Partial Label Learning via Label Enhancement

Ning Xu, Jiaqi Lv, Xin Geng*
MOE Key Laboratory of Computer Network and Information Integration, China
School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
{xning, lvjiaqi, xgeng}@seu.edu.cn

Abstract

Partial label learning aims to learn from training examples each associated with a set of candidate labels, among which only one label is valid for the training example. The common strategy to induce predictive model is trying to disambiguate the candidate label set, such as disambiguation by identifying the ground-truth label iteratively or disambiguation by treating each candidate label equally. Nonetheless, these strategies ignore considering the generalized *label distribution* corresponding to each instance since the generalized label distribution is not explicitly available in the training set. In this paper, a new partial label learning strategy named PL-LE is proposed to learn from partial label examples via *label enhancement*. Specifically, the generalized label distributions are recovered by leveraging the topological information of the feature space. After that, a multi-class predictive model is learned by fitting a regularized multi-output regressor with the generalized label distributions. Extensive experiments show that PL-LE performs favorably against state-of-the-art partial label learning approaches.

Introduction

Partial label (PL) learning deals with the problem where each training example is associated with a set of candidate labels, among which only one label is valid (Cour, Sapp, and Taskar 2011; Chen et al. 2014; Yu and Zhang 2017). In recent years, partial label learning techniques have been found useful in solving many real-world scenarios such as web mining (Jie and Orabona 2010), multimedia content analysis (Zeng et al. 2013; Chen, Patel, and Chellappa in press), ecoinformatics (Liu and Dietterich 2012; Tang and Zhang 2017), etc.

Formally speaking, let $\mathcal{X} = \mathbb{R}^q$ be the q-dimensional instance space and $\mathcal{Y} = \{y_1, y_2, y_3, ..., y_c\}$ be the label space with c class labels. Given the partial label training set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i)|1 \leq i \leq n\}$, the task of partial label learning is to induce a multi-class classifier $f : \mathcal{X} \mapsto \mathcal{Y}$ from $\mathcal{D}$. Here, $\boldsymbol{x}_i \in \mathcal{X}$ is a q-dimensional feature vector and $S_i \subseteq \mathcal{Y}$ is the associated candidate label set. Partial label learning takes the key assumption that the ground-truth label y_i corresponding to $\boldsymbol{x}_i$ resides in its candidate label set S_i and therefore cannot be directly accessed by the learning algorithm.

Intuitively, the basic strategy for handling partial label learning problem is disambiguation, i.e., trying to identify the ground-truth label from the candidate label set associated with each training example, where existing strategies include disambiguation by identification or disambiguation by averaging. For identification-based disambiguation, the ground-truth label is regarded as latent variable and identified through iterative refining procedure such as EM (Jin and Ghahramani 2003; Nguyen and Caruana 2008; Liu and Dietterich 2012; Chen et al. 2014; Yu and Zhang 2017). For averaging-based disambiguation, all the candidate labels are treated equally and the prediction is made by averaging their modeling outputs (Hüllermeier and Beringer 2006; Cour, Sapp, and Taskar 2011; Zhang and Yu 2015).

In order to handle partial label learning problem, we can explicitly assign a *description degree* to each label instead of disambiguation. This is similar to *label distribution learning* (LDL) (Geng 2016). In LDL, the description degrees $d_{\boldsymbol{x}}^{y_j}$ of all the labels constitute a real-valued vector called *label distribution*. Here $d_{\boldsymbol{x}}^{y_j} \in [0, 1]$ and $\sum_y d_{\boldsymbol{x}}^{y} = 1$. Note that the normalized labeling confidence vector in the feature-ware PL approach (Zhang, Zhou, and Liu 2016) can be viewed as label distribution. In order to accommodate more flexibility on PL data sets, the description degree is generalized in this paper: 1) $d_{\boldsymbol{x}}^{y_j} \in (0, 1), \forall\, y_j \in S_i$ denotes the label relevance over each candidate label. 2) $d_{\boldsymbol{x}}^{y_j} \in (-1, 0), \forall\, y_j \notin S_i$ denotes the label irrelevance over each non-candidate label. Then, the generalized description degrees (GDD) of all the labels constitute the generalized label distribution (GLD).

GDDs in partial label learning are essentially relative in mainly two aspects:

- The relevance among candidate labels is different rather than exactly equal. For example, in Figure 1(a), candidate painting style can be freely provided by web users, while the relevance of each style is different.
- The irrelevance of each non-candidate label may be very different. For example, in Figure 1(b), for a car, the label airplane is more irrelevant than the label tank.

However, GLD is not explicitly available in the training sets. It needs to be somehow recovered from the training set, a process which is named as *label enhancement* (LE)

*Corresponding author

(a) Online object annotation

(b) Target

Figure 1: Two examples about the generalized description degrees in partial label learning.

(Xu, Tao, and Geng 2018). Accordingly, a novel partial label learning algorithm named PL-LE, i.e., *Partial Label learning via Label Enhancement*, is proposed in this paper. PL-LE can recover flexible GLDs via leveraging the topological information of the feature space. After that, a multi-class predictive model is learned by fitting a regularized multi-output regressor with the recovered GLDs.

The rest of this paper is organized as follows. Firstly, related works on partial label learning are briefly reviewed. Secondly, technical details of the proposed approach are introduced. Thirdly, the results of the comparative experiments are reported. Finally, we conclude this paper.

Related Work

As shown in Section 1, supervision information conveyed by PL training examples is implicit as the ground-truth label is hidden within the candidate label set. Therefore, partial label learning can be regarded as a *weak supervision* learning framework with implicit labeling information. Generally, partial label learning is related to several well-established weakly-supervised learning frameworks such as *semi-supervised learning*, *multi-instance learning* and *multi-label learning*. Nevertheless, the type of weak supervision information handled by partial label learning is different to those counterpart frameworks.

In *semi-supervised learning* (Zhu and Goldberg 2009), the task is to learn a classifier $f : \mathcal{X} \mapsto \mathcal{Y}$ from both labeled and unlabeled examples. For unlabeled data the ground-truth label assumes the entire label space, while for PL data the ground-truth label is confined within its candidate label set. In *multi-instance learning* (Amores 2013), the task is to learn a classifier $f : 2^{\mathcal{X}} \mapsto \mathcal{Y}$ from examples each represented as a labeled bag of instances, where a single label is assigned to a set of instances for multi-instance example while a set of labels are assigned to a single instance for PL example. In *multi-label learning* (Zhang and Zhou 2014; Hou, Geng, and Zhang 2016), the task is to learn a classifier $f : \mathcal{X} \mapsto 2^{\mathcal{Y}}$ from training examples each associated with multiple labels, where the associated labels are all valid ones for multi-label example while the associated labels are only candidate ones for PL example.

Most existing algorithms aim to fulfill the learning task by fitting widely-used learning techniques to partial label data. For maximum likelihood techniques, the likelihood of observing each PL training example is defined over its candidate label set instead of the unknown ground-truth label (Jin and Ghahramani 2003; Liu and Dietterich 2012). K-nearest neighbor techniques determine class label of unseen instance via voting among the candidate labels of its neighboring examples (Hüllermeier and Beringer 2006; Zhang and Yu 2015). For maximum margin techniques, the classification margins over the PL training examples are defined by discriminating modeling outputs from candidate labels and non-candidate labels (Nguyen and Caruana 2008; Yu and Zhang 2017). For boosting techniques, the weight over each PL training example and the confidence over the candidate labels are updated in each boosting round (Tang and Zhang 2017).

Other than the above-mentioned works, there are a few works which work by fitting PL data to existing learning techniques. The CLPL approach (Cour, Sapp, and Taskar 2011) maps a d-dimensional instance in $\mathcal{X}$ into a $d \times q$-dimensional feature vector for each class label in $\mathcal{Y}$. For each PL training example $(\boldsymbol{x}_i, S_i)$, one positive example is generated by averaging mapped feature vectors w.r.t. candidate labels in S_i and $q - |S_i|$ negative examples are generated by taking the mapped feature vector w.r.t. each non-candidate label in $\mathcal{Y} \setminus S_i$. The PL-ECOC approach (Zhang, Yu, and Tang 2017) transforms each instance into a binary example via leveraging ECOC coding matrix (Dietterich and Bakiri 1995; Zhou 2012). For each PL training example $(\boldsymbol{x}_i, S_i)$, it is regarded as a positive or negative example if its candidate label set S_i entirely falls into the column dichotomy of the coding matrix.

In the next section, a novel partial label learning approach will be introduced. Different from existing partial label learning approaches, the generalized label distributions are recovered and utilized to facilitate the learning procedure. To our best knowledge, it is the first attempt to propose GLD to solve PL problem via label enhancement.

The Proposed Approach

As shown in Section 1, the task of partial label learning is to induce a multi-class classifier $f : \mathcal{X} \mapsto \mathcal{Y}$ from the partial label training set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i) | 1 \leq i \leq n\}$. Specifically, for each PL training example $(\boldsymbol{x}_i, S_i)$, the logical label vector $\boldsymbol{l}_i = (l_{\boldsymbol{x}_i}^{y_1}, l_{\boldsymbol{x}_i}^{y_2}, ..., l_{\boldsymbol{x}_i}^{y_c})^\top \in \{-1, 1\}^c$ is used to represent whether each label y_j is among the candidate label set. In the proposed approach, GLD is denoted by the vector $\boldsymbol{d}_i = (d_{\boldsymbol{x}_i}^{y_1}, d_{\boldsymbol{x}_i}^{y_2}, ..., d_{\boldsymbol{x}_i}^{y_c})^\top$.

In the next subsections, the two stages of PL-LE, i.e., generalized label distribution recovery and predictive model induction, will be scrutinized respectively.

Generalized Label Distribution Recovery

Given a PL training set $\mathcal{D}$, we construct the feature matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ and the logical label matrix $\boldsymbol{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_n]$. To recover the reasonable GLD matrix $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_n]$, we consider the model

$$\boldsymbol{d}_i = \boldsymbol{W}^\top \varphi(\boldsymbol{x}_i) + \boldsymbol{s} = \hat{\boldsymbol{W}} \boldsymbol{\phi}_i, \tag{1}$$

where $\boldsymbol{W} = [\boldsymbol{w}^1, ..., \boldsymbol{w}^c]$ is a weight matrix and $\boldsymbol{s} = (s^1, ..., s^c)^\top$ is a bias vector. $\varphi(\boldsymbol{x})$ is a nonlinear transforma-

tion of $\boldsymbol{x}$ to a higher dimensional feature space. For convenient describing, we set $\hat{\boldsymbol{W}} = [\boldsymbol{W}^\top, \boldsymbol{s}]$ and $\boldsymbol{\phi}_i = [\varphi(\boldsymbol{x}_i); 1]$. Accordingly, the goal of our method is to determine the best parameter $\hat{\boldsymbol{W}}^*$ that can generate a reasonable GLD $\boldsymbol{d}_i$ given the instance $\boldsymbol{x}_i$. Then, the optimization problem becomes

$$\min_{\hat{\boldsymbol{W}}} L(\hat{\boldsymbol{W}}) + \lambda R(\hat{\boldsymbol{W}}), \tag{2}$$

where L is a loss function, R is the function to mine hidden GDDs, and λ is the parameter trading off the two terms. Note that GLD recovery is essentially a pre-processing applied to the training set, which is different from standard supervised learning. Therefore, we does not need to consider the overfitting problem. Since the labeling information in GLD is inherited from the initial logical labels, we choose the least squares (LS) loss function as

$$\begin{aligned} L(\hat{\boldsymbol{W}}) &= \sum_{i=1}^{n} \|\hat{\boldsymbol{W}}\boldsymbol{\phi}_i - \boldsymbol{l}_i\|^2 \\ &= \mathrm{tr}[(\hat{\boldsymbol{W}}\boldsymbol{\Phi} - \boldsymbol{L})^\top(\hat{\boldsymbol{W}}\boldsymbol{\Phi} - \boldsymbol{L})], \end{aligned} \tag{3}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_n]$.

By leveraging the topological information of the feature space, hidden GDDs can be mined from the training examples. Therefore, we specify the $n \times n$ local similarity matrix $\boldsymbol{A}$ whose elements are calculated as follows.

- Step 1. We put an edge between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ if $\boldsymbol{x}_i$ is among K-nearest neighbors of $\boldsymbol{x}_j$ or $\boldsymbol{x}_j$ is among K-nearest neighbors of $\boldsymbol{x}_i$.
- Step 2. If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are connected, a_{ij} is specified as

$$a_{ij} = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2}\right). \tag{4}$$

Otherwise, a_{ij} is set to 0.

According to the smoothness assumption (Zhu, Lafferty, and Rosenfeld 2005), if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have a high degree of similarity, as measured by a_{ij}, then $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$ should be near to one another. This intuition leads to the following function which we wish to minimize:

$$\begin{aligned} R(\hat{\boldsymbol{W}}) &= \sum_{i,j} a_{ij}\|\boldsymbol{d}_i - \boldsymbol{d}_j\|^2 \\ &= \mathrm{tr}(\boldsymbol{D}\boldsymbol{G}\boldsymbol{D}^\top) \\ &= \mathrm{tr}(\hat{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{G}\boldsymbol{\Phi}^\top\hat{\boldsymbol{W}}^\top), \end{aligned} \tag{5}$$

where $\boldsymbol{G} = \hat{\boldsymbol{A}} - \boldsymbol{A}$ is the graph Laplacian and $\hat{\boldsymbol{A}}$ is the diagonal matrix whose elements are $\hat{a}_{ii} = \sum_{j=1}^{n} a_{ij}$.

Formulating the GLD recovery problem into an optimization framework over Eq. (3) and Eq. (5) yields the target function of $\hat{\boldsymbol{W}}$

$$\begin{aligned} T(\hat{\boldsymbol{W}}) = \mathrm{tr}[(\hat{\boldsymbol{W}}\boldsymbol{\Phi} - \boldsymbol{L})^\top(\hat{\boldsymbol{W}}\boldsymbol{\Phi} - \boldsymbol{L})] \\ +\lambda\mathrm{tr}(\hat{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{G}\boldsymbol{\Phi}^\top\hat{\boldsymbol{W}}^\top). \end{aligned} \tag{6}$$

Besides, we add a constraint to ensure that the GDD in recovered GLD possesses the same sign with the logical label and takes value with reasonable magnitude:

$$\forall 1 \le i \le n, 1 \le j \le c, 0 < d_{\boldsymbol{x}_i}^{y_j} l_{\boldsymbol{x}_i}^{y_j} < 1. \tag{7}$$

Note that Eq. (6) can be rewritten as:

$$\begin{aligned} T(\hat{\boldsymbol{w}}) = \sum_{j=1}^{c} \hat{\boldsymbol{w}}^j(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + 2\lambda\boldsymbol{\Phi}\boldsymbol{G}\boldsymbol{\Phi}^\top)\hat{\boldsymbol{w}}^{j\top} \\ - 2\hat{\boldsymbol{w}}^j\boldsymbol{\Phi}\boldsymbol{l}^j + \boldsymbol{l}^j\boldsymbol{l}^{j\top}, \end{aligned} \tag{8}$$

where $\hat{\boldsymbol{w}}^j$ is the j-th row of the parameter matrix $\hat{\boldsymbol{W}}$ and $\boldsymbol{l}^j$ is the j-th row of the logical label matrix $\boldsymbol{L}$. Accordingly, for the parameter matrix $\hat{\boldsymbol{W}}$, its j-th row $\hat{\boldsymbol{w}}^j$ can be determined by solving the following constrained quadratic programming process:

$$\begin{aligned} &\min_{\hat{\boldsymbol{w}}^j} \hat{\boldsymbol{w}}^j(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + 2\lambda\boldsymbol{\Phi}\boldsymbol{G}\boldsymbol{\Phi}^\top)\hat{\boldsymbol{w}}^{j\top} - 2\hat{\boldsymbol{w}}^j\boldsymbol{\Phi}\boldsymbol{l}^j \\ &\text{s.t.} \quad 0 < d_{\boldsymbol{x}_i}^{y_j} l_{\boldsymbol{x}_i}^{y_j} < 1, \forall\, 1 \le i \le n. \end{aligned} \tag{9}$$

When the best parameter $\hat{\boldsymbol{W}}^*$ is determined, the generalized label distribution $\boldsymbol{d}_i$ can be generated through Eq. (1).

Predictive Model Induction

Given the recovered $\boldsymbol{d}_i$ of $\boldsymbol{x}_i$, the original PL training set can be transformed into $\mathcal{E} = \{(\boldsymbol{x}_i, \boldsymbol{d}_i) | 1 \le i \le n\}$. As $\boldsymbol{d}_i$ for each training example $(\boldsymbol{x}_i, \boldsymbol{d}_i)$ is numerical, it is natural to induce the predictive model by employing multi-output regression techniques. Similar to the MSVR, we generalize a regressor to solve the multi-dimensional case. In addition, our regressor not only concerns the distance between the predicted and the real values, but also the sign consistency of them. It leads to the minimization of

$$\Omega(\boldsymbol{\Theta}, \boldsymbol{b}) = \frac{1}{2}\sum_{j=1}^{c}\|\boldsymbol{\theta}^j\|^2 + C_1\sum_{i=1}^{n}\Omega_{1i} + C_2\sum_{i=1}^{n}\Omega_{2i}, \tag{10}$$

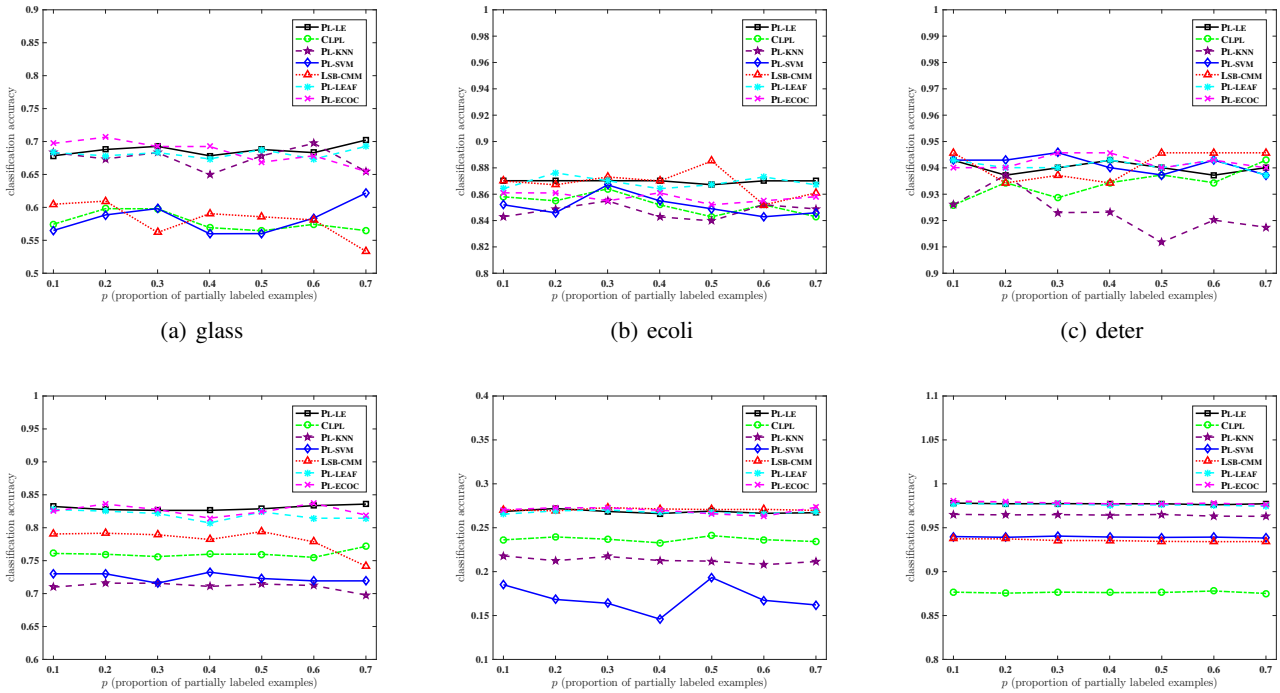
where $\boldsymbol{\Theta} = [\boldsymbol{\theta}^1, ..., \boldsymbol{\theta}^c]$, $\boldsymbol{b} = [b^1, ..., b^c]$, Ω_1 and Ω_2 are the regression loss and the sign loss, respectively.

As shown in Eq. 10, the first term of $\Omega(\boldsymbol{\Theta}, \boldsymbol{b})$ controls the complexity of the induced model. In addition, the second term of $\Omega(\boldsymbol{\Theta}, \boldsymbol{b})$ is defined to consider all dimensions into a unique restriction and yield a single support vector for all dimensions:

$$\Omega_{1i} = \begin{cases} 0 & r_i < \varepsilon \\ r_i^2 - 2r_i\varepsilon + \varepsilon^2 & r_i \ge \varepsilon, \end{cases} \tag{11}$$

where $r_i = \|\boldsymbol{e}_i\| = \sqrt{\boldsymbol{e}_i^\top \boldsymbol{e}_i}$, $\boldsymbol{e}_i = \boldsymbol{d}_i - \varphi(\boldsymbol{x}_i)^\top\boldsymbol{\Theta} - \boldsymbol{b}$. This will create an insensitive zone determined by ε around the estimate, i.e., the loss of r less than ε will be ignored. The third term is used to make the signs of the predictive output and the logical label same as much as possible:

$$\Omega_{2i} = -\sum_{j=1}^{c} l_i^j(\varphi(\boldsymbol{x}_i)^\top\boldsymbol{\theta}^j + b^j). \tag{12}$$



(a) glass (b) ecoli (c) deter

(d) vehicle (e) abalone (f) usps

Figure 2: Classification accuracy of each comparing algorithm changes as p (proportion of partially labeled examples) increases from 0.1 to 0.7 (with one false positive candidate label $[r = 1]$).

The meaning of Eq. (12) is that if the signs of the predictive output and the logical label are different, there will be some positive loss, otherwise the loss will be negative.

To minimize $\Omega(\boldsymbol{\Theta}, \boldsymbol{b})$, we use an iterative quasi-Newton method called Iterative Re-Weighted Least Square (IRWLS) (Pérez-Cruz et al. 2000). Firstly, $\Omega_1(\boldsymbol{\Theta}, \boldsymbol{b})$ is approximated by its first order Taylor expansion at the solution of the current k-th iteration, denoted by $\boldsymbol{\Theta}^{(k)}$ and $\boldsymbol{b}^{(k)}$:

$$\Omega'_{1i} = \Omega_{1i}^{(k)} + \left.\frac{d\Omega_1}{dr}\right|_{r_i^{(k)}} \frac{(\boldsymbol{e}_i^{(k)})^\top}{r_i^{(k)}} \left(\boldsymbol{e}_i - \boldsymbol{e}_i^{(k)}\right), \tag{13}$$

where $\boldsymbol{e}_i^{(k)}$ and $r_i^{(k)}$ are calculated from $\boldsymbol{\Theta}^{(k)}$ and $\boldsymbol{b}^{(k)}$. Then a quadratic approximation is further constructed as

$$\begin{aligned}\Omega''_{1i} =& \Omega_{1i}^{(k)} + \left.\frac{d\Omega_1}{dr}\right|_{r_i^{(k)}} \frac{r_i^2 - (r_i^{(k)})^2}{2r_i^{(k)}} \\ =& \frac{1}{2} a_i r_i^2 + \tau,\end{aligned} \tag{14}$$

where

$$a_i = \frac{1}{r_i^{(k)}} \left.\frac{d\Omega_1}{dr}\right|_{r_i^{(k)}} = \begin{cases} 0 & r_i^{(k)} < \varepsilon \\ \frac{2\left(r_i^{(k)} - \varepsilon\right)}{r_i^{(k)}} & r_i^{(k)} \geq \varepsilon, \end{cases} \tag{15}$$

and τ is a constant term that does not depend on either $\boldsymbol{\Theta}^{(k)}$ or $\boldsymbol{b}^{(k)}$. Combining Eq. (10), (12) and (14), we can get

$$\begin{aligned}\Omega''(\boldsymbol{\Theta}, \boldsymbol{b}) =& \frac{1}{2} \sum_{j=1}^{c} \|\boldsymbol{\theta}^j\|^2 + \frac{1}{2} C_1 \sum_{i=1}^{n} a_i r_i^2 \\ & - C_2 \sum_{i=1}^{n} \sum_{j=1}^{c} l_i^j (\varphi(\boldsymbol{x}_i)^\top \boldsymbol{\theta}^j + b^j) + \tau.\end{aligned} \tag{16}$$

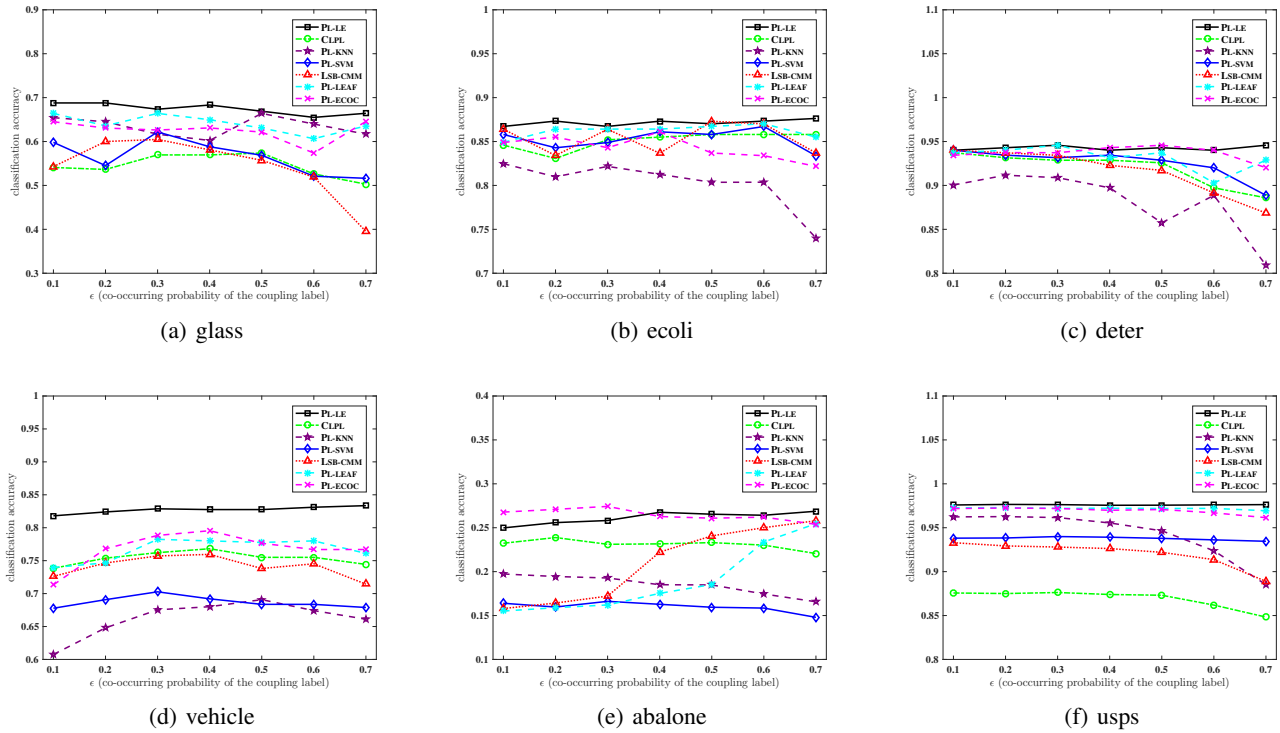
It is a piecewise quadratic problem whose optimum can be integrated as solving a system of linear equations for $j = 1, \ldots, c$:

$$\begin{bmatrix} C_1 \boldsymbol{\Phi}^\top \boldsymbol{F} \boldsymbol{\Phi} + \boldsymbol{I} & C_1 \boldsymbol{\Phi}^\top \boldsymbol{a} \\ C_1 \boldsymbol{a}^\top \boldsymbol{\Phi} & C_1 \mathbf{1}^\top \boldsymbol{a} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}^j \\ b^j \end{bmatrix} = \begin{bmatrix} C_1 \boldsymbol{\Phi}^\top \boldsymbol{F} \boldsymbol{d}^j + C_2 \boldsymbol{\Phi}^\top \boldsymbol{l}^j \\ C_1 \boldsymbol{a}^\top \boldsymbol{d}^j + C_2 \mathbf{1}^\top \boldsymbol{l}^j \end{bmatrix}, \tag{17}$$

Table 1: Characteristics of the controlled UCI data sets.

Data Set	#Examples	#Features	# Labels
glass	214	9	6
ecoli	336	7	8
deter	358	23	6
vehicle	846	18	4
abalone	4,177	7	29
usps	9,298	256	10

Configurations
(I) $r = 1, p \in \{0.1, 0.2, \ldots, 0.7\}$
(II) $r = 2, p \in \{0.1, 0.2, \ldots, 0.7\}$
(III) $r = 3, p \in \{0.1, 0.2, \ldots, 0.7\}$
(IV) $p = 1, r = 1, \epsilon \in \{0.1, 0.2, \ldots, 0.7\}$



(a) glass (b) ecoli (c) deter

(d) vehicle (e) abalone (f) usps

Figure 3: Classification accuracy of each comparing algorithm changes as ϵ (co-occurring probability of the coupling label) increases from 0.1 to 0.7 (with 100% partially labeled examples [$p = 1$] and one false positive candidate label [$r = 1$]).

where $\boldsymbol{\Phi} = [\varphi(\boldsymbol{x}_1), ..., \varphi(\boldsymbol{x}_n)]^\top$, $\boldsymbol{a} = [a_1, ..., a_n]^\top$, $\boldsymbol{F}_i^k = a_i\delta_i^k$ (δ_i^k is the Kronecker's delta function), and $\boldsymbol{l}^j = [l_1^j, \ldots, l_n^j]^\top$. Then, the direction of the optimal solution of Eq. (17) is used as the descending direction for the optimization of $\Omega(\boldsymbol{\Theta}, \boldsymbol{b})$, and the solution for the next iteration ($\boldsymbol{\Theta}^{(k+1)}$ and $\boldsymbol{b}^{(k+1)}$) is obtained via a line search algorithm along this direction.

According to the representor's theorem (Smola 1999), under fairly general conditions, a learning problem can be expressed as a linear combination of the training examples in the feature space, i.e. $\boldsymbol{\theta}^j = \sum_i \boldsymbol{\eta}^j \varphi(\boldsymbol{x}_i)$. If we replace this expression into Eq. (9) and Eq. (17), it will generate the inner product$< \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) >$, and then the kernel trick can be applied.

Let $\boldsymbol{\Theta}^*$ and $\boldsymbol{b}^*$ be the resulting model after the whole iterative optimization process, PL-LE makes prediction on the class label of unseen instance $\boldsymbol{x}$ as follows:

$$f(\boldsymbol{x}) = \arg\max_{y_j \in \mathcal{Y}} \varphi(\boldsymbol{x})^\top \boldsymbol{\theta}^{*j} + b^{*j} \tag{18}$$

Experiments

Methodology

The performance of PL-LE is compared against six state-of-the-art partial label learning approaches, each configured with parameters suggested in respective literature:

- CLPL (Cour, Sapp, and Taskar 2011) which transforms partial label learning problem into binary learning problem via feature mapping with convex loss optimization [suggested configuration: SVM with squared hinge loss].
- PL-KNN (Hüllermeier and Beringer 2006) which adopts K-nearest neighbor technique to learn from PL data via weighted voting [suggested configuration: $k = 10$].
- PL-SVM (Nguyen and Caruana 2008) which adopts maximum margin technique to learn from PL data via l_2 regularization [suggested configuration: regularization parameter pool with $\{10^{-3}, \ldots, 10^3\}$].
- LSB-CMM (Liu and Dietterich 2012) which adopts maximum likelihood to learn from PL data via mixture models [suggested configuration: $5q$ mixture components].
- PL-LEAF (Zhang, Zhou, and Liu 2016) which adopts a two-stage approach to learn from partial label examples based on feature-aware disambiguation. [suggested configuration: $K = 10$, $C_1 = 10$, $C_2 = 1$].
- PL-ECOC (Zhang, Yu, and Tang 2017) which transforms partial label learning problem into binary learning problem via ECOC coding matrix [suggested configuration: codeword length $L = \lceil 10 \log_2(q) \rceil$].

For PL-LE, the parameter λ is set to 0.01 and the number of neighbors K is set to 20. The parameters C_1 and C_2 are set to 1 and 1, respectively. The kernel function in PL-LE is Gaussian kernel.

Table 2: Win/tie/loss counts (pairwise t-test at 0.05 significance level) on the classification performance of PL-LE against each comparing approach.

	PL-LE **against**					
	CLPL	PL-KNN	PL-SVM	LSB-CMM	PL-LEAF	PL-ECOC
varying p [r=1]	26/16/0	20/20/0	23/19/0	12/30/0	0/42/0	0/42/0
varying p [r=2]	27/15/0	26/16/0	24/18/0	12/30/0	0/42/0	1/41/0
varying p [r=3]	27/15/0	25/17/0	27/15/0	15/27/0	2/40/0	1/41/0
varying ϵ [p, r=1]	25/17/0	28/14/0	28/14/0	24/18/0	8/34/0	6/36/0
In Total	**105/63/0**	**101/67/0**	**102/66/0**	**63/105/0**	**10/158/0**	**8/160/0**

Table 3: Characteristic of the real-world partial label data sets.

Data Set	#Examples	#Features	#Class Labels	avg. #CLs	Task Domain
FG-NET	1,002	262	78	7.48	*facial age estimation* (Panis and Lanitis 2015)
Lost	1,122	108	16	2.23	*automatic face naming* (Cour, Sapp, and Taskar 2011)
MSRCv2	1,758	48	23	3.16	*object classification* (Liu and Dietterich 2012)
BirdSong	4,998	38	13	2.18	*bird song classification* (Briggs, Fern, and Raich 2012)
Soccer Player	17,472	279	171	2.09	*automatic face naming* (Zeng et al. 2013)
Yahoo! News	22,991	163	219	1.91	*automatic face naming* (Guillaumin, Verbeek, and Schmid 2010)

Controlled UCI Data Sets

Table 1 summarizes the characteristics of six controlled UCI data sets (Bache and Lichman 2013). Concretely, following the widely-used controlling protocol, an artificial partial label data set is derived from one multi-class UCI data set by configuring three controlling parameters p, r and ϵ (Cour, Sapp, and Taskar 2011; Liu and Dietterich 2012; Chen et al. 2014; Zhang, Yu, and Tang 2017). Here, p controls the proportion of examples which are partially labeled (i.e. $|S_i| > 1$), r controls the number of false positive labels in the candidate label set (i.e. $|S_i| = r+1$), and ϵ controls the co-occurring probability between one extra candidate label and the ground-truth label. As shown in Table 1, a total of 28 (4×7) parameter configurations are considered for each controlled UCI data set.

Figure 2 illustrates the classification accuracy of each comparing algorithm as p increases from 0.1 to 0.7 with step-size 0.1 ($r = 1$). Along with the ground-truth label, one class label in $\mathcal{Y}$ will be randomly picked up to constitute the candidate label set. Due to page limit, figures for the cases of $r = 2$ and $r = 3$ are not illustrated here while similar results to Figure 2 can be observed as well. Figure 3 illustrates the classification accuracy of each comparing algorithm as ϵ increases from 0.1 to 0.7 with step-size 0.1 ($p = 1, r = 1$). Given any label $y \in \mathcal{Y}$, one extra label $y' \in \mathcal{Y}$ is designated as the coupling label which co-occurs with y in the candidate label set with probability ϵ. Otherwise, any other class label would be randomly chosen to co-occur with y.

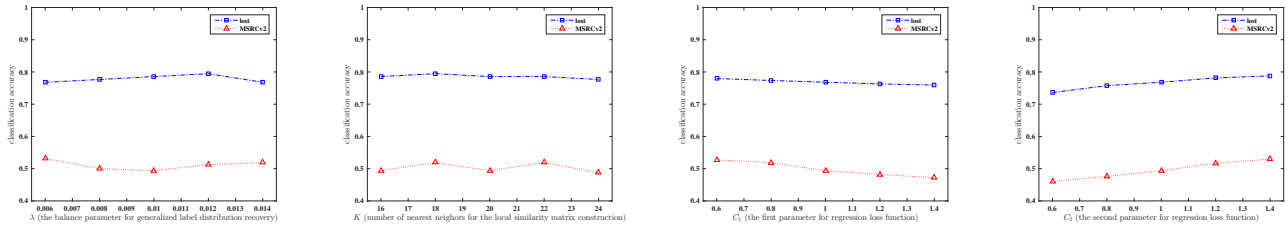
As illustrated in Figures 2 and 3, the performance of PL-LE is highly competitive to other comparing algorithms in most cases. Furthermore, pairwise t-test at 0.05 significance level is conducted based on the results of ten-fold cross-validation. Table 2 reports the win/tie/loss counts between PL-LE and each comparing approach. Specifically, out of the 168 statistical tests (28 configurations x 6 UCI data sets), it is shown that:

- Across all the controlling parameter configurations and controlled UCI data sets, none of the comparing algorithms have outperformed PL-LE significantly.
- Comparing to averaging-based disambiguation approaches (in total), PL-LE achieves superior performance against CLPL and PL-KNN in 62.5% cases (105 out of 168) and 60.1% cases (101 out of 168) respectively.
- Comparing to identification-based disambiguation approaches (in total), PL-LE achieves superior performance against PL-SVM and LSB-CMM in 60.7% cases (102 out of 168) and 37.5% cases (63 out of 168) respectively.
- PL-LE achieves comparable performance against PL-LEAF and PL-ECOC in 94.0% cases (158 out of 168) and 95.2% cases (160 out of 168) respectively. In addition, PL-LE achieves superior performance against PL-LEAF and PL-ECOC in 6.0% cases (10 out of 168) and 4.8% cases (9 out of 168) respectively.

Real-World Data Sets

Table 3 summarizes the characteristics of real-world partial label data sets, which are collected from several application domains including `FG-NET` (Panis and Lanitis 2015) for facial age estimation, `Lost` (Cour, Sapp, and Taskar 2011), `Soccer Player` (Zeng et al. 2013) and `Yahoo!News` (Guillaumin, Verbeek, and Schmid 2010) for automatic face naming from images or videos, `MSRCv2` (Liu and Dietterich 2012) for object classification, and `BirdSong` (Briggs, Fern, and Raich 2012) for bird song classification. The average number of candidate labels (avg. #CLs) for each real-world partial label data set is also recorded in Table 3.

Table 4 reports the mean classification accuracy as well as standard deviation of each comparing algorithm. Pairwise t-test at 0.05 significance level is conducted based on the



(a) Varying λ (b) Varying K (c) Varying C_1 (d) Varying C_2

Figure 4: Parameter sensitivity analysis for PL-LE on the Lost and MSRCv2 data sets. (a) Classification accuracy of PL-LE changes as λ increases from 0.006 to 0.014 with step-size 0.002 ($K = 20, C_1 = 1, C_2 = 1$); (b) Classification accuracy of PL-LE changes as K increases from 16 to 24 with step-size 2 ($\lambda = 0.01, C_1 = 1, C_2 = 1$); (c) Classification accuracy of PL-LE changes as C_1 increases from 0.6 to 1.4 with step-size 0.2 ($\lambda = 0.01, K = 20, C_2 = 1$); (d) Classification accuracy of PL-LE changes as C_2 increases from 0.6 to 1.4 with step-size 0.2 ($\lambda = 0.01, K = 20, C_1 = 1$).

Table 4: Classification accuracy (mean±std) of each comparing algorithm on the real-world partial label data sets. In addition, •/◦ indicates whether is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 0.05 significance level).

	FG-NET	Lost	MSRCv2	BirdSong	Soccer Player	Yahoo! News
PL-LE	0.082±0.023	0.773±0.043	0.499±0.037	0.730±0.013	0.536±0.020	0.653±0.006
CLPL	0.063±0.027	0.742±0.038	0.413±0.041•	0.632±0.019•	0.368±0.010•	0.462±0.009•
PL-KNN	0.038±0.025•	0.424±0.036•	0.448±0.037•	0.614±0.021•	0.497±0.015•	0.457±0.004•
PL-SVM	0.063±0.029	0.729±0.042•	0.461±0.046	0.660±0.037•	0.464±0.011•	0.629±0.010•
LSB-CMM	0.059±0.025	0.693±0.035•	0.473±0.037	0.672±0.056•	0.498±0.017•	0.645±0.005•
PL-LEAF	0.076±0.037	0.717±0.059•	0.498±0.035	0.723±0.013	0.532±0.017	0.641±0.006•
PL-ECOC	0.040±0.018•	0.653±0.053•	0.440±0.039•	0.731±0.013	0.494±0.015•	0.610±0.009•

ten-fold cross-validation, where the test outcomes between PL-LE and the comparing approaches are also recorded.

As shown in Table 4, it is impressive to observe that:

- On all data sets, PL-LE achieves superior or at least comparable performance against all the comparing approaches.
- On all data sets, PL-LE significantly outperforms PL-KNN.
- PL-LE significantly outperforms PL-ECOC on FG-NET, Lost, MSRCv2, Soccer Player and Yahoo!News.
- PL-LE achieves superior performance against all the comparing approaches except PL-LEAF on the two large-scale data sets (Soccer Player and Yahoo!News) .

Note that the proposed method performs better on larger datasets. It is because that label enhancement can better recover the hidden GLD in lager datasets.

Sensitivity Analysis

In this subsection, performance sensitivity of the proposed PL-LE approach w.r.t. its parameters λ, K, C_1 and C_2 will be further analyzed.

Figure 4 illustrates how PL-LE performs under different parameter configurations. For clarity of illustration, two data sets (MSRCv2 and Lost) are chosen here for sensitivity analysis while similar observations also hold on other data sets.

As shown in Figure 4, it is obvious that the performance of PL-LE is stable across a broad range of each parameter. This property is quite desirable as one can make use of PL-LE to achieve robust classification performance without the need of parameter fine-tuning. Therefore, the parameter configuration for PL-LE in Subsection 4.1 naturally follows from these observations.

Conclusion

In this paper, the problem of partial label learning is studied where a novel approach PL-LE is proposed. Different from existing strategies, PL-LE considers the generalized label distribution in the training data sets. Since generalized label distribution is not explicitly available in the training sets, PL-LE recovers the generalized label distribution via leveraging the topological information of the feature space, and then induces the predictive model based on multi-output regression analysis. Effectiveness of the proposed approach is validated via comprehensive experiments on both controlled UCI data sets and real-world PL data sets.

It is interesting to investigate effective ways to make full use of the generalized label distribution in partial label learning. Furthermore, label enhancement need to be investigated when the partial label sets of PL training examples exhibit certain structures. In the future, it is also important to explore other techniques to recover the generalized label distribution for partial label learning.

Acknowledgments

This research was supported by the National Key Research & Development Plan of China (No. 2017YFB1002801), the National Science Foundation of China (61622203), the Jiangsu Natural Science Funds for Distinguished Young Scholar (BK20140022), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Collaborative Innovation Center of Wireless Communications Technology.

References

Amores, J. 2013. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence* 201:81–105.

Bache, K., and Lichman, M. 2013. UCI machine learning repository. School of Information and Computer Sciences, University of California, Irvine.

Briggs, F.; Fern, X. Z.; and Raich, R. 2012. Rank-loss support instance machines for MIML instance annotation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 534–542.

Chen, Y.-C.; Patel, V. M.; Chellappa, R.; and Phillips, P. J. 2014. Ambiguously labeled learning using dictionaries. *IEEE Transactios on Information Forensics and Security* 9(12):2076–2088.

Chen, C.-H.; Patel, V. M.; and Chellappa, R. in press. Learning from ambiguously labeled face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Cour, T.; Sapp, B.; and Taskar, B. 2011. Learning from partial labels. *Journal of Machine Learning Research* 12(May):1501–1536.

Dietterich, T. G., and Bakiri, G. 1995. Solving multiclass learning problem via error-correcting output codes. *Journal of Artificial Intelligence Research* 2(1):263–286.

Geng, X. 2016. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering* 28(7):1734–1748.

Guillaumin, M.; Verbeek, J.; and Schmid, C. 2010. Multiple instance metric learning from automatically labeled bags of faces. In *Lecture Notes in Computer Science 6311*. Berlin: Springer. 634–647.

Hou, P.; Geng, X.; and Zhang, M.-L. 2016. Multi-label manifold learning. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 1680–1686.

Hüllermeier, E., and Beringer, J. 2006. Learning from ambiguously labeled examples. *Intelligent Data Analysis* 10(5):419–439.

Jie, L., and Orabona, F. 2010. Learning from candidate labeling sets. In *Advances in Neural Information Processing Systems 23*. 1504–1512.

Jin, R., and Ghahramani, Z. 2003. Learning with multiple labels. In *Advances in Neural Information Processing Systems 15*, 897–904.

Liu, L., and Dietterich, T. 2012. A conditional multinomial mixture model for superset label learning. In *Advances in Neural Information Processing Systems 25*, 557–565.

Nguyen, N., and Caruana, R. 2008. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 381–389.

Panis, G., and Lanitis, A. 2015. An overview of research activities in facial age estimation using the fg-net aging database. In *Lecture Notes in Computer Science 8926*. Berlin: Springer. 737–750.

Pérez-Cruz, F.; Navia-Vázquez, A.; Alarcón-Diana, P. L.; and Artes-Rodriguez, A. 2000. An irwls procedure for svr. In *Signal Processing Conference, European*, 1–4. IEEE.

Smola, A. J. 1999. *Learning with kernels*. Ph.D. Thesis, GMD, Birlinghoven, German.

Tang, C.-Z., and Zhang, M.-L. 2017. Confidence-rated discriminative partial label learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2611–2617.

Xu, N.; Tao, A.; and Geng, X. 2018. Label enhancement for label distribution learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2926–2932.

Yu, F., and Zhang, M.-L. 2017. Maximum margin partial label learning. *Machine Learning* 106(4):573–593.

Zeng, Z.; Xiao, S.; Jia, K.; Chan, T.-H.; Gao, S.; Xu, D.; and Ma, Y. 2013. Learning by associating ambiguously labeled images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 708–715.

Zhang, M.-L., and Yu, F. 2015. Solving the partial label learning problem: An instance-based approach. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 4048–4054.

Zhang, M.-L., and Zhou, Z.-H. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26(8):1819–1837.

Zhang, M.-L.; Yu, F.; and Tang, C.-Z. 2017. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering* 29(10):2155–2167.

Zhang, M.-L.; Zhou, B.-B.; and Liu, X.-Y. 2016. Partial label learning via feature-aware disambiguation. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1335–1344.

Zhou, Z.-H. 2012. *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL: Chapman & Hall/CRC.

Zhu, X., and Goldberg, A. B. 2009. Introduction to semi-supervised learning. In *Synthesis Lectures to Artificial Intelligence and Machine Learning*. San Francisco, CA: Morgan & Claypool Publishers. 1–130.

Zhu, X.; Lafferty, J.; and Rosenfeld, R. 2005. *Semi-supervised learning with graphs*. Carnegie Mellon University, language technologies institute, school of computer science.