Leveraging Latent Label Distributions for Partial Label Learning

Lei Feng and Bo An
School of Computer Science and Engineering, Nanyang Technological University, Singapore
feng0093@e.ntu.edu.sg, boan@ntu.edu.sg

Abstract

In partial label learning, each training example is assigned a set of candidate labels, only one of which is the ground-truth label. Existing partial label learning frameworks either assume each candidate label of equal confidence or consider the ground-truth label as a latent variable hidden in the indiscriminate candidate label set, while the different labeling confidence levels of the candidate labels are regrettably ignored. In this paper, we formalize the different labeling confidence levels as the latent label distributions, and propose a novel unified framework to estimate the latent label distributions while training the model simultaneously. Specifically, we present a biconvex formulation with constrained local consistency and adopt an alternating method to solve this optimization problem. The process of alternating optimization exactly facilitates the mutual adaption of the model training and the constrained label propagation. Extensive experimental results on controlled UCI datasets as well as real-world datasets clearly show the effectiveness of the proposed approach.

1 Introduction

Partial label (PL) learning is a specific type of weakly supervised learning [Cour et al., 2011], in which each instance is associated with a set of candidate labels. However, only one of the candidate labels is the ground-truth label, which is concealed in the training process. This learning problem is also termed as ambiguous label learning [Hüllermeier and Beringer, 2006; Zeng et al., 2013; Chen et al., 2014; Chen et al., 2017] or superset label learning [Liu and Dietterich, 2012; Liu and Dietterich, 2014; Hüllermeier and Cheng, 2015; Gong et al., 2017]. Since precisely labeled data are too expensive to be collected in reality, partial label learning has various application domains, such as ecoinformatics [Liu and Dietterich, 2012], image annotation [Cour et al., 2009; Zeng et al., 2013] and web mining [Luo and Orabona, 2010], etc.

Formally speaking, suppose we have \( m \) training examples \( \mathcal{X} = \{x_1, x_2, \ldots, x_m\} \subseteq \mathbb{R}^{m \times n} \) with \( n \) dimensions, and their candidate label sets are denoted by \( \{S_1, S_2, \ldots, S_m\} \), respectively. The ground-truth labels of these \( m \) examples are \( \{y_1, y_2, \ldots, y_m\} \) with \( y_i \in S_i (i \in [m]) \), while they are not directly accessible in the model training. Given the label space denoted by \( \mathcal{Y} = \{1, 2, \ldots, l\} \), the task of partial label learning is to learn a function: \( f : \mathcal{X} \rightarrow \mathcal{Y} \) from the imprecisely labeled training set \( \mathcal{D} = \{(x_i, S_i) | i \in [m]\} \) to accurately predict the label of the test example.

To learn with such PL examples, the key is how to properly deal with the candidate labels. To this end, there are mainly two learning frameworks, including the average-based framework and the identification-based framework. For the average-based framework, each candidate label is treated equally in the model training [Cour et al., 2011; Zhang et al., 2016]. For the identification-based framework, the ground-truth label is considered as a latent variable hidden in the indiscriminate candidate label set [Jin and Ghahramani, 2003; Liu and Dietterich, 2012; Chen et al., 2014; Nguyen and Caruana, 2008; Yu and Zhang, 2016]. They all make predictions by aggregating the modeling outputs of the candidate labels without discrimination, while the confidence of each candidate label being the ground-truth label is regretfully ignored. As a consequence, these approaches may be suboptimal, since each candidate label normally makes different contributions to the model training.

To address this problem, we formalize the different labeling confidence levels of the candidate labels as the latent label distributions, and propose the LALO (partial label learning with LAtent Label distributiOns) approach. LALO first introduces a novel unified framework that estimates the latent label distributions while training the model simultaneously, and then presents a biconvex formulation with constrained local consistency, finally adopts an alternating method to solve this optimization problem. On the one hand, the inductive model is discriminatively trained by minimizing the least squares loss of fitting the latent label distributions. On the other hand, the latent label distributions are regularized by the modeling outputs via a constrained label propagation procedure specifically for the PL properties. Through the mutual promotion of the model training and the label propagation, the ground-truth label can be identified by optimally estimating the label distributions. The effectiveness of LALO is validated by experiments on 4 controlled UCI datasets and 5 real-world datasets.

The rest of this paper is organized as follows. Section 2
briefly reviews related work. Section 3 introduces the LA-
LO approach. Section 4 presents the technical details of the
alternating optimization method. Section 5 reports the ex-
perimental results of comparative studies. In the end, Section 6
concludes this paper and discusses future research issues.

2 Related Work

Due to the difficulty in dealing with ambiguous labeling
information of PL examples, there are only two common par-
tial label learning frameworks, including the average-based
framework and the identification-based framework.

The average-based framework normally treats each can-
didate label equally in the model training, and averages the
modeling outputs of all the candidate labels for prediction-
s. Following this framework, some instance-based ap-
proaches [Hültermeier and Beringer, 2006; Zhang and Yu, 2015]
predict a test instance by averaging the candidate labeling
information of its neighbors. In addition, some parametric
approaches assume a parametric model \( F(x_i, y; \theta) \) [Cour et
al., 2011; Zhang et al., 2016] that discriminates the average
modeling output of the candidate labels from that of the non-
candidate labels, i.e., \( \max(\sum_{i=1}^{m} \left( \frac{1}{|S_i|} \sum_{y \in S_i} F(x_i, y; \theta) \right) - \frac{1}{|S_i|} \sum_{y \notin S_i} (S_i F(x_i, y; \theta))) \) where \( S_i \) and \( S_i \) denote the
candidate and non-candidate label set respectively. Although
this framework is intuitive, the obvious drawback is that the
ground-truth label may be overwhelmed by other candidate
(false positive) labels without discrimination.

Instead of maximizing the average modeling output of all
the candidate labels, the identification-based framework aims
to directly maximizing the modeling output of exactly one
candidate label, which is distinguished as the ground-truth la-
bel. Existing approaches following this framework consider the
ground-truth label as a latent variable determined by \( y_i = \arg \max_{y \in S_i} F(x_i, y; \theta) \). Generally, the objective function
is optimized according to the maximum likelihood criterion:
\( \max(\sum_{i=1}^{m} \log(\sum_{y \in S_i} \frac{1}{|S_i|} F(x_i, y; \theta))) \) [Jin and Ghahra-
mani, 2003; Liu and Dietterich, 2012] or the maximum
margin criterion: \( \max(\sum_{i=1}^{m} \max_{y \in S_i} \frac{1}{|S_i|} F(x_i, y; \theta) - \max_{y \in S_i} \frac{1}{|S_i|} F(x_i, y; \theta))) \) [Nguyen and Caruana, 2008;
Yu and Zhang, 2016]. Because of indiscriminately target-
ing the ground-truth label within the candidate label set, the
identification-based framework is sensitive to the false posi-
tive labels that co-occur with the ground-truth label.

In a nutshell, the above learning frameworks train the
model with the modeling outputs of the candidate labels indi-
scriminate (i.e., the same weight \( \frac{1}{|S_i|} \)), while the different la-
beling confidence levels of the candidate labels are regrett-
ably ignored. To address this problem, a novel unified par-
tial label learning framework will be introduced in the next
section. Following this framework, a biconvex formulation
is presented to estimate the latent label distributions while training the
model simultaneously.

3 The LALO Approach

For each training example, we receive a feature vector \( x_i \in \mathbb{R}^n \) and its corresponding label vector \( y_i \in \{0, 1\}^l \) with \( l \)
labels. Suppose \( m \) denotes the number of training examples,
\( X \in \mathbb{R}^{m \times n} \) and \( Y \in \{0, 1\}^{m \times l} \) are the instance matrix and
label matrix, respectively. In this setting, \( y_{ij} = 1 \) means the
\( i \)-th training sample is assigned the \( j \)-th label.

Existing partial label learning frameworks indiscriminate-
ly train the model with noise-corrupted label matrix \( Y \in \{0, 1\}^{m \times l} \), in which the labeling confidence of each can-
didate label is not discriminated. However, each candidate
label normally makes different contributions to the model train-
ing. To capture the labeling confidence (relative import-
ance) of each candidate label, we propose to train the model
with the latent label distributions. Specifically, for a train-
ing example \( x_i \in \mathbb{R}^n \), its latent label distribution is de-
noted by \( p_i \in [0, 1]^l \). By arranging the labeling distributions
of \( m \) training examples, we form the label distribution ma-
trix \( P = [p_1^T, p_2^T, \cdots, p_m^T]^T \in [0, 1]^{m \times l} \). By substituting
\( Y \in [0, 1]^{m \times l} \) with \( P \in [0, 1]^{m \times l} \), we thus propose a novel
unified framework that estimates the latent label distributions
while training the model simultaneously:

\[
\min \sum_{i=1}^{m} L(x_i, p_i, f) + \lambda \Omega(f) + \mu \Psi(P) \tag{1}
\]

where \( L \) denotes the prescribed loss function, \( \Omega \) controls the
complexity of the model \( f \). \( \Psi \) aims to guarantee an accurate
estimation of the label distribution matrix \( P \), and \( \lambda, \mu \) are
parameters trading off these three terms.

Unlike the average-based framework and the identifica-
tion-based framework, our proposed framework naturally treats
the modeling outputs of the candidate labels in a discrimina-
tive manner due to the label distribution matrix \( P \), which can
indicate the different contributions of the candidate labels. To
optimally estimate \( P \), we assume it should have the following
property: local consistency, i.e., nearby (similar) instances are
supposed to have similar label distributions. Specifically,
if the \( i \)-th instance \( x_i \) is similar to the \( j \)-th instance \( x_j \),
their corresponding label distributions \( p_i \) and \( p_j \) should also
be similar. In order to characterize the similarity between in-
stances, we construct the similarity matrix \( S = [s_{ij}]_{m \times m} \)
by the symmetry-favored \( k \)-NN graph [Liu and Chang, 2009].
Specifically, \( s_{ij} = \exp(-||x_i - x_j||^2/\sigma^2) \) if \( j \in \mathcal{N}_i \),
or otherwise \( s_{ij} = 0 \). The set \( \mathcal{N}_i \) saves the indices of the \( k \)-
nearest neighbors of \( x_i \), and the parameter \( \sigma \) is defined by
\( \sigma = \sum_{i=1}^{m} ||x_i - x_{i_n}||/m \) where \( x_{i_n} \) denotes the \( k \)-th near-
nest neighbor of \( x_i \). To ensure that \( S \) is symmetric, we finally
set \( S = S + S^T \). In this way, we define \( \Psi(P) \) as follows:

\[
\Psi(P) = \sum_{i,j} s_{ij} \left( \frac{P_i}{\sqrt{d_{ii}}} - \frac{P_j}{\sqrt{d_{jj}}} \right)^2 \tag{2}
\]

s.t. \( \sum_{j} p_{ij} = 1, \quad \forall i \in [m] \)

\( \Theta_{m \times l} \leq P \leq Y \)

where \( d_{ii} = \sum_{j} s_{ij} \) is the degree of the vertex \( x_i \) in the graph
corresponding to the similarity matrix, and \( \Theta_{m \times l} \in \{0\}^{m \times l} \).
The first constraint formalizes the labeling confidence lev-
els of all the labels as label distributions. The second con-
straint guarantees that the ground-truth label is strictly in the
candidate label set, and the labeling confidence of each non-
candidate label must be 0. Following the above settings, we
propose to train the model by minimizing the least squares
loss of fitting the label distributions:

\[ L(x_i, p_i, f) = \|x_i W + b^T - p_i\|^2 \]  (3)

where \( W \in \mathbb{R}^{n \times l} \) and \( b \in \mathbb{R}^l \) are the model parameters. For the
regularization term to control the model complexity, we
adopt the widely-used squared Frobenius norm of \( W \):

\[ \Omega(f) = \|W\|_F^2 \]  (4)

Finally, to further facilitate a kernel extension for the general
nonlinear case, we present the formulation as a constrained
optimization problem:

\[
\begin{align*}
\min_{W,b,P} & \sum_i \|e_i\|^2_2 + \lambda \|W\|^2_F + \mu \sum_{i,j} s_{ij} \left( \frac{p_i}{\sqrt{d_{ii}}} - \frac{p_j}{\sqrt{d_{jj}}} \right)^2 \\
\text{s.t.} & \quad p_i = z_i W + b^T + e_i, \quad \forall i \in [m] \\
& \quad \sum_j p_{ij} = 1, \quad \forall i \in [m] \\
& \quad 0_{m \times l} \preceq P \preceq Y
\end{align*}
\]

where \( z_i = \phi(x_i) \) and \( \phi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^h \) is a feature mapping
that maps the feature space to some higher (maybe infinite)
dimensional Hilbert space with \( h \) dimensions.

4 Alternating Optimization

Obviously, the optimization problem (5) is a biconvex
problem [Gorski et al., 2007], and we solve this problem in an
alternating way. Specifically, we first optimize the objec-
tive function with respective to \( W \) and \( b \) when \( P \) is fixed,
and then optimize the objective function with respective to
\( P \) when \( W \) and \( b \) are both fixed. This procedure is repeated
until convergence or the maximum number of iterations is
reached.

Updating \( W \) and \( b \)

When \( P \in \mathbb{R}^{m \times l} \) is fixed, the optimization problem (5) with
respect to \( W \) and \( b \) can be stated as follows:

\[
\begin{align*}
\min_{W,b} & \quad \text{tr} (\Xi^T \Xi) + \lambda \text{tr} (W^T W) \\
\text{s.t.} & \quad P = ZW + 1_m b^T + \Xi \quad (6)
\end{align*}
\]

where \( \Xi = [e_1, e_2, \ldots, e_m]^T \in \mathbb{R}^{m \times l} \), \( \text{tr}(\cdot) \) is the trace
norm operator with the property \( \text{tr}(W^T W) = \|W\|^2_F \), and
\( 1_m = [1, 1, \ldots, 1]^T \in \mathbb{R}^m \). Then, the Lagrangian
of this problem can be expressed as:

\[
\begin{align*}
L(W, b, \Xi, A) &= \text{tr} (\Xi^T \Xi) + \lambda \text{tr} (W^T W) - \\
&\quad \text{tr} (A^T (ZW + 1_m b^T + \Xi - P))
\end{align*}
\]

where \( A = [a_1, a_2, \ldots, a_m]^T \in \mathbb{R}^{m \times l} \) is the matrix
that stores the Lagrange multipliers. In this way, the following
equations will be induced according to the KKT conditions:

\[
\begin{align*}
\frac{\partial L}{\partial \Xi} &= 0 \Rightarrow A = 2\Xi \Rightarrow \frac{\partial L}{\partial A} = 0 \Rightarrow ZW + 1_m b^T + \Xi = P \\
\frac{\partial L}{\partial W} &= 0 \Rightarrow W = \frac{1}{2\lambda} Z^T A \Rightarrow \frac{\partial L}{\partial b} = 0 \Rightarrow A^T 1_m = 0 \quad (8)
\end{align*}
\]

Above linear equations can be solved by following steps:

\[
\begin{align*}
ZW + 1_m b^T + \Xi &= P \\
\frac{1}{2\lambda} ZZ^T A + 1_m b^T + \frac{1}{2} A &= P
\end{align*}
\]

Here, we define the positive definite matrix \( H \) by
\( H = \frac{1}{\lambda} K + \frac{1}{\lambda} 1_m \times 1_m \) and \( K = ZZ^T \in \mathbb{R}^{m \times m} \) is given by its
elements \( k_{ij} = \text{exp}(\frac{\|x_i - x_j\|^2}{(2\sigma^2)}) \) is employed with \( \sigma \) set to the
average distance of all pairs of training examples. The ma-
trix \( 1_m \times 1_m \) is an identity matrix with \( m \) rows and \( m \) columns.
Then we can obtain:

\[
\begin{align*}
HA + 1_m b^T &= P \\
A + H^{-1} 1_m b^T &= H^{-1} P \\
1_m H^{-1} 1_m b^T &= 1_m H^{-1} P \\
b^T &= \frac{1_m H^{-1} P}{1_m H^{-1} 1_m}
\end{align*}
\]

For computational convenience, we define \( s = 1_m H^{-1} \in \mathbb{R}^{1 \times m} \), and the results are reported as follows:

\[
b^T = \frac{s P}{s 1_m} \\
A = H^{-1} (P - 1_m b^T)
\]

Updating \( P \)

When \( W \) and \( b \) are fixed, the modeling output matrix \( Q \in \mathbb{R}^{m \times l} \)
is denoted by \( Q = ZW + 1_m b^T \). By eliminating \( \Xi \), we can obtain:

\[
\begin{align*}
\min_P & \quad \|P - Q\|^2_2 + \mu \sum_{i,j} s_{ij} \left( \frac{p_i}{\sqrt{d_{ii}}} - \frac{p_j}{\sqrt{d_{jj}}} \right)^2 \\
\text{s.t.} & \quad \sum_j p_{ij} = 1, \quad \forall i \in [m] \\
& \quad 0_{m \times l} \preceq P \preceq Y
\end{align*}
\]

Here, this optimization problem is actually a constrained label
propagation problem [Zhou et al., 2004], where \( \mu \) specifies the relative amount of labeling information from the neighbor
points and the modeling outputs. The first constraint guaran-
tees that a label distribution is consistently assigned to each
instance in the process of label propagation. The second con-
straint guarantees that labels are only propagated among can-
didate labels. While in semi-supervised settings [Zhu and
Goldberg, 2009], labels are normally propagated from labeled
elements to unlabeled examples. In addition, traditional label
propagation problems normally treat the observed label
matrix \( Y \) as the initial label matrix. In contrast, since the
observed label matrix \( Y \) is a noise-corrupted version in parti-
cial label learning, we take the modeling output matrix \( Q \)
as the initial label matrix for each optimization iteration, there-
by adjusting the confidence level of each candidate label iter-
avtively. The optimization problem (12) can be reformulated
as a standard Quadratic Programming (QP) problem, which
can be solved by any off-the-shelf QP tools. The detailed
information is given in Appendix A.
Algorithm 1 The LALO Algorithm

Inputs:
\( D \): the PL training set \( D = \{ (X, Y) \} \)
\( k \): the number of nearest neighbors used for the similarity matrix
\( \lambda, \mu \): the parameters trading off each term in the loss function

Output:
y: the predicted label for the test example \( x \)

Process:
1. construct the similarity matrix by the symmetry-favored \( k \)-NN graph;
2. calculate the kernel matrix \( K = [K(x_i, x_j)]_{m \times m} \);
3. initialize \( P \) according to (13);
4. repeat
5. update \( b \) and \( A \) according to (11);
6. update \( Q = \frac{1}{K}KA + 1_m b^T \);
7. calculate \( \tilde{p} \) by solving (16) with a general QP procedure;
8. update \( P \) by reshaping \( \tilde{p} \in \mathbb{R}^{ml} \) into \( P \in \mathbb{R}^{m \times l} \);
9. until convergence or the maximum number of iterations;
10. return the predicted label \( y \) according to (14).

At the beginning of the alternating optimization, we initialize the label distribution matrix \( P = [p_{ij}]_{m \times l} \) as follows:

\[
p_{ij} = \begin{cases} 
\frac{1}{m} w_{ij}, & \text{if } y_{ij} = 1 \\
0, & \text{otherwise}
\end{cases} \quad (13)
\]

After \( \tilde{p} \) is figured out, we can easily obtain the label distribution matrix \( P \) by reshaping \( \tilde{p} \in \mathbb{R}^{ml} \) into \( P \in \mathbb{R}^{m \times l} \). After the completion of the optimization process, the predicted label \( y \) of the test example \( x \) by LALO is given as follows:

\[
y = \arg \max_{k \in [l]} \sum_{i=1}^{m} a_{ik}K(x, x_i) + b_k \quad (14)
\]

The pseudo code of LALO is presented in Algorithm 1. Since the proposed formulation (5) is biconvex, it can be solved by the alternating optimization method with guaranteed convergence [Gorski et al., 2007], and we set the maximum number of iterations as 50.

5 Experiments

5.1 Experimental Setup

In this section, we conduct extensive experiments on artificial (i.e., controlled UCI datasets) and real-world datasets to evaluate the performance of LALO. The main characteristics of these datasets are reported in Table 1.

Following the widely-used controlling protocol [Chen et al., 2014; Cour et al., 2011; Liu and Dietterich, 2012; Yu and Zhang, 2016; Zhang and Yu, 2015; Zhang et al., 2016; Zhang et al., 2017], each UCI dataset is controlled by three parameters \( p, r \) and \( \epsilon \) to generate artificial PL datasets. Here, \( p \) controls the proportion of training examples that are partially labeled, \( r \) controls the number of false positive labels within the candidate label set, and \( \epsilon \) controls the co-occurring probability of a specific false positive label and the ground-truth label.

In addition, we have also collected 5 real-world PL datasets\(^1\), including Soccer Player [Zeng et al., 2013], Lost [Cour et al., 2011], Yahoo! News [Guillaumin et al., 2010], FG-NET [Papanikolopoulos and Lanitis, 2014], and MSRC-v2 [Liu and Dietterich, 2012]. These real-world datasets come from several application domains. For automatic face naming (Lost, Soccer Player and Yahoo! News), each face cropped from an image or a video frame is considered as an instance and the names extracted from the corresponding image captions or video subtitles work as candidate labels. For objective classification (MSRC-v2), image segments are considered as instances and objects appearing in the same image work as candidate labels. For facial age estimation (FG-NET), each human face is represented as an instance, and the age annotations obtained by crowdsourcing are candidate labels. Besides, the average number of the candidate labels (Avg. CLs) for each real-world dataset is also recorded in Table 1.

The performance of LALO is compared with five state-of-the-art partial label learning algorithms, each configured with recommended parameters according to the respective literature:

- PL-KNN [Hullermeier and Beringer, 2006]: an \( k \)-nearest neighbor approach following the average-based framework. (Recommended configuration: \( k = 10 \)).
- CLPL [Cour et al., 2011]: a parametric approach following the average-based framework. (Recommended configuration: SVM with squared hinge loss).
- IPAL [Zhang and Yu, 2015]: an instance-based approach following the average-based framework. (Recommended configuration: \( \alpha = 0.95, k = 10, T = 100 \)).
- PL-SVM [Nguyen and Caruana, 2008]: a maximum margin approach following the identification-based framework. (Recommended configuration: regularization parameter pool with \{10\(^{-5}\), \ldots , 10\(^{8}\}\}).
- LSB-CMM [Liu and Dietterich, 2012]: a maximum likelihood approach following the identification-based framework. (Recommended configuration: 1 mixture component).

The parameters employed by LALO are set as \( k = 10, \lambda = 0.05, \mu = 0.005 \). The sensitivity analysis of LALO’s parameter configuration is conducted in Subsection 5.3. On each artificial and real-world dataset, ten runs of 50%/50% random train/test splits are performed, and the averaged accuracies (with standard deviations) are recorded for all algorithms. In addition, we use the \( t \)-test at 0.05 significance level for two independent samples to investigate whether LALO is significantly superior/inferior to the comparing algorithms for all experiments.

5.2 Experimental Results

Controlled UCI Datasets

Figure 1 reports the classification accuracy of each algorithm as \( \epsilon \) ranges from 0.1 to 0.7 with step size 0.1 when \( p \) and \( r \) are

\(^1\)These data sets are publicly available at: http://cse.seu.edu.cn/PersonalPage/zhangml/Resources.htm#partial_data
Table 1: Characteristics of the experimental datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>glass</th>
<th>usps</th>
<th>letter</th>
<th>deter</th>
<th>Lost</th>
<th>FG-NET</th>
<th>MSRCv2</th>
<th>Soccer Player</th>
<th>Yahoo! News</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>214</td>
<td>9298</td>
<td>20000</td>
<td>358</td>
<td>1122</td>
<td>1002</td>
<td>1758</td>
<td>17472</td>
<td>22991</td>
</tr>
<tr>
<td>Features</td>
<td>10</td>
<td>256</td>
<td>16</td>
<td>23</td>
<td>108</td>
<td>262</td>
<td>48</td>
<td>279</td>
<td>163</td>
</tr>
<tr>
<td>Classes</td>
<td>5</td>
<td>10</td>
<td>26</td>
<td>6</td>
<td>16</td>
<td>78</td>
<td>32</td>
<td>171</td>
<td>219</td>
</tr>
<tr>
<td>Avg. CLs</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2.23</td>
<td>7.48</td>
<td>3.16</td>
<td>2.09</td>
<td>1.91</td>
</tr>
</tbody>
</table>

Figure 1: Classification performance on controlled UCI datasets with \( \epsilon \) ranging from 0.1 to 0.7 (\( p = 1, r = 1 \)).

Figure 2: Classification performance on controlled UCI datasets with \( p \) ranging from 0.1 to 0.7 (\( r = 1 \)).

Both fixed at 1. For each ground-truth label \( y \in \mathcal{Y} \), one extra label \( y' \neq y \) is selected as the coupled label that co-occurs with \( y \) in the candidate label set with probability \( \epsilon \), and any other label is chosen to be the false positive label with the probability \( 1 - \epsilon \). Figure 2 reports the classification accuracy of each algorithm as \( p \) ranges from 0.1 to 0.7 with step size 0.1 when \( r \) is set to 1. In this setting, \( r \) labels are randomly selected as the false positive labels in the candidate label set for the PL examples. In addition, we also do experiments on controlled UCI datasets as \( p \) ranges from 0.1 to 0.7 with \( r \) set to 2 and 3. Due to the limited space, these results are not reported here\(^2\), while they are quite similar to that in Figure 2 (\( r = 1 \)).

As shown in Figure 1 to 2, LALO outperforms the comparing algorithms in most cases. Besides, the detailed win/tie/loss counts between LALO and other comparing algorithms are recorded in Table 2. Out of the 112 results (4 UCI datasets×28 configurations), we can find that LALO can achieve superior or at least comparable performance against all comparing algorithms in most cases, and lose to them in only a few cases.

### Real-World Datasets

The predictive accuracy of each algorithm on real-world datasets is recorded in Table 3. Note that the average number of candidate labels (Avg. CLs) of the dataset FG-NET is quite large, which causes an extremely low classification accuracy of each algorithm. For better evaluation of this facial age estimation task, two extra experiments are conducted on the dataset FG-NET where a test example is considered to be correctly classified if the difference between the predicted age and the ground-truth age is no more than 3 years.

---

\(^2\)Figures and code package for LALO are publicly available at: https://sites.google.com/site/ramber1995paper/publications
Table 3: Classification accuracy of each algorithm on the real-world datasets. Furthermore, ∗/o indicates whether LALO is statistically superior/inferior to the comparing algorithm.

Figure 3: Parameter sensitivity analysis of LALO on the real-world datasets Lost and MSRCv2.

(MAE3) or 5 years (MAE5). As shown in Table 3, it is obvious that LALO significantly outperforms all the counterpart algorithms on these real-world datasets except for CLPL on Lost, PL-SVM on Yahoo! News, and IPAL on MSRCv2 and Soccer Player, and LALO is never significantly outperformed by any comparing algorithm.

5.3 Sensitivity Analysis

We also study the sensitivity of LALO with respect to its three parameters λ, μ, and k. Figure 3 shows the performance of LALO under different parameter configurations. From Figure 3, we can easily find that the parameter configuration specified for LALO in Subsection 5.1 (λ = 0.05, μ = 0.005, k = 10) naturally follows the sensitivity curves. In addition, Figure 3 also reports the difference of the label distribution by dij = P... ... ... ... ...

6 Conclusion

In this paper, we propose a novel unified partial label learning framework and present a biconvex formulation to leverage the latent label distributions for the model training. Extensive experimental results validate the effectiveness of the proposed approach named LALO. Since LALO serves as a bridge between the model training and label propagation, this work can be naturally extended to inductive semi-supervised learning based on label propagation. Besides, it is also interesting to exploit the consistency of the feature space and the label space in other manners.

Acknowledgements

The authors want to thank Prof. Min-Ling Zhang for help on the controlled UCI datasets and the comparing algorithms. This work was supported by MOE, NRF, and NTU.

A Quadratic Programming Formulation

To solve the problem (12), we set \( \mathbf{\tilde{p}} = \text{vec}(\mathbf{P}) \in [0,1]^{ml} \) where vec(·) is the vectorization operator. Likewise, \( \mathbf{\tilde{q}} = \text{vec}(\mathbf{Q}) \in \mathbb{R}^{ml} \) and \( \mathbf{\tilde{y}} = \text{vec}(\mathbf{Y}) \in \{0,1\}^{ml} \). To deal with the equality constraint using \( \mathbf{\tilde{p}} \), we pick up the indices of \( \mathbf{\tilde{p}} \) by defining a set \( C = \{C_0,C_1,\ldots,C_{m-1}\} \) as follows:

\[
j \in C_i \quad \text{if} \quad \tilde{x}_i \neq m, \quad \forall j \in [m] \quad (15)
\]

Using these notations, the problem (12) can be written as:

\[
\begin{align*}
& \min_{\mathbf{p}} \quad \frac{1}{2} \mathbf{H} \mathbf{p} + \mathbf{f}^T \mathbf{p} \\
& \text{s.t.} \quad \sum_{j \in C_i} \tilde{p}_j = 1, \quad \forall C_i \subseteq C \\
& \quad 0_{ml} \leq \mathbf{p} \leq \mathbf{\tilde{y}}
\end{align*}
\]

where \( \mathbf{f} = -2\mathbf{q} \) and \( \mathbf{H} \in \mathbb{R}^{ml \times ml} \) is defined as follows:

\[
\mathbf{H} = \begin{bmatrix}
\mathbf{T} & 0_{m \times m} & \cdots & 0_{m \times m} \\
0_{m \times m} & \mathbf{T} & \cdots & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
0_{m \times m} & \cdots & 0_{m \times m} & \mathbf{T}
\end{bmatrix} \quad (17)
\]

Here, \( \mathbf{T} \) is a square matrix defined by \( \mathbf{T} = 2((\mu + 1)\mathbf{I}_{m \times m} - \mu\mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}) \in \mathbb{R}^{m \times m} \) where \( \mathbf{D} \) is a diagonal matrix with its diagonal element defined by \( d_{ii} = \sum_j s_{ij} \). In this way, the optimization problem (16) can be efficiently solved by any off-the-shelf QP toolbox.
References


