Supplementary Material for
“Progressive Identification of True Labels for Partial-Label Learning”

Jiaqi Lv† 1  Miao Xu2,3  Lei Feng4  Gang Niu2  Xin Geng1  Masashi Sugiyama2,5

A. Proof of Lemma 2

Cross-Entropy loss According to (Masnadi-Shirazi & Vasconcelos, 2009), since the $\ell_{CE}$ is non-negative, minimizing the conditional risk $\mathbb{E}_{p(y|x)}[\ell_{CE}(g(X), Y)|X]$, $\forall X \in \mathcal{X}$ is an alternative of minimizing $\mathcal{R}(g)$. The conditional risk can be written as

$$\mathcal{C}(g) = - \sum_{i=1}^{c} p(Y = i|X) \log(g_i(X)), \quad \text{s.t.} \quad \sum_{i=1}^{c} g_i(X) = 1.$$ 

By the Lagrange Multiplier method (Bertsekas, 1997), we have

$$\mathcal{L} = - \sum_{i=1}^{c} p(Y = i|X) \log(g_i(X)) + \lambda(\sum_{i=1}^{c} g_i(X) - 1).$$

To minimize $\mathcal{L}$, we take the partial derivative of $\mathcal{L}$ with respect to $g_i$ and set it be 0:

$$g^*_i(X) = \frac{1}{\lambda} p(Y = i|X).$$

Because $\sum_{i=1}^{c} g^*_i(X) = 1$ and $\sum_{i=1}^{c} p(Y = i|X) = 1$, we have

$$\sum_{i=1}^{c} g^*_i(X) = \frac{1}{\lambda} \sum_{i=1}^{c} p(Y = i|X) = 1.$$ 

Therefore, we can obtain $\lambda = 1$ that ensures $g^*_i(X) = p(Y = i|X), \forall i \in [c], \forall X \in \mathcal{X}$, which concludes the proof.

Mean squared error loss Analogously, if the mean squared error loss is used, we can write the optimization problem as

$$\mathcal{C}(g) = \sum_{i=1}^{c} (p(Y = i|X) - g_i(X))^2, \quad \text{s.t.} \quad \sum_{i=1}^{c} g_i(X) = 1.$$ 

By the Lagrange Multiplier method, we have

$$\mathcal{L} = \sum_{i=1}^{c} (p(Y = i|X) - g_i(X))^2 - \lambda'(\sum_{i=1}^{c} g_i(X) - 1).$$

By setting the derivative to 0, we obtain

$$g^*_i(X) = \frac{\lambda'}{2} + p(Y = i|X).$$

†Preliminary work was done during an internship at RIKEN AIP. 1School of Computer Science and Engineering, Southeast University, Nanjing, China 2RIKEN Center for Advanced Intelligence Project, Tokyo, Japan 3The University of Queensland, Australia 4School of Computer Science and Engineering, Nanyang Technological University, Singapore 5University of Tokyo, Tokyo, Japan. Correspondence to: Xin Geng <xgeng@seu.edu.cn>.

Because $\sum_{i=1}^{c} g_i^*(X) = 1$ and $\sum_{i=1}^{c} g_i^*(X) = 1$, we have

$$\sum_{i=1}^{c} g_i^*(X) = \frac{\lambda' c}{2} + \sum_{i=1}^{c} p(Y = i|X).$$

Since $c \neq 0$, we can obtain $\lambda' = 0$. In this way, $g_i^*(X) = p(Y = i|X), \forall i \in [c], \forall X \in X'$, which concludes the proof. \qed

B. Proof of Theorem 1

First we prove $g^*$ is the optimal classifier for PLL by substituting the $g^*$ into the PLL risk estimator Eq. (5):

$$R_{PLL}(g^*) = \mathbb{E}_{(X,S) \sim p(x,s)}[\min_{i \in S} \ell(g^*(X), e^i)] = \int \sum_{S \in \mathcal{S}} \min_{i \in S} \ell(g^*(X), e^i)p(s|x)p(x)dX$$

$$= \int \sum_{S \in \mathcal{S}} \min_{i \in \mathcal{Y}} \ell(g^*(X), e^i)p(y|x)p(y|x)p(x)dX$$

$$= \int \sum_{Y \in \mathcal{Y}} \sum_{S \in \mathcal{S}} \ell(g^*(X), e^{Y_X})p(s|x)p(y|x)p(x)dX$$

$$= \int \sum_{Y \in \mathcal{Y}} \ell(g^*(X), e^{Y_x})p(x,y)dX = R(g^*) = 0.$$

where we have used $\min_{i \in \mathcal{S}} \ell(g^*(X), e^i) = \ell(g^*(X), e^{Y_X})$ because $\ell$ is a proper loss and the derministic assumption is made. This indicates that the PLL risk has been minimized by $g^*$.

On the other hand, we prove $g^*$ is the only solution to Eq. (5) by contradiction, namely, there is at least one other solution $h$ enables $R_{PLL}(h) = 0$, and predicts different label $Y^h \neq Y^i$ for at least one instance $X$. Hence for any $S \ni Y^h$ we have

$$\min_{i \in \mathcal{S}} \ell(h(X), e^i) = \ell(h(X), e^{Y^h}) = 0.$$

Nevertheless, the above equality is always true unless $Y^h$ is invariably included in the candidate label set of $X$, i.e., $\Pr_{S \sim p(s|x,y)}(Y^h \in S) = 1.$ Obviously, this contradicts the small ambiguity degree condition. Therefore, there is one, and only one minimizer of the PLL risk estimator, which is the same as the minimizer learned from ordinarily labeled data. The proof is complete. \qed

C. Proof of Theorem 2

First, we show the uniform deviation bound, which is useful to derive the estimation error bound.

Lemma 3. For any $\delta > 0$, we have with probability at least $1 - \delta$,

$$\sup_{g \in \mathcal{G}} \left| R_{PLL}(g) - \hat{R}_{PLL}(g) \right| \leq 2M \min_{\ell}(\ell_{PLL} \circ \mathcal{G}) + M \sqrt{\frac{\log(2/\delta)}{2n}}.$$

Proof. Consider the one-side uniform deviation $\sup_{g \in \mathcal{G}} R_{PLL}(g) - \hat{R}_{PLL}(g)$. Since the loss function $\ell$ is upper-bounded by $M$, the change of it will be no more than $M/n$ after replacing some $x$. Then, by McDiarmid’s inequality (McDiarmid, 1989), for any $\delta > 0$, with probability at least $1 - \delta/2$, the following holds:

$$\sup_{g \in \mathcal{G}} R_{PLL}(g) - \hat{R}_{PLL}(g) \leq \mathbb{E} \left[ \sup_{g \in \mathcal{G}} R_{PLL}(g) - \hat{R}_{PLL}(g) \right] + M \sqrt{\frac{\log(2/\delta)}{2n}}.$$
By symmetrization (Vapnik, 1998), it is a routine work to show that

$$\mathbb{E} \left[ \sup_{g \in \mathcal{G}} R_{PLL}(g) - \hat{R}_{PLL}(g) \right] \leq 2\mathcal{R}_n(\ell_{PLL} \circ \mathcal{G}).$$

The one-side uniform deviation $\sup_{g \in \mathcal{G}} \hat{R}_{PLL}(g) - R_{PLL}(g)$ can be bounded similarly. □

Then we upper bound $\mathcal{R}_n(\ell_{PLL} \circ \mathcal{G})$.

**Lemma 4.** Suppose $\ell_{PLL}$ is defined as Eq. (4), it holds that

$$\mathcal{R}_n(\ell_{PLL} \circ \mathcal{G}) \leq c\mathcal{R}_n(\ell \circ \mathcal{G}) \leq \sqrt{2c}L \ell \sum_{y=1}^{c} \mathcal{R}_n(G_y).$$

**Proof.** By definition of $\ell_{PLL}$, $\ell_{PLL} \circ \mathcal{G}(x, S_i) = \min_{y \in \mathcal{S}} \ell \circ \mathcal{G}(x, y) = \min_{y \in [c]} \ell \circ \mathcal{G}(x, y)$. Given sample sized $n$, we first prove the result in the case $c = 2$. The min operator can be written as

$$\min\{z_1, z_2\} = \frac{1}{2}[z_1 + z_2 - |z_1 - z_2|].$$

In this way, we can write

$$\mathcal{R}_n(\ell_{PLL} \circ \mathcal{G}) = \mathbb{E}_\sigma \left[ \sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \ell(g(x_i), s_i) \right] = \mathbb{E}_\sigma \left[ \sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \min\{\ell(g(x_i), y_1), \ell(g(x_i), y_2)\} \right]

= \mathbb{E}_\sigma \left[ \sup_{g \in \mathcal{G}} \frac{1}{2n} \sum_{i=1}^{n} \sigma_i \left[ \ell(g(x_i), y_1) + \ell(g(x_i), y_2) - |\ell(g(x_i), y_1) - \ell(g(x_i), y_2)| \right] \right]

\leq \mathbb{E}_\sigma \left[ \sup_{g \in \mathcal{G}} \frac{1}{2n} \sum_{i=1}^{n} \sigma_i \ell(g(x_i), y_1) \right] + \mathbb{E}_\sigma \left[ \sup_{g \in \mathcal{G}} \frac{1}{2n} \sum_{i=1}^{n} \sigma_i \ell(g(x_i), y_2) \right]

+ \mathbb{E}_\sigma \left[ \sup_{g \in \mathcal{G}} \frac{1}{2n} \sum_{i=1}^{n} \sigma_i \ell(g(x_i), y_1) - \ell(g(x_i), y_2) \right]

= \frac{1}{2} \left( \mathcal{R}_n(\ell \circ \mathcal{G}) + \mathcal{R}_n(\ell \circ \mathcal{G}) \right) + \mathbb{E}_\sigma \left[ \sup_{g \in \mathcal{G}} \frac{1}{2n} \sum_{i=1}^{n} \sigma_i |\ell(g(x_i), y_1) - \ell(g(x_i), y_2)| \right].$$

Since $x \mapsto |x|$ is a 1-Lipschitz function, by Talagrand’s contraction lemma (Ledoux & Talagrand, 2013), the last term can be bounded:

$$\mathbb{E}_\sigma \left[ \sup_{g \in \mathcal{G}} \frac{1}{2n} \sum_{i=1}^{n} \sigma_i |\ell(g(x_i), y_1) - \ell(g(x_i), y_2)| \right] \leq \mathbb{E}_\sigma \left[ \sup_{g \in \mathcal{G}} \frac{1}{2n} \sum_{i=1}^{n} \sigma_i (\ell(g(x_i), y_1) - \ell(g(x_i), y_2)) \right] \leq \frac{1}{2} \left( \mathcal{R}_n(\ell \circ \mathcal{G}) + \mathcal{R}_n(\ell \circ \mathcal{G}) \right).$$

Combining Eq. (11) and Eq. (12) yields $\mathcal{R}_n(\ell_{PLL} \circ \mathcal{G}) \leq \mathcal{R}_n(\ell \circ \mathcal{G}) \leq \mathcal{R}_n(\ell \circ \mathcal{G})$. The general case can be derived from the case $c = 2$ using $\min\{z_1, \ldots, z_c\} = \min\{z_1, \min\{z_2, \ldots, z_c\}\}$ and an immediate recurrence.

Then we apply the Rademacher vector contraction inequality (Maurer, 2016),

$$\mathcal{R}_n(\ell \circ \mathcal{G}) \leq \sqrt{2L} \ell \sum_{y=1}^{c} \mathcal{R}_n(G_y).$$

The proof is completed. □
Based on Lemma 3 and 4, the estimation error bound Eq. (7) is proven through

\[
\mathcal{R}_{PLL}(\hat{g}_{PLL}) - \mathcal{R}_{PLL}(g^*_{PLL}) = \left( \mathcal{R}_{PLL}(\hat{g}_{PLL}) - \hat{\mathcal{R}}_{PLL}(\hat{g}_{PLL}) \right) + \left( \hat{\mathcal{R}}_{PLL}(\hat{g}_{PLL}) - \hat{\mathcal{R}}_{PLL}(g^*_{PLL}) \right) \\
\leq \left( \mathcal{R}_{PLL}(\hat{g}_{PLL}) - \mathcal{R}_{PLL}(g^*_{PLL}) \right) + \left( \hat{\mathcal{R}}_{PLL}(\hat{g}_{PLL}) - \mathcal{R}_{PLL}(g^*_{PLL}) \right) \\
\leq 2 \sup_{g \in \mathcal{G}} \left| \mathcal{R}_{PLL}(g) - \hat{\mathcal{R}}_{PLL}(g) \right| \\
\leq 4\sqrt{2}cL \ell \sum_{y=1}^{c} \mathcal{R}_n(\mathcal{G}_y) + 2M \sqrt{\frac{\log(2/\delta)}{2n}}.
\]

\[\hat{R}_{PLL}(\hat{g}_{PLL}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{n} \sum_{j=1}^{c} \ell_{CE}(g_j(x_i), e_i^{s_j}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} \ell_{CE}(g_j(x_i), w_{ij}e_i^{s_j}).
\]

Thus, the weights \( w \) can be moved into the loss function and yields:

\[
\hat{R}_{PLL} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} \ell_{CE}(g_j(x_i), z_{ij}) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} \ell_{CE}(g_j(x_i), e_i^{s_j}).
\]

where \( z_{ij} = w_{ij}I(j \in s_i) \). The optimal \( g^* \) of Eq. (14) is essentially equivalent to \( g^* \) learned from Eq. (13). Therefore, our method is a strict extension of (Jin & Ghahramani, 2003).

**E. Benchmark Datasets**

**E.1. Setup**

Table 4 describes the benchmark datasets and the corresponding models of them.
MNIST  This is a grayscale image dataset of handwritten digits from 0 to 9 where the size of the images is 28 × 28.

The linear model is a linear-in-input model:  \( \cdot \)-10, and MLP refers to a 5-layer FC with ReLU as the activation function:  \( \cdot \)-300-300-300-300-10. Batch normalization (Ioffe & Szegedy, 2015) was applied before hidden layers. For both models, the softmax function was applied to the output layer, and \( \ell_2 \)-regularization was added. The two models were trained by SGD with the default momentum parameter (\( \beta = 0.9 \)), and the batch size was set to 256.

Fashion-MNIST  This is a grayscale image dataset similarly to MNIST. In Fashion-MNIST, each instance is a 28 × 28 grayscale image and associated with a label from 10 fashion item classes. The models and optimizer were the same as MNIST.

Kuzushiji-MNIST  This is another grayscale image dataset similarly to MNIST. In Kuzushiji-MNIST, each instance is a 28 × 28 grayscale image and associated with a label from 10 cursive Japanese (Kuzushiji) characters. The models and optimizer were the same as MNIST.

CIFAR-10  This dataset consists of 60,000 32 × 32 × 3 colored image in RGB format in 10 classes.

The detailed architecture of ConvNet (Laine & Aila, 2017) is as follows.

\[
\begin{align*}
0\text{th (input) layer:} & \quad (32*32*3) - \\
1\text{st to 4th layers:} & \quad [C(3*3, 128)]*3 - \text{Max Pooling} - \\
5\text{th to 8th layers:} & \quad [C(3*3, 256)]*3 - \text{Max Pooling} - \\
9\text{th to 11th layers:} & \quad C(3*3, 512) - C(3*3, 256) - C(3*3, 128) - \\
12\text{th layers:} & \quad \text{Average Pooling-10}
\end{align*}
\]

where C(3*3, 128) means 128 channels of 3×3 convolutions followed by Leaky-ReLU (LReLU) active function (Maas et al., 2013). [· · ·] means 3 such layers, etc.

The detailed architecture of ResNet (He et al., 2016) was as follows.

\[
\begin{align*}
0\text{th (input) layer:} & \quad (32*32*3) - \\
1\text{st to 11th layers:} & \quad C(3*3, 16) - [C(3*3, 16), C(3*3, 16)]*5 - \\
12\text{th to 21st layers:} & \quad [C(3*3, 32), C(3*3, 32)]*5 - \\
22\text{nd to 31st layers:} & \quad [C(3*3, 64), C(3*3, 64)]*5 - \\
32\text{nd layer:} & \quad \text{Average Pooling-10}
\end{align*}
\]

where [· · ·] means a building block (He et al., 2016). These two models were trained by SGD with the default momentum parameter and the batch size was 256.

An example of a binomial flipping with \( q = 0.1 \) and of a pair flipping with \( q = 0.5 \) used on MNIST are below, respectively:

\[
\begin{bmatrix}
1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1
\end{bmatrix} \quad \begin{bmatrix}
1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.1 & 1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.1 & 0.1 & 1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 1 & 0.1 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 1 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 1
\end{bmatrix} \quad \begin{bmatrix}
0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

E.2. Transductive Results

Figure 3 illustrates the transductive results on the benchmark datasets, i.e., the ability in identifying the true labels in the training set. We can see that PRODEN has a strong ability to find the true labels.

E.3. Test Results in the Pair Case

Figure 4 illustrates the test results on the benchmark datasets in the pair case. They show a similar phenomenon to Figure 2 that PRODEN is affected slightly and CCN is affected severely when the ambiguity degree goes large.
F. UCI datasets

F.1. Characteristic of the UCI Datasets and Setup

Table 5 summaries the characteristic of the UCI datasets. We normalized these dataset by the Z-scores by convention and use the linear model trained by SGD with momentum 0.9.

F.2. Comparing Methods

The comparing PLL methods are listed as follows.

- **SURE (Feng & An, 2019)**: an iterative EM-based method [suggested configuration: $\lambda, \beta \in \mathbb{R}$]
Table 6 provides additional experiments to investigate the performances of each comparing methods on the UCI datasets with the pair flipping strategy. It shows that PRODEN generally achieves superior performance against other parametric and non-parametric methods. Our advantage is less obvious compared with the non-parametric method IPAL, whereas the performance of PRODEN could be easily increased by employing a deeper network.

F.3. Results

Tabel 6 provides additional experiments to investigate the performances of each comparing methods on the UCI datasets with the pair flipping strategy. It shows that PRODEN generally achieves superior performance against other parametric and non-parametric methods. Our advantage is less obvious compared with the non-parametric method IPAL, whereas the performance of PRODEN could be easily increased by employing a deeper network.
Supplementary Material

Table 5. Summary of UCI datasets and models.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Train</th>
<th># Test</th>
<th># Feature</th>
<th># Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast</td>
<td>1,335</td>
<td>149</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>Texture</td>
<td>4,950</td>
<td>550</td>
<td>40</td>
<td>11</td>
</tr>
<tr>
<td>Dermatology</td>
<td>329</td>
<td>37</td>
<td>34</td>
<td>6</td>
</tr>
<tr>
<td>Synthetic-Control</td>
<td>540</td>
<td>60</td>
<td>60</td>
<td>6</td>
</tr>
<tr>
<td>20Newsgroups</td>
<td>16,961</td>
<td>1,885</td>
<td>300</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 6. Test accuracy (mean ± std) on the UCI datasets in the pair case.

<table>
<thead>
<tr>
<th>q</th>
<th>Yeast</th>
<th>Texture</th>
<th>Dermatology</th>
<th>Synthetic Control</th>
<th>20Newsgroups</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>56.38±4.71%</td>
<td>99.71±0.12%</td>
<td>96.16±3.27%</td>
<td>98.05±1.58%</td>
<td>78.05±0.97%</td>
</tr>
<tr>
<td>0.9</td>
<td>44.03±4.12%</td>
<td>99.35±0.31%</td>
<td>93.68±6.80%</td>
<td>96.33±1.12%</td>
<td>70.71±0.72%</td>
</tr>
<tr>
<td>0.5</td>
<td>56.26±4.74%</td>
<td>99.13±0.40%</td>
<td>93.15±2.56%</td>
<td>87.40±7.25%</td>
<td>76.90±0.92%</td>
</tr>
<tr>
<td>0.9</td>
<td>42.62±5.52%</td>
<td>78.74±4.09%</td>
<td>68.14±6.89%</td>
<td>57.90±5.08%</td>
<td>57.15±0.71%</td>
</tr>
<tr>
<td>0.5</td>
<td>24.39±4.40%</td>
<td>94.25±0.48%</td>
<td>73.81±6.18%</td>
<td>64.68±3.08%</td>
<td>63.68±0.67%</td>
</tr>
<tr>
<td>0.9</td>
<td>16.50±3.43%</td>
<td>61.85±2.29%</td>
<td>48.03±11.42%</td>
<td>37.15±6.91%</td>
<td>45.86±0.95%</td>
</tr>
<tr>
<td>0.5</td>
<td>56.38±4.71%</td>
<td>98.71±0.28%</td>
<td>95.89±3.75%</td>
<td>78.87±11.94%</td>
<td>74.38±0.90%</td>
</tr>
<tr>
<td>0.9</td>
<td>42.62±5.52%</td>
<td>78.74±4.09%</td>
<td>68.14±6.89%</td>
<td>57.90±5.08%</td>
<td>57.15±0.71%</td>
</tr>
<tr>
<td>0.5</td>
<td>24.39±4.40%</td>
<td>94.25±0.48%</td>
<td>73.81±6.18%</td>
<td>64.68±3.08%</td>
<td>63.68±0.67%</td>
</tr>
<tr>
<td>0.9</td>
<td>16.50±3.43%</td>
<td>61.85±2.29%</td>
<td>48.03±11.42%</td>
<td>37.15±6.91%</td>
<td>45.86±0.95%</td>
</tr>
<tr>
<td>0.5</td>
<td>56.38±4.71%</td>
<td>98.71±0.28%</td>
<td>95.89±3.75%</td>
<td>78.87±11.94%</td>
<td>74.38±0.90%</td>
</tr>
<tr>
<td>0.9</td>
<td>42.62±5.52%</td>
<td>78.74±4.09%</td>
<td>68.14±6.89%</td>
<td>57.90±5.08%</td>
<td>57.15±0.71%</td>
</tr>
<tr>
<td>0.5</td>
<td>24.39±4.40%</td>
<td>94.25±0.48%</td>
<td>73.81±6.18%</td>
<td>64.68±3.08%</td>
<td>63.68±0.67%</td>
</tr>
<tr>
<td>0.9</td>
<td>16.50±3.43%</td>
<td>61.85±2.29%</td>
<td>48.03±11.42%</td>
<td>37.15±6.91%</td>
<td>45.86±0.95%</td>
</tr>
<tr>
<td>0.5</td>
<td>56.38±4.71%</td>
<td>98.71±0.28%</td>
<td>95.89±3.75%</td>
<td>78.87±11.94%</td>
<td>74.38±0.90%</td>
</tr>
<tr>
<td>0.9</td>
<td>42.62±5.52%</td>
<td>78.74±4.09%</td>
<td>68.14±6.89%</td>
<td>57.90±5.08%</td>
<td>57.15±0.71%</td>
</tr>
<tr>
<td>0.5</td>
<td>24.39±4.40%</td>
<td>94.25±0.48%</td>
<td>73.81±6.18%</td>
<td>64.68±3.08%</td>
<td>63.68±0.67%</td>
</tr>
<tr>
<td>0.9</td>
<td>16.50±3.43%</td>
<td>61.85±2.29%</td>
<td>48.03±11.42%</td>
<td>37.15±6.91%</td>
<td>45.86±0.95%</td>
</tr>
<tr>
<td>0.5</td>
<td>56.38±4.71%</td>
<td>98.71±0.28%</td>
<td>95.89±3.75%</td>
<td>78.87±11.94%</td>
<td>74.38±0.90%</td>
</tr>
<tr>
<td>0.9</td>
<td>42.62±5.52%</td>
<td>78.74±4.09%</td>
<td>68.14±6.89%</td>
<td>57.90±5.08%</td>
<td>57.15±0.71%</td>
</tr>
</tbody>
</table>

G. Characteristic of the Real-world Datasets and Setup

Table 7 summarizes the characteristic of the real-world datasets and the corresponding models. The preprocessing, model and optimizer were the same as UCI datasets.
Table 7. Summary of real-world partial label datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Examples</th>
<th># Feature</th>
<th># Class</th>
<th># Avg. CLs</th>
<th>Task Domain</th>
<th>Model g(x; Θ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lost</td>
<td>1122</td>
<td>108</td>
<td>16</td>
<td>2.23</td>
<td>automatic face naming (Panis &amp; Lanitis, 2014)</td>
<td>Linear model</td>
</tr>
<tr>
<td>BirdSong</td>
<td>4998</td>
<td>38</td>
<td>13</td>
<td>2.18</td>
<td>bird song classification (Briggs et al., 2012)</td>
<td>Linear model</td>
</tr>
<tr>
<td>MSRCv2</td>
<td>1758</td>
<td>48</td>
<td>23</td>
<td>3.16</td>
<td>object classification (Liu &amp; Dietterich, 2012)</td>
<td>Linear model</td>
</tr>
<tr>
<td>Soccer Player</td>
<td>17472</td>
<td>279</td>
<td>171</td>
<td>2.09</td>
<td>automatic face naming (Zeng et al., 2013)</td>
<td>Linear model</td>
</tr>
<tr>
<td>Yahoo! News</td>
<td>22991</td>
<td>163</td>
<td>219</td>
<td>1.91</td>
<td>automatic face naming (Guillaumin et al., 2010)</td>
<td>Linear model</td>
</tr>
</tbody>
</table>

References


