknown genes are treated as unlabeled data. In security systems, authorised user upload a series of photographs of his own face while any picture of a person in front of the camera is regarded as unlabeled. Some classical applications such as text classification [Fung *et al.*, 2006; Kanoun *et al.*, 2011] and information retrieval [Latulippe *et al.*, 2013; Schwenker and Trentin, 2014; Yu, 2003; Nguyen *et al.*, 2011; Li *et al.*, 2014; Liu *et al.*, 2017] can also be regarded as PU learning problems.

Some effective algorithms have been developed to solve the PU learning problem. Biased support vector machine (Biased SVM) solves the PU problem by treating unlabeled data as negative data with noise [Liu and Tao, 2016] and using a cost-sensitive SVM to generate the classifier [Liu *et al.*, 2003]. [Elkan and Noto, 2008] showed that the probability predicted by a classifier trained on positive and unlabeled examples has a constant difference from the true conditional probability. The resulting classifier can thus be constructed by first training an initial classifier using traditional supervised learning methods on positive and unlabeled data and then estimating the constant difference based on positive data. [du Plessis *et al.*, 2014] proved that convex loss functions are inapplicable to learning PU problem classifiers, due to the systematic estimation bias, and instead recommended non-convex loss functions, e.g. ramp loss, for PU learning. [Plessis *et al.*, 2015] then demonstrated that employing different convex loss functions over positive examples and unlabeled examples overcomes the difficulty in optimizing non-convex loss functions in [du Plessis *et al.*, 2014] while guaranteeing learning performance.

Although existing PU learning algorithms are effective and show promising performance in a number of different applications, they share the limitation that only binary classes are involved. Hence, they are not easy to adapt to the multi-class data which is common in real-world applications. For example, more than one person might be authorized by the security system, thus their face images are grouped into distinct positive classes. A personalized email filter should allow some spam to pass through the system in addition to non-spam emails, which are also organized into several positive classes.

Here we study the Multi-Positive and Unlabeled learning (MPU) problem, in which labeled data from multiple positive classes and unlabeled data from either the positive classes or an unknown negative class are provided for learning. We aim to construct different convex loss functions for labeled and unlabeled data to eliminate the systematic estimation bias. We achieve this by learning a discriminant function $F : \mathcal{X} \times \mathcal{Z} \rightarrow \mathbb{R}$ over the input and the encoded output pairs $(\mathbf{x}_1, \mathbf{z}_1), \cdots, (\mathbf{x}_n, \mathbf{z}_n) \in \mathcal{X} \times \mathcal{Z}$, with the new output space $\mathcal{Z}$ generated by encoding the original output space $\mathcal{Y}$. We provide a theoretical analysis of the generalization error bound of MPU, which is no worse than $k\sqrt{k}$ times of the fully supervised multi-class classification methods when the size of the data in different classes is of the same order. Finally, our experiments over synthetic and real-world MPU datasets demonstrate the practical significance of studying classification problems involving multiple positive data from different classes and unlabeled data.

2 Problem Formulation

Supposing the data are from k classes, the first $k-1$ classes are regarded as positive while the k-th class is negative. In the MPU problem of interest, we assume that the labeled data are sampled from $k-1$ positive classes and the unlabeled data could be from either positive classes or a negative class. Given a training set:

$$T = \{(\mathbf{x}_l, y_l)\}_{l=1}^{n_l} \cup \{(\mathbf{x}_u)\}_{u=1}^{n_u}, \tag{1}$$

where n_l and n_u represent the number of labeled and unlabeled samples respectively, and $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$, $y_i \in \mathcal{Y} = \{1, 2, ..., k\}$, MPU aims to learn a decision function $f : \mathcal{X} \to \mathcal{Y}$ based on the training data.

2.1 Multi-Positive and Unlabeled Learning

In classical multi-class classification, given the class prior $\pi_i = p(y = i)$, $i = 1, 2, ..., k$, we can learn the classifier based on the decision function $f(\mathbf{x})$ by minimizing the expected misclassification rate:

$$R(f) := \sum_{i=1}^{k} \pi_i R_i(f), \tag{2}$$

where $\sum_{i=1}^{k} \pi_i = 1$. $R_i(f) = P_i(f(\mathbf{x}) \neq i)$ denotes the expected misclassification rate on the i-th class, and P_i is the marginal probability. By learning a discriminant function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ which can be treated as the reliability or the score of the prediction over input/output pairs, we can derive the prediction by maximizing F given some input $\mathbf{x}$ [Tsochantaridis *et al.*, 2004]. Therefore, the decision function f can be expressed as:

$$f(\mathbf{x}; \mathbf{W}) = \arg\max_{y \epsilon \mathcal{Y}} F(\mathbf{x}, y; \mathbf{W}), \tag{3}$$

where $\mathbf{W}$ is the parameter matrix. For simplicity, we define $F(\mathbf{x}, y) = F(\mathbf{x}, y; \mathbf{W})$ in the following paragraph.

In the MPU problem, the unlabeled samples is a mixture of samples in k classes, therefore the distribution of unlabeled data can be denoted by a linear combination of the distribution of the samples in k classes. Defining P_X as the marginal probability of unlabeled data:

$$P_X = \sum_{i=1}^{k} \pi_i P_i = \sum_{i=1}^{k-1} \pi_i P_i + (1 - \sum_{i=1}^{k-1} \pi_i) P_k, \tag{4}$$

where π_i is the unknown class prior which can be estimated with the method in [Blanchard *et al.*, 2010]. Note that we have assumed that the k-th class is negative, then there are no labeled data in the k-th class. $R(f)$ must, therefore, be reformulated to exclude $R_k(f)$. We introduce $R_X(f)$ to denote the probability that the unlabeled sample has not been classified as the k-th class:

$$\begin{aligned} R_X(f) &= P_X(f(\mathbf{x}) \neq k) \\ &= \sum_{i=1}^{k-1} \pi_i P_i(f(\mathbf{x}) \neq k) + (1 - \sum_{i=1}^{k-1} \pi_i) P_k(f(\mathbf{x}) \neq k) \\ &= \sum_{i=1}^{k-1} \pi_i P_i(f(\mathbf{x}) \neq k) + (1 - \sum_{i=1}^{k-1} \pi_i) R_k(f). \end{aligned} \tag{5}$$

Note that $R_k(f)$ can be expressed with $R_X(f)$, and the distribution of the unlabeled data can be easily obtained from the input data. Hence, applying Eq.(5) to Eq.(2) and note that $\sum_{i=1}^{k} \pi_i = 1$, the expected misclassification rate $R(f)$ can be written as:

$$R(f) = \sum_{i=1}^{k-1} \pi_i R_i(f) + R_X(f) - \sum_{i=1}^{k-1} \pi_i P_i(f(\mathbf{x}) \neq k), \tag{6}$$

in which $P_i(f(X) \neq k)$ represents the probability that the sample in the i-th class has not been classified as the k-th class. Note that $R_i(f)$ ($R_X(f)$) can be treated as the probability that the sample in the i-th class (in the unlabeled data) has not been classified as the i-th (k-th) class. Therefore, each term of the expected misclassification rate $R(f)$ can be expressed as:

$$P_i(f(\mathbf{x}) \neq j) = \sum_{m=1, m \neq j}^{k} P_i(f(\mathbf{x}) = m), \tag{7}$$

in which i is the true label of the sample (which is unknown for unlabeled data). The left hand side of Eq.(7) is the rate that the sample in the i-th class has not been classified as the j-th class, which can be decomposed as the summation of $k-1$ terms, each of which represents the rate of classifying the sample in the i-th class to the m-th class.

Based on the analysis above, denoting $F(\mathbf{x}^{(i)}, f(\mathbf{x}) = j)$ as the prediction score of the sample in the i-th class which is classified as the j-th class, the empirical loss of the sample can be defined as:

$$L(F(\mathbf{x}^{(i)}, f(\mathbf{x}) \neq j)) = \frac{1}{k-1} \sum_{m=1, m \neq j}^{k} l(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = m)), \tag{8}$$

in which $F(\mathbf{x}^{(i)}, f(\mathbf{x}) \neq j)$ means the empirical loss that the sample in the i-th class has not been classified as the j-th class. And $l(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = m))$ is the loss of misclassifying a sample from the i-th class into the m-th class. Note that the loss in Eq.(8) is computed on $k-1$ classes since each term of the expected misclassification rate $R(f)$ is computed on $k-1$ classes according to Eq.(7).

Therefore, given Eq.(8), the loss function can be expressed based on Eq.(6):

$$\begin{aligned} J(F) = &\sum_{i=1}^{k-1} \pi_i E_i[L(F(\mathbf{x}^{(i)}, f(\mathbf{x}) \neq i))] + E_X[L(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))] \\ &- \sum_{i=1}^{k-1} \pi_i E_i[L(F(\mathbf{x}^{(i)}, f(\mathbf{x}) \neq k))] \\ = &\sum_{i=1}^{k-1} \pi_i E_i[L(F(\mathbf{x}^{(i)}, f(\mathbf{x}) \neq i)) - L(F(\mathbf{x}^{(i)}, f(\mathbf{x}) \neq k))] \\ &+ E_X[L(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))] \\ = &\sum_{i=1}^{k-1} \pi_i \frac{1}{k-1} E_i[l(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = k)) - l(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = i))] \\ &+ E_X[L(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))]. \end{aligned} \tag{9}$$

in which $\tilde{y}$ is some unknown label of the unlabeled data, and the last equation is derived from Eq.(8) since there are $k-2$ same terms between two $L(\cdot)$'s and only two $l(\cdot)$'s are remained after subtraction.

Since adopting the same convex loss function for positive and unlabeled data can lead to systematic estimation bias which causes an incorrect classification boundary [du Plessis *et al.*, 2014], it is appropriate to use distinct loss functions over labeled and unlabeled data. For efficiency, it is better that both loss functions are convex. Therefore, we define:

$$\begin{aligned} & l(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = j)) \\ & = \frac{1}{2}\max(0, 1 - (F(\mathbf{x}^{(i)}, f(\mathbf{x}) = i) - F(\mathbf{x}^{(i)}, f(\mathbf{x}) = j))). \end{aligned} \tag{10}$$

Since $l(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = i)) = \frac{1}{2}$, Eq.(9) becomes:

$$\begin{aligned} J(F) = & \sum_{i=1}^{k-1} \pi_i \frac{1}{k-1} E_i[l(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = k)) - \frac{1}{2}] \\ & + E_X[L(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))]. \end{aligned} \tag{11}$$

It is worth noting that the hinge loss is appropriate here to get a convex model because when the sample is correctly labeled, the loss proposed in Eq.(10) is a fixed value ($l(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = i)) = 1/2$). Therefore, both $l(\cdot)$ and $L(\cdot)$ in Eq.(11) and the model (Eq.(11)) are convex.

2.2 Discriminant Function Construction

To specify Eq.(11), we need to construct the discriminant function $F(\mathbf{x}, f(\mathbf{x}); \mathbf{W})$.

To handle multiple classes effectively, we encode class labels using vectors $(\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k) \in \mathbb{R}^r$, where r indicates the length of the code and $\mathbf{z}_i$ is the codeword of the i-th class. Given the parameter matrix $\mathbf{W}_{r \times d}$ (in which d is the dimension of the input space), we define:

$$F(\mathbf{x}^{(i)}, f(\mathbf{x}) = j; \mathbf{W}) = \langle \mathbf{W}\mathbf{x}^{(i)}, \mathbf{z}_j \rangle = (\mathbf{W}\mathbf{x}^{(i)})^\top \mathbf{z}_j. \tag{12}$$

In Eq.(12), the original input space $\mathcal{X}$ is mapped into $\mathcal{Z}$, and the geometric result of the inner product is the projection value of the input data to the codeword $\mathbf{z}_j$, which can be treated as the score on the j-th class. The decision function can be expressed as Eq.(3).

Note that in order to achieve an effective result, the prediction score of $\mathbf{x}^{(i)}$ on the i-th class should be the largest one, and the margins between the scores of $\mathbf{x}^{(i)}$ on different classes should be large enough. The following optimization problem is a possible approach to encode y by maximizing the margins between the codewords:

$$\begin{aligned} & \max_{r, \mathbf{z}_1, \cdots, \mathbf{z}_k} [\min_{i \neq j} ||\mathbf{z}_i - \mathbf{z}_j||^2] \\ & \text{s.t.} \quad ||\mathbf{z}_i|| = 1, \mathbf{z}_i \in \mathbb{R}^r, i = 1, 2, \cdots, k. \end{aligned} \tag{13}$$

in which the length of the codewords is fixed as $r = k-1$. It has been proved in [Saberian and Vasconcelos, 2011] that the vertices of a $k-1$ dimensional regular simplex centered at the origin are the solutions of Eq.(13).

Assuming that there are n_i labeled samples belonging to the i-th class, $i = 1, 2, \cdots, k-1$, and n_u samples are unlabeled. Applying Eq.(12) to Eq.(11), the empirical loss function can be written as:

$$\begin{aligned} \hat{J}(\mathbf{W}, \tilde{\mathbf{y}}) = & \frac{\lambda}{2}||\mathbf{W}||_F^2 + \frac{1}{k-1}\sum_{i=1}^{k-1}\frac{\pi_i}{2n_i}\sum_{j=1}^{n_i}[(\mathbf{W}\mathbf{x}_j^{(i)})^\top(\mathbf{z}_k - \mathbf{z}_i)]_+ \\ & + \frac{1}{k-1}\sum_{l=1}^{k-1}\frac{1}{2n_u}\sum_{j'=1}^{n_u}[1 + (\mathbf{W}\mathbf{x}_{j'}^{(\tilde{y}_{j'})})^\top(\mathbf{z}_l - \mathbf{z}_{\tilde{y}_{j'}})]_+, \end{aligned} \tag{14}$$

where $\mathbf{x}_j^{(i)}$ denotes the j-th sample in the i-th class and $\mathbf{x}_{j'}^{(\tilde{y}_{j'})}$ denotes the j'-th sample which is unlabeled, whose true label is $\tilde{y}_{j'}$ which is unknown. The first term is the regularization term, and $[x]_+ = \max(x, 0)$.

Note that the parameter matrix $\mathbf{W}$ is the only parameter needing to be optimized in the first term on the right hand side of Eq.(14). In the second term, however, since the input data are unlabeled, we also need to optimize $\mathbf{z}_{\tilde{y}_{j'}}$, which is the encoded result of the true class of the unlabeled data. Therefore, defining $\tilde{\mathbf{y}} = (\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_{n_u})$ (where $\tilde{y}_i \in \{1, 2, ..., k\}$), there are two parameters $\mathbf{W}$ and $\tilde{\mathbf{y}}$ needing to be optimized.

2.3 Optimization of Problem (Eq.(14))

Giving the empirical loss function Eq.(14), there are two parameter matrices in the loss function that need to be optimized, where $\mathbf{W}$ is the parameter matrix in the discriminant function Eq.(12) and $\tilde{\mathbf{y}}$ denotes the true labels of the unlabeled samples. The proposed optimization function Eq.(14) is non-convex. However, fixing one of the variables, and the problem becomes convex w.r.t. the other one. Thus, we propose an optimization algorithm by fixing one of the variables alternatively until the optimization function converges. The details are given by the pseudo-code in Algorithm 1.

Algorithm 1 Multi-positive and unlabeled learning

```
Input: Number of classes $k$, dataset $T = \{(\mathbf{x}_l, y_l)\}_{l=1}^{n_l} \cup \{(\mathbf{x}_u)\}_{u=1}^{n_u}$, encoded matrix $\mathbf{Z}$.
Initialization: set $\mathbf{W} \leftarrow \mathbf{W}_0, \tilde{\mathbf{y}} \leftarrow \tilde{\mathbf{y}}_0$.
  repeat
    • Fixed $\mathbf{W}$, update $\tilde{\mathbf{y}}$ that minimize $\hat{J}(\mathbf{W}, \tilde{\mathbf{y}})$
      for $i = 1$ to $n_u$ do
        for $c = 1$ to $k$ do
          Compute $\hat{J}(\mathbf{W}, \tilde{\mathbf{y}}|\tilde{y}_i = c)$
        end for
        $j = \arg\min_c \hat{J}(\mathbf{W}, \tilde{\mathbf{y}}|\tilde{y}_i = c)$
        $\tilde{y}_i = j$
      end for
    • Fixed $\tilde{\mathbf{y}}$, update $\mathbf{W}$ that minimize $\hat{J}(\mathbf{W}, \tilde{\mathbf{y}})$
  until converge
Output: parameter matrix $\mathbf{W}$
```

3 Theoretical Analysis

In this section, we provide the theoretical analysis of the generalization error bounds of the proposed method. The proofs are shown in Appendix A∼Appendix C.

Firstly, substitute a generalized form for Eq.(10):

$$\begin{aligned} & \hat{l}_\rho(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = j)) \\ & = \frac{1}{2}\max(0, 1 - \frac{1}{\rho}(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = i) - F(\mathbf{x}^{(i)}, f(\mathbf{x}) = j))). \end{aligned}$$

For a given dataset, there is an upper bound D, that makes $0 \leq \hat{l}_\rho \leq D$. Without loss of generality, scaling l_ρ which makes $\hat{l}_\rho \in [0, 1]$, the normalized loss l_ρ is defined as:

$$\begin{aligned} & l_\rho(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = j)) \\ = & \frac{1}{2}\max(0, \frac{1}{D} - \frac{1}{\rho D}(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = i) - F(\mathbf{x}^{(i)}, f(\mathbf{x}) = j))), \end{aligned} \tag{15}$$

and Eq.(8) becomes:

$$L_\rho(F(\mathbf{x}^{(i)}, f(\mathbf{x}) \neq j)) = \frac{1}{k-1} \sum_{m=1, m\neq j}^{k} l_\rho(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = m)), \tag{16}$$

so Eq.(11) can be rewritten as:

$$\begin{aligned} J(F) = &\sum_{i=1}^{k-1} \pi_i E_{p(\mathbf{x}|y=i)}[l_\rho(F(\mathbf{x}^{(i)}, f(\mathbf{x}) = k)) - \frac{1}{2}] \\ &+ E_{p(\mathbf{x})}[L_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))], \end{aligned} \tag{17}$$

where the first term represents the loss of the labeled positive data and the second term represents the loss of the unlabeled data.

Define $\alpha = \sup_{x \in R^d} \sqrt{k(x,x)}$ where $k(\cdot,\cdot)$ is a PDS kernel and $\beta = \sup(||\mathbf{w}||)$ for some $\beta \geq 0$.

In the following paragraph, we show the generalization error bounds of the two terms in Eq.(17) in Theorem 2 and 3 respectively. We begin with the first term, and the constant $\frac{1}{2}$ is ignored for simplicity.

Theorem 1. *Let $h_j(\mathbf{x}^{(i)}) = F(\mathbf{x}^{(i)}, f(\mathbf{x}) = j) \in H_i \subseteq \mathbb{R}^{\mathcal{X}\times\mathcal{Y}}$ where H_i is a hypothesis set with $\mathcal{Y} = 1, 2, ..., k$. Fix $\rho > 0$. For any $0 < \delta < 1$, with probability at least $1 - \delta$, the generalization bound holds for all $h_i \in H_i$:*

$$\begin{aligned} &E_{p(\mathbf{x}|y=i)}[l_\rho(h_i(\mathbf{x}^{(i)}), h_k(\mathbf{x}^{(i)}))] - \frac{1}{n_i}\sum_{j=1}^{n_i} l_\rho(h_i(\mathbf{x}_j^{(i)}), h_k(\mathbf{x}_j^{(i)})) \\ &\leq \frac{2k^2}{\rho D}(\sum_{c=1}^{k-1} \frac{\alpha\beta}{\sqrt{n_c}} + \frac{\alpha\beta}{\sqrt{n_u}}) + \sqrt{\frac{log\frac{1}{\delta}}{2n_i}}. \end{aligned} \tag{18}$$

Theorem 2 presents the error bound for labeled data from each positive class, so we can summarize them to get the error bound of the first term in Eq.(17).

Note that it is difficult to get the error bound of the second term in Eq.(17), so we aim to decompose it into two terms using the following lemma.

Lemma 1. *Define*

$$L'_\rho(F(\mathbf{x}^{(i)}, f(\mathbf{x}) \neq j)) = \frac{k}{2k-y} L_\rho(F(\mathbf{x}^{(i)}, f(\mathbf{x}) \neq j)),$$

the last term of Eq.(17) can be decomposed as follows:

$$\begin{aligned} &E_{p(\mathbf{x})}[L_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))] \\ &= \sum_{c=1}^{k-1} \pi_c^*(\frac{2k-c}{k}) E_{p(\mathbf{x}|y=c)}[L'_\rho(F(\mathbf{x}^{(c)}, f(\mathbf{x}) \neq k))] \\ &+ E_{p(\mathbf{x},y)}[L'_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))]. \end{aligned} \tag{19}$$

where $\pi_c^* := p(y = c)$ is the true class prior of the c-th class.

Based on the decomposition result from Lemma 1, the generalization error bound of the second term in Eq.(17) can be obtained by bounding each term in Eq.(19) (see Appendix C for detail). The result is shown in the following theorem:

Theorem 2. *Fix $\rho > 0$. For any $0 < \delta < 1$, with probability at least $1 - \delta$ over the samples in $N = N_1 \cup \cdots \cup N_{k-1} \cup N_u$, the generalization bound holds for all $h_i \in H_i$:*

$$\begin{aligned} &E_{p(\mathbf{x})}[L_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))] - \frac{1}{n_u}\sum_{j'=1}^{n_u} L'_\rho(F(\mathbf{x}_{j'}^{(\tilde{y})}, f(\mathbf{x}_{j'}) \neq k)) \\ &\leq \sum_{c=1}^{k-1} \frac{\pi_c^*}{n_c}(\frac{2k-c}{k}) \sum_{j=1}^{n_c} L'_\rho(F(\mathbf{x}^{(c)}, f(\mathbf{x}) \neq k)) \\ &+ \sum_{c=1}^{k-1} \pi_c^* \frac{2k^2}{\rho D}(\sum_{c'=1}^{k-1} \frac{\alpha\beta}{\sqrt{n_{c'}}} + \frac{\alpha\beta}{\sqrt{n_u}}) + \frac{2k^2}{\rho D}(\sum_{c'=1}^{k-1} \frac{\alpha\beta}{\sqrt{n'_c}} + \frac{\alpha\beta}{\sqrt{n_u}}) \\ &+ \sum_{c=1}^{k-1} \pi_c^*(\frac{2k-c}{k})\sqrt{\frac{\log\frac{1}{\delta}}{2n_c}} + \sqrt{\frac{log\frac{1}{\delta}}{2n_u}}. \end{aligned} \tag{20}$$

Summarizing the inequality in Theorem 2 and 3, we get the generalization error bound of Eq.(17), and the order of the error bound is shown below:

Theorem 3. *As $n_i \to \infty, i = 1, 2, ..., k-1$; $n_u \to \infty$ and $k \to \infty$, the generalization error bound of the proposed method is of order:*

$$\mathcal{O}\left(k^2(\frac{1}{\sqrt{n_1}} + \cdots + \frac{1}{\sqrt{n_{k-1}}} + \frac{1}{\sqrt{n_u}})\right). \tag{21}$$

Note that for fully labeled data, the samples are i.i.d., and the generalization error bound would be of order $\mathcal{O}(k^2/\sqrt{n_1 + \cdots + n_{k-1} + n_u})$. The proposed method is, therefore, no worse than $k\sqrt{k}$ times of the fully supervised multi-class classification methods when the size of the data in different classes is of the same order.

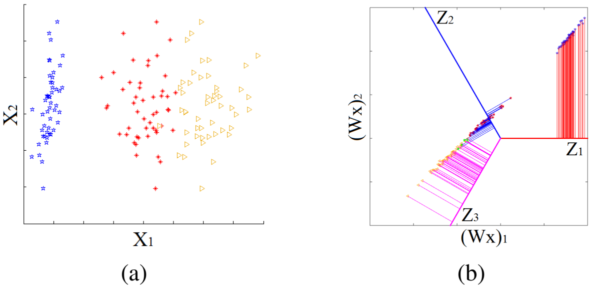
4 Experiments

We conducted two sets of experiments: a toy experiment to qualitatively show the properties of the proposed algorithm, and a quantitative evaluation of the algorithm on real-world datasets. The new MPU algorithm was compared with the binary PU learning methods (BPU) such as BPU(DH) method in [Plessis *et al.*, 2015] which created a double hinge loss function for the unlabeled data, BPU(ramp) method in [du Plessis *et al.*, 2014] where the non-convex ramp loss was used, and biased SVM method [Liu *et al.*, 2003]. All binary methods were generalized to solve the multi-class problem with one-versus-all method.

4.1 Experiment on Toy Data

We first conducted an experiment on the IRIS dataset from the UCI repository. There are 3 classes in the dataset with 50 samples for each. The first two classes were treated as positive and the third class as negative. For the simplicity of illustration, we reduced the dimensionality to 2 via principal component analysis, with 50 labeled data samples (25 in class 1 and 25 in class 2) and 100 unlabeled data samples drawn (Fig.1 (a)). The examples were projected into space $\mathcal{Z}$ and the classification result is shown in Fig.1 (b).

The three lines represent the codeword $\mathbf{z}_i$, while the three classes of mapped input data are shown in different colors (the misclassified points are in green). Each point was projected to the three codewords (only one projection is drawn for simplicity). The projection value of input data to $\mathbf{z}_i$ represents the score on the i-th class. The label of the class with the maximum score was assigned to the input data. Nearly all of the data was classified correctly.



(a) (b)

Figure 1: (a) Sample distribution with the dimensionality reduced to 2. Class 1,2 and 3 are drawn in blue, red and orange respectively. (b) The classification results in space $\mathcal{Z}$.

Table 1: Datasets statistics

Dataset	examples	variables	classes
Image Segment	2310	19	7
Letter	20000	16	26
USPS	11000	256	10
MNIST	70000	784	10

4.2 Real-world Data Set

Experiments were conducted on four different datasets, and the relevant metadata for each dataset are shown in Table 1.

The conventional methods for the binary PU learning problem were generalized to solve the MPU learning problem with one-versus-all method and were compared with the new MPU algorithm in terms of classification accuracy and training time. All datasets were preprocessed, with half of the samples in each positive class regarded as labeled, while the other half together with all the samples in the negative class regarded as unlabeled. The class priors were assumed to be known at the time of training[1]. Furthermore, the fully labeled datasets trained with linear SVM were compared with all the methods mentioned above.

Classification accuracies are shown in Table 2 (All methods use linear kernal). The proposed MPU algorithm achieves the best result of all the PU learning methods and is comparable to the linear SVM method trained on fully labeled data, since our one-step method allows direct model to be trained using the given input data and obtains the label based on all the model decisions, while the generalized BPU methods share the disadvantage such as error accumulation that one-versus-all method has.

Table 2: Classification accuracy(with 20% test data).

Method	Image Segment	Letter	USPS	MNIST
Linear SVM	93.51	84.05	96.32	93.61
MPU	**90.26**	**70.12**	**92.86**	**90.78**
BPU(DH)	88.31	62.98	89.13	86.92
BPU(Ramp)	88.74	63.18	89.94	87.04
Biased-SVM	85.93	45.27	87.14	84.98

Table 3: Classification error rate on 'Image Segment'.

Dataset	MPU	BPU(DH)	BPU(Ramp)	Biased SVM
Class 1vs7	**0.15**	0.61	0.45	0.30
Class 2vs7	**2.73**	3.18	3.18	2.88
Class 3vs7	3.64	3.79	**3.48**	4.55
Class 4vs7	**1.06**	1.66	1.67	1.97
Class 5vs7	**0.45**	0.76	0.61	0.61
Class 6vs7	0.61	0.76	0.61	**0.45**

The binary PU learning results are shown in Table 3. The seventh class was fixed as the negative class, while the other classes were treated as positive class. One of the positive class was chosen at a time, and half of the examples in the chosen positive class were labeled while the other half together with the negative examples were unlabeled. Our proposed method shows comparable performance on the binary PU learning problem with other BPU methods in terms of classification accuracy.

Next, we illustrate the robustness of our proposed methods on the choice of the negative class. Fig.2 shows the classification accuracies for choosing different classes as the negative class. The black line represents the accuracy of the method training on a fully labeled dataset. It is shown that the fluctuate of the classification accuracy is small when choosing different classes as the negative class. Thus our proposed methods is robust on the choice of the negative class.

The training time on six datasets given in Fig.3 show that the MPU algorithm is faster than other generalized binary PU methods on large datasets, since the one-versus-all method needs to train k classifiers with all samples in the datasets.

5 Conclusion

This paper presents the MPU algorithm for solving the multi-positive and unlabeled learning problem. We show that the MPU algorithm allows direct model to be trained with respect to the multi-class input data rather than via a two-step approach which often leads to a high classification error rate. The theoretical analysis shows that the generalization error bounds of the MPU algorithm are comparable to $k\sqrt{k}$ times of fully supervised multi-class classification methods when the size of the data in different classes is of the same order. Experimentally, the proposed MPU algorithm outperforms current state-of-the-art methods in the MPU problem and is as accurate in the BPU problem with a less computational burden. Moreover, our method is comparable to the methods training using fully labeled data and robustly chooses the negative class.

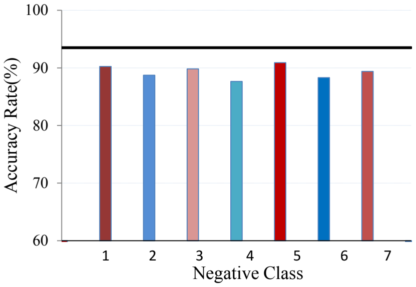
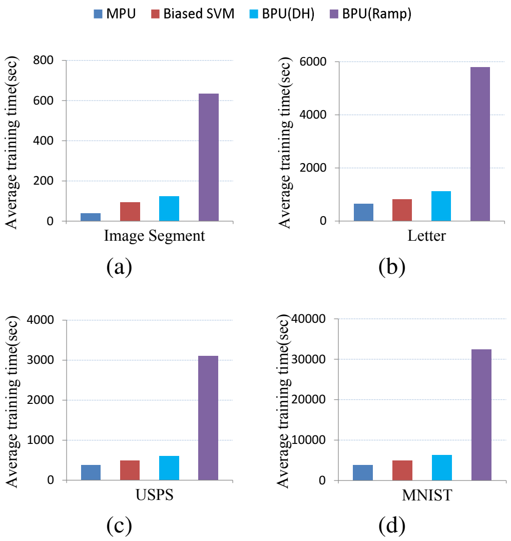


Figure 2: The classification accuracies with different classes as the negative class.

[1] The class priors can be estimated with the methods in [Blanchard *et al.*, 2010] in practice.



(a) (b) (c) (d)

Figure 3: Average training time of different methods on four datasets.

A Proof of Theorem 1

Based on the definitions 3.1 and 3.2 in [Mohri *et al.*, 2012], given $N_i = (\mathbf{x}_1^{(i)}, ..., \mathbf{x}_{n_i}^{(i)})$, $i = 1, 2, ..., k-1; N_u = (\mathbf{x}_1, ..., \mathbf{x}_{n_u})$; $N = N_1 \cup \cdots \cup N_{k-1} \cup N_u$, define the empirical Rademacher complexity of H_i with respect to the sample N_i:

$$\hat{\mathcal{R}}_{N_i}(H_i) = \underset{\sigma}{E}[\sup_{h_i \in H_i} \frac{1}{n_i} \sum_{j=1}^{n_i} \sigma_i h_i(\mathbf{x}_j^{(i)})], \tag{22}$$

where $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_{n_i})^\top$ and $\sigma_i \in \{-1, +1\}$. Denote by $\mathcal{R}_{n_i}(H_i)$ the Rademacher complexity w.r.t. $p(\mathbf{x}|y = i)$, and $\mathcal{R}_{n_u}(H)$ the Rademacher complexity w.r.t. $p(\mathbf{x}$, then we have:

$$\mathcal{R}_{n_i}(H_i) = \underset{\mathbf{x} \in N_i}{E}[\hat{\mathcal{R}}_{N_i}(H_i)]. \tag{23}$$

We then have (Proposition 8.1 in [Mohri *et al.*, 2012]):

$$\begin{aligned} &\mathcal{R}_{n_i}(H_i) \leq \frac{\alpha\beta}{\sqrt{n_i}}, \quad i = 1, 2, ..., k-1 \\ &\mathcal{R}_{n_u}(H) \leq \frac{\alpha\beta}{\sqrt{n_u}}. \end{aligned} \tag{24}$$

where $\alpha = \sup_{x \in R^d} \sqrt{k(x, x)}$ with $k(\cdot, \cdot)$ be a PDS kernel and $\beta = \sup(||\mathbf{w}||)$ for some $\beta \geq 0$.

Define $\mathcal{G} = \max\{h_1, h_2, ..., h_k\}$, then the empirical Rademacher complexity of $\mathcal{G}$ can be upper bounded by (Lemma 8.1 in [Mohri *et al.*, 2012]):

$$\hat{\mathcal{R}}_N(\mathcal{G}) \leq \sum_{c=1}^{k-1} \frac{\alpha\beta}{\sqrt{n_c}} + \frac{\alpha\beta}{\sqrt{n_u}}. \tag{25}$$

Applying Eq.(25) to Theorem 8.1 in [Mohri *et al.*, 2012], and note that the Lipschitz constant of l_ρ is $\frac{1}{\rho D}$. We then finish the proof.

B Proof of Lemma 1

Assuming that $\pi_c^* := p(y = c)$ is the true class prior of the c-th class. Subsequently,

$$\begin{aligned} E_{p(\mathbf{x})}[L_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))] &= \int_{\mathbb{R}^d} L_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))p(\mathbf{x})d\mathbf{x} \\ &= \int_{\mathbb{R}^d} \sum_y L_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))p(\mathbf{x}, y)d\mathbf{x}. \end{aligned} \tag{26}$$

Given

$$L'_\rho(F(\mathbf{x}^{(i)}, f(\mathbf{x}) \neq j)) = \frac{k}{2k - y} L_\rho(F(\mathbf{x}^{(i)}, f(\mathbf{x}) \neq j)),$$

Eq.(26) then become:

$$\begin{aligned} &E_{p(\mathbf{x})}[L_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))] \\ &= \int_{\mathbb{R}^d} \sum_y L'_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))(\frac{2k - y}{k})p(\mathbf{x}, y)d\mathbf{x} \\ &= \int_{\mathbb{R}^d} \sum_y L'_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))[\sum_{c=1}^{k} (1 + \frac{2k - c}{k})p(\mathbf{x}, y = c)]d\mathbf{x} \\ &= \sum_{c=1}^{k-1} \pi_c(\frac{2k - c}{k}) \int_{\mathbb{R}^d} L'_\rho(F(\mathbf{x}^{(c)}, f(\mathbf{x}) \neq k))p(\mathbf{x}|y = c)d\mathbf{x} \\ &\quad + \int_{\mathbb{R}^d} \sum_y L'_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))p(\mathbf{x}, y)d\mathbf{x} \\ &= \sum_{c=1}^{k-1} \pi_c(\frac{2k - c}{k})E_{p(\mathbf{x}|y=c)}[L'_\rho(F(\mathbf{x}^{(c)}, f(\mathbf{x}) \neq k))] \\ &\quad + E_{p(\mathbf{x},y)}[L'_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))]. \end{aligned}$$

This decomposition is important to the following proof of the error bound.

C Proof of Theorem 2

In this appendix, we first give the error bound of each term in the right hand side of Eq.(19) using Lemma 2 and 3.

Lemma 2. *Fix $\rho > 0$. For any $0 < \delta < 1$, with probability at least $1 - \delta$ over the samples in $N_u = (\mathbf{x}_1, ..., \mathbf{x}_{n_u})$, the generalization bound holds for all $h_i \in H_i$:*

$$\begin{aligned} &E_{p(\mathbf{x},y)}[L'_\rho(F(\mathbf{x}^{(\tilde{y})}, f(\mathbf{x}) \neq k))] - \frac{1}{n_u} \sum_{j'=1}^{n_u} L'_\rho(F(\mathbf{x}_{j'}^{(\tilde{y})}, f(\mathbf{x}_{j'}) \neq k)) \\ &\qquad \leq \frac{2k^2}{\rho D}(\sum_{c=1}^{k-1} \frac{\alpha\beta}{\sqrt{n_c}} + \frac{\alpha\beta}{\sqrt{n_u}}) + \sqrt{\frac{log\frac{1}{\delta}}{2n_u}}. \end{aligned} \tag{27}$$

Lemma 3. *Fix $\rho > 0$. For any $0 < \delta < 1$, with probability at least $1 - \delta$ over the samples in $N_i = (\mathbf{x}_1^{(i)}, ..., \mathbf{x}_{n_i}^{(i)})$, the generalization bound holds for all $h_i \in H_i$:*

$$\begin{aligned} &E_{p(\mathbf{x}|y=i)}[L'_\rho(F(\mathbf{x}^{(i)}, f(\mathbf{x}) \neq k))] - \frac{1}{n_i} \sum_{j=1}^{n_i} L'_\rho(F(\mathbf{x}_j^{(i)}, f(\mathbf{x}_j) \neq k)) \\ &\leq \frac{k}{2k - c} \frac{2k^2}{\rho D}(\sum_{c=1}^{k-1} \frac{\alpha\beta}{\sqrt{n_c}} + \frac{\alpha\beta}{\sqrt{n_u}}) + \sqrt{\frac{log\frac{1}{\delta}}{2n_i}}. \end{aligned} \tag{28}$$

Note that L'_ρ maps to $[0, 1]$, and the Lipschitz constant of L'_ρ in Lemma 2 and 3 is $\frac{1}{\rho D}$ and $\frac{k}{2k-c}\frac{1}{\rho D}$ respectively, so the proof is analogous to the one in Theorem 1.

Now the generalization error bound of the second term in Eq.(17) can easily be proved by applying the inequalities in Lemma 2 and 3 into the equality in Lemma 1.

Acknowledgements

This research is partially supported by grants from NSFC 61375026 and 2015BAF15B00, and Australian Research Council Projects FT-130101457, and DP-140102164, and LP-150100671.

References

[Blanchard *et al.*, 2010] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Semi-supervised novelty detection. *The Journal of Machine Learning Research*, 11:2973–3009, 2010.

[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.

[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. ACM, 2008.

[Fung *et al.*, 2006] Gabriel Pui Cheong Fung, Jeffrey X Yu, Hongjun Lu, and Philip S Yu. Text classification without negative examples revisit. *Knowledge and Data Engineering, IEEE Transactions on*, 18(1):6–20, 2006.

[Kanoun *et al.*, 2011] Slim Kanoun, Adel M Alimi, and Yves Lecourtier. Natural language morphology integration in off-line arabic optical text recognition. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(2):579–590, 2011.

[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 *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2480–2487. AAAI Press, 2013.

[Li *et al.*, 2011] Wenkai Li, Qinghua Guo, and Charles Elkan. A positive and unlabeled learning algorithm for one-class classification of remote-sensing data. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(2):717–725, 2011.

[Li *et al.*, 2014] Sheng Li, Ming Shao, and Yun Fu. Locality linear fitting one-class svm with low-rank constraints for outlier detection. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 676–683. IEEE, 2014.

[Liu and Tao, 2016] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2016.

[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 *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 179–186. IEEE, 2003.

[Liu *et al.*, 2017] Tongliang Liu, Dacheng Tao, Mingli Song, and Stephen J Maybank. Algorithm-dependent generalization bounds for multi-task learning. *IEEE transactions on pattern analysis and machine intelligence*, 39(2):227–241, 2017.

[Mohri *et al.*, 2012] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[Nguyen *et al.*, 2011] Minh Nhut Nguyen, Xiaoli-Li Li, and See-Kiong Ng. Positive unlabeled leaning for time series classification. In *IJCAI*, volume 11, pages 1421–1426. Citeseer, 2011.

[Plessis *et al.*, 2015] Marthinus D Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1386–1394, 2015.

[Saberian and Vasconcelos, 2011] Mohammad J Saberian and Nuno Vasconcelos. Multiclass boosting: Theory and algorithms. In *Advances in Neural Information Processing Systems*, pages 2124–2132, 2011.

[Schwenker and Trentin, 2014] Friedhelm Schwenker and Edmondo Trentin. Pattern classification and clustering: a review of partially supervised learning approaches. *Pattern Recognition Letters*, 37:4–14, 2014.

[Tsochantaridis *et al.*, 2004] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104. ACM, 2004.

[Yang *et al.*, 2014] Peng Yang, Xiaoli Li, Hon-Nian Chua, Chee-Keong Kwoh, and See-Kiong Ng. Ensemble positive unlabeled learning for disease gene identification. 2014.

[Yu, 2003] Hwanjo Yu. Svmc: Single-class classification with support vector machines. In *IJCAI*, pages 567–574. Citeseer, 2003.