Disambiguation-Free Partial Label Learning

Min-Ling Zhang*

Abstract

Partial label learning deals with the problem where each training example is associated with a set of *candidate* labels, among which only one is correct. The common strategy is to try to disambiguate their candidate labels, such as by identifying the ground-truth label iteratively or by treating each candidate label equally. Nevertheless, the above disambiguation strategy is prone to be misled by the false positive label(s) within candidate label set. In this paper, a new disambiguation-free approach to partial label learning is proposed by employing the well-known error-correcting output codes (ECOC) techniques. Specifically, to build the binary classifier with respect to each column coding, any partially labeled example will be regarded as a positive or negative training example only if its candidate label set *entirely* falls into the coding dichotomy. Experiments on controlled and real-world data sets clearly validate the effectiveness of the proposed approach.

Keywords: Classification, weakly-supervised learning, partially labeled data, error-correcting output codes

1 Introduction

Partial label (PL) learning refers to the problem where the training example is associated with a set of candidate labels, among which only one label corresponds to the ground-truth [10, 16]. The need to learn from data with partial labels arises in many real-world applications such as automatic face naming in videos [9] and webpages [15], image classification [23], bird song classification [17], etc.

Formally speaking, let $\mathcal{X} = \mathbb{R}^d$ be the input space and $\mathcal{Y} = \{y_1, y_2, \ldots, y_q\}$ be the output space with q possible class labels, the task of partial label learning is to induce a *multi-class* classifier $f : \mathcal{X} \to \mathcal{Y}$ from the partial label training set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i) \mid 1 \leq i \leq m\}$. For each PL training example $(\boldsymbol{x}_i, S_i)$, $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector $(x_{i1}, x_{i2}, \ldots, x_{id})^\top$ while $S_i \subseteq \mathcal{Y}$ is the set of candidate labels associated with $\boldsymbol{x}_i$. In partial label learning, the (unknown) ground-truth label y_i for $\boldsymbol{x}_i$ is assumed to appear in the candidate label set, i.e. $y_i \in S_i$.[1]

Apparently, the major difficulty for partial label learning lies in that the ground-truth label of the training example is concealed in its candidate label set and thus is not directly accessible by the learning algorithm. Therefore, the common strategy to learn from partially labeled examples is to try to *disambiguate* the set of candidate labels. One way is to regard the ground-truth label as latent variable which is identified via iterative refining procedure such as EM [13, 16, 17, 18, 21]. Another way is to treat all the candidate labels equally where the final prediction is made by averaging their modeling outputs [10, 14].

Though disambiguation presents as an intuitive and reasonable strategy to partial label learning, its effectiveness is largely affected by the *false positive* label(s) within candidate label set (i.e. $S_i \setminus \{y_i\}$). For disambiguation by ground-truth label identification, the identified label $\hat{y}_i$ refined in each iteration might turn out to be the false positive label instead of the ground-truth one. Similarly, for disambiguation by candidate label averaging, the modeling outputs yielded by false positive labels might overwhelm the essential modeling output yielded by the ground-truth label. For either of the above disambiguation ways, the negative influence brought by false positive labels will be more pronounced as the size of candidate label set increases.

In this paper, we adopt another strategy to learn from partially labeled examples which does not rely on disambiguating candidate labels. Accordingly, a new approach named Pl-ecoc is proposed, which adapts one of the famous multi-class learning techniques, i.e. *error-correcting output codes* (ECOC) [11, 25], to help fulfill the partial label learning task. The key adaptation employed by Pl-ecoc lies in how the binary classifiers corresponding to the ECOC coding matrix are built. For each column of the binary coding matrix, one binary classifier is built based on binary training examples derived from the partial label training set. Specifically, any partially labeled example will be regarded as a positive or negative training example only if its candidate label set *entirely* falls into the positive or nega-

*School of Computer Science and Engineering, MOE Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing 210096, China. Email: zhangml@seu.edu.cn

[1] In some literatures, partial label learning is also termed as *ambiguous label learning* [14], *soft label learning* [7, 8] or *superset label learning* [17].

tive dichotomy specified by the column coding. In this way, the set of candidate labels is treated as an entirety without resorting to the any disambiguation procedure. Comparative studies on controlled UCI data sets as well as real-world partial label data sets show that Pl-ecoc achieves highly competitive performance against other well-established approaches.

The rest of this paper is organized as follows. Section 2 briefly reviews related works on partial label learning. Section 3 gives detailed descriptions on the proposed approach. Section 4 reports the results of experimental studies. Finally, Section 5 concludes.

2 Related Work

In view of the supervision spectrum, partial label learning can be regarded as a *weakly-supervised* learning scenario with implicit labeling information. It lies between the two ends of the supervision spectrum, i.e. traditional supervised learning with explicit supervision and unsupervised learning with blind supervision. Although it is related to other weakly-supervised learning scenarios such as *semi-supervised learning*, *multi-instance learning* and *multi-label learning*, the type of weak supervision information handled by partial label learning is different to all of them.

Semi-supervised learning [6, 26] deals with training examples which are either explicitly labeled ($|S_i| = 1$) or unlabeled ($|S_i| = q$), while in partial label learning most training examples are *partially* labeled ($1 < |S_i| < q$). Multi-instance learning [2, 12] deals with training examples whose labels are assigned at the level of multi-instance bags, while in partial label learning labels are assigned at the level of *individual* instances. Multi-label learning [20, 24] deals with training examples which are assigned with a set of valid labels, while in partial label learning the set of labels assigned to training examples are only *candidate* ones.

Existing approaches to partial label learning work by disambiguating candidate labels assigned to each training example. One way toward disambiguation is to assume some discriminative model $F(\boldsymbol{x}, y; \boldsymbol{\theta})$ with which the ground-truth label is identified as $\hat{y}_i = \arg\max_{y \in \mathcal{Y}} F(\boldsymbol{x}, y; \boldsymbol{\theta})$. Here, the model parameters $\boldsymbol{\theta}$ are *iteratively* refined by optimizing certain objectives over PL training examples, such as the maximum likelihood criterion: $\sum_{i=1}^{m} \log \left(\sum_{y \in S_i} F(\boldsymbol{x}_i, y; \boldsymbol{\theta}) \right)$ [13, 16, 17, 21], or the maximum margin criterion: $\sum_{i=1}^{m} \left(\max\limits_{y \in S_i} F(\boldsymbol{x}_i, y; \boldsymbol{\theta}) - \max\limits_{y \notin S_i} F(\boldsymbol{x}_i, y; \boldsymbol{\theta}) \right)$ [18].

Another way toward disambiguation is to assume equal contribution of each candidate label in the modeling process. For parametric models, the *averaged* output over all candidate labels $\frac{1}{|S_i|} \sum_{y \in S_i} F(\boldsymbol{x}_i, y; \boldsymbol{\theta})$ is distinguished from outputs over non-candidate labels $F(\boldsymbol{x}_i, y; \boldsymbol{\theta})$ ($y \notin S_i$) [10]. For non-parametric models, the predicted label for test example $\boldsymbol{x}^*$ is determined by voting among the candidate labels of its neighboring examples: $f(\boldsymbol{x}^*) = \arg\max_{y \in \mathcal{Y}} \sum_{j \in \mathcal{N}(\boldsymbol{x}^*)} \llbracket y \in S_j \rrbracket$[2][14].

The limited literatures reviewed in this section clearly reflect the difficulties in learning from partial label data, which is mainly due to the ambiguous labeling information conveyed by PL training examples. To circumvent potential issues encountered during disambiguation, a simple yet effective disambiguation-free partial label learning approach is proposed as follows.

3 The Pl-ecoc Approach

3.1 Binary Decomposition to Multi-class Classification

Recall that the ultimate goal of partial label learning is to induce a multi-class classifier $f : \mathcal{X} \to \mathcal{Y}$ which maps from the instance space to the label space. In traditional supervised learning, the arguably most popular mechanism is to transform the multi-class learning problem into a number of binary learning problems via *one-vs-rest* or *one-vs-one* decomposition.

The one-vs-rest decomposition induces a total of q binary classifiers, one for each class label y_j ($1 \leq j \leq q$). Each binary classifier is built by treating training examples from y_j as positive ones and the others as negative ones, and the final prediction is determined by the binary classifier with largest output on the test example. The one-vs-one decomposition induces a total of $\binom{q}{2}$ binary classifiers, one for each pair of class labels (y_j, y_k) ($j < k$). Each binary classifier is built by treating training examples from y_j as positive ones and those from y_k as negative ones, and the final prediction is determined by choosing the class label receiving maximal votes from those binary classifiers on the test example.

Unfortunately, neither the one-vs-rest nor the one-vs-one decomposition can be employed to induce the multi-class classifier under partial label learning scenario. As the ground-truth label of the PL training example is not available explicitly, the training examples required to build the corresponding binary classifiers can not be properly decided. In this paper, a new approach named Pl-ecoc is proposed by adapting the ECOC techniques, which is capable of learning from partially labeled examples while at the same time maintains the simplicity merit owned by binary decomposition mechanism.

[2] For any predicate π, $\llbracket \pi \rrbracket$ returns 1 if π holds and 0 otherwise. In addition, $\mathcal{N}(\boldsymbol{x}^*)$ stores the indexes of $\boldsymbol{x}^*$'s neighboring examples.

Table 1: Pseudo-code of Pl-ecoc.

```
$y$=Pl-ecoc($\mathcal{D}$, $L$, $\mathfrak{B}$, $thr$, $\boldsymbol{x}$)
Inputs:
$\mathcal{D}$ :   partial-label training set $\{(\boldsymbol{x}_i, S_i) \mid 1 \leq i \leq m\}$ $\big(\boldsymbol{x}_i \in \mathcal{X}, S_i \subseteq \mathcal{Y}, \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{y_1, y_2, \cdots, y_q\}\big)$
$L$ :   ECOC codeword length
$\mathfrak{B}$ :   binary learner for classifier building
$thr$ :   thresholding binary training set size w.r.t. the ECOC coding matrix
$\boldsymbol{x}^*$ :   test example ($\boldsymbol{x}^* \in \mathcal{X}$)
Outputs:
$y^*$ :   predicted class label for $\boldsymbol{x}^*$ ($y^* \in \mathcal{Y}$)
Process:
1.  $l = 0$;
2.  While $l \neq L$ do
3.    Randomly generate a $q$-bits column coding $\boldsymbol{v} = [v_1, v_2, \ldots, v_q]^\top \in \{-1, +1\}^q$;
4.    Dichotomize the label space according to $\boldsymbol{v}$: $\mathcal{Y}_{\boldsymbol{v}}^+ = \{y_j \mid \boldsymbol{v}_j = +1,\ 1 \leq j \leq q\}$, $\mathcal{Y}_{\boldsymbol{v}}^- = \mathcal{Y} \setminus \mathcal{Y}_{\boldsymbol{v}}^+$;
5.    Initialize a binary training set $\mathcal{D}_{\boldsymbol{v}} = \emptyset$;
6.    for $i = 1$ to $m$ do
7.      if ($S_i \subseteq \mathcal{Y}_{\boldsymbol{v}}^+$) then $\mathcal{D}_{\boldsymbol{v}} = \mathcal{D}_{\boldsymbol{v}} \bigcup \{(\boldsymbol{x}_i, +1)\}$ endif;
8.      if ($S_i \subseteq \mathcal{Y}_{\boldsymbol{v}}^-$) then $\mathcal{D}_{\boldsymbol{v}} = \mathcal{D}_{\boldsymbol{v}} \bigcup \{(\boldsymbol{x}_i, -1)\}$ endif;
9.    endfor
10.   if $|\mathcal{D}_{\boldsymbol{v}}| \geq thr$ then
11.     $l = l + 1$;
12.     Set the $l$-th column of the coding matrix $\mathbf{M}$ to $\boldsymbol{v}$: $\mathbf{M}(:, l) = \boldsymbol{v}$;
13.     Build the binary classifier $h_l$ by invoking $\mathfrak{B}$ on $\mathcal{D}_{\boldsymbol{v}}$, i.e. $h_l \leftarrow \mathfrak{B}(\mathcal{D}_{\boldsymbol{v}})$;
14.   endif
15. End of While
16. Generate codeword $\boldsymbol{h}(\boldsymbol{x}^*)$ by querying binary classifiers' outputs: $\boldsymbol{h}(\boldsymbol{x}^*) = [h_1(\boldsymbol{x}^*), h_2(\boldsymbol{x}^*), \ldots, h_L(\boldsymbol{x}^*)]^\top$;
17. Return $y^* = f(\boldsymbol{x}^*)$ according to Eq.(3.1).
```

3.2 Partial Label Learning with ECOC As a well-established mechanism toward multi-class classifier induction, ECOC [11, 25] still follows the binary decomposition philosophy via a coding-decoding procedure. In the coding phase, a $q \times L$ binary coding matrix $\mathbf{M} \in \{+1, -1\}^{q \times L}$ is assumed to facilitate the learning process. Each row of the coding matrix $\mathbf{M}(j, :)$ represents an L-bits *codeword* for one class label y_j. On the other hand, each column of the coding matrix $\mathbf{M}(:, l)$ specifies a *dichotomy* over the label space $\mathcal{Y}$ with $\mathcal{Y}_l^+ = \{y_j \mid \mathbf{M}(j, l) = +1,\ 1 \leq j \leq q\}$ and $\mathcal{Y}_l^- = \{y_j \mid \mathbf{M}(j, l) = -1,\ 1 \leq j \leq q\}$. Based on this, one binary classifier $h_l : \mathcal{X} \to \{-1, +1\}$ is built for each column by treating training examples from $\mathcal{Y}_l^+$ as positive ones and those from $\mathcal{Y}_l^-$ as negative ones.

In the decoding phase, for any test example $\boldsymbol{x}^*$, an L-bits codeword $\boldsymbol{h}(\boldsymbol{x}^*)$ is generated by concatenating the predictive outputs of the L binary classifiers: $\boldsymbol{h}(\boldsymbol{x}^*) = [h_1(\boldsymbol{x}^*), h_2(\boldsymbol{x}^*), \ldots, h_L(\boldsymbol{x}^*)]^\top$. After that, the class label whose codeword is *closest* to $h(\boldsymbol{x}^*)$ is returned as the final prediction for $\boldsymbol{x}^*$:

$$f(\boldsymbol{x}^*) = \arg\min\nolimits_{y_j\ (1 \leq j \leq q)} \mathtt{dist}(h(\boldsymbol{x}^*), \mathbf{M}(j, :)) \tag{3.1}$$

Here, the distance function $\mathtt{dist}(\cdot, \cdot)$ can be implemented in various ways such as hamming distance [11], Euclidean distance [19], loss-based distance [1, 25], etc.

In this paper, we adapt the ECOC techniques to fit in the partial label learning scenario. Here, the key adaptation lies in how to build the binary classifier with respect to each column coding. Given any PL training example $(\boldsymbol{x}_i, S_i)$, we regard the candidate label set S_i associated with $\boldsymbol{x}_i$ as an entirety. Under this perspective, to build the binary classifier h_l for the l-th column of the coding matrix, $\boldsymbol{x}_i$ will be used as a positive or negative training example only if S_i entirely falls into $\mathcal{Y}_l^+$ or $\mathcal{Y}_l^-$. Otherwise, $\boldsymbol{x}_i$ will not contribute in the building process of h_l.

Table 1 summarizes the coding phase (Steps 1 to 15)

Table 2: Characteristics of the experimental data sets.

Controlled UCI Data Sets				`Configurations`
Data set	# Examples	# Features	# Class Labels	
abalone	4177	7	29	(I) $r = 1$, $p \in \{0.1, 0.2, \ldots, 0.7\}$ **[Figure 1]**
ecoli	336	7	8	(II) $r = 2$, $p \in \{0.1, 0.2, \ldots, 0.7\}$ **[Figure 2]**
pendigits	10992	16	10	(III) $r = 3$, $p \in \{0.1, 0.2, \ldots, 0.7\}$ **[Figure 3]**
segment	2310	18	7	(IV) $p = 1$, $r = 1$, $\epsilon \in \{0.1, 0.2, \ldots, 0.7\}$ **[Figure 4]**

Real-World Data Sets					
Data set	# Examples	# Features	# Class Labels	Avg. # CLs	Domain
Lost	1122	108	16	2.23	*automatic face naming* [9, 10]
BirdSong	4998	38	13	2.18	*bird song classification* [4, 17]
MSRCv2	1758	48	23	3.16	*object classification* [17, 22]

and the decoding phase (Steps 16 to 17) of the proposed PL-ECOC approach in pseudo-code form.

During the encoding phase, each iteration of the `while` loop tries to instantiate one column of the coding matrix $\mathbf{M}$. Firstly, one potential column coding $\boldsymbol{v}$ is generated at random (Step 3), based on which a dichotomization $\mathcal{Y}_{\boldsymbol{v}}^{+}$ and $\mathcal{Y}_{\boldsymbol{v}}^{-}$ over the label space is specified (Step 4). After that, a binary training set $\mathcal{D}_{\boldsymbol{v}}$ is created by picking up PL examples whose candidate label sets are fully enclosed by either $\mathcal{Y}_{\boldsymbol{v}}^{+}$ or $\mathcal{Y}_{\boldsymbol{v}}^{-}$ (Steps 5 to 9). To avoid non-informative training set with few training examples, a thresholding parameter (i.e. *thr*) is used to control the minimum admissible training set size (Step 10). Once the thresholding condition is satisfied, the potential codeword $\boldsymbol{v}$ is accepted (Step 11) and then used to instantiate a new column of the coding matrix $\mathbf{M}$ (Step 12) as well as build the corresponding binary classifier h_l (Step 13). The `while` loop terminates when the required number of codewords (i.e. L) have been generated (Step 2).[3]

During the decoding phase, following the standard ECOC procedure, the class label whose codeword is closest to the codeword of the test example (i.e. $\boldsymbol{h}(\boldsymbol{x}^*)$) is returned as the final prediction (Steps 16 to 17). In this paper, the widely-used hamming distance is chosen to implement the distance function $\texttt{dist}(\cdot, \cdot)$ between two binary codewords, i.e. $\texttt{dist}(\boldsymbol{u}, \boldsymbol{v}) = \sum_{l=1}^{L} [\![u_l \neq v_l]\!]$.

As shown in Table 1, PL-ECOC does not rely on any disambiguation strategy toward the candidate label set which is instead treated in an integrative manner. Similar to the standard ECOC mechanism, the working process of PL-ECOC is conceptually simple and amenable to different choices of the binary learner $\mathfrak{B}$. Furthermore, as reported in the next section, the performance of PL-ECOC is highly competitive against peer partial label learning approaches.

4 Experiments

4.1 Experimental Settings To evaluate the performance of the proposed approach, two series of experiments are conducted on controlled UCI data sets [3] as well as real-world partial label data sets. Table 2 summarizes characteristics of the experimental data sets.

Given any multi-class UCI data set, an artificial partial label data set is generated under different configurations of the controlling parameters p, r and ϵ. Here, p controls the proportion of examples which are partially labeled (i.e. $|S_i| \neq 1$), r controls the number of additional candidate labels together with the ground-truth label (i.e. $|S_i| = r + 1$), and ϵ controls the co-occurring probability between one extra candidate label and the ground-truth label [10, 17]. Table 1 lists the parameter configurations considered in this paper.

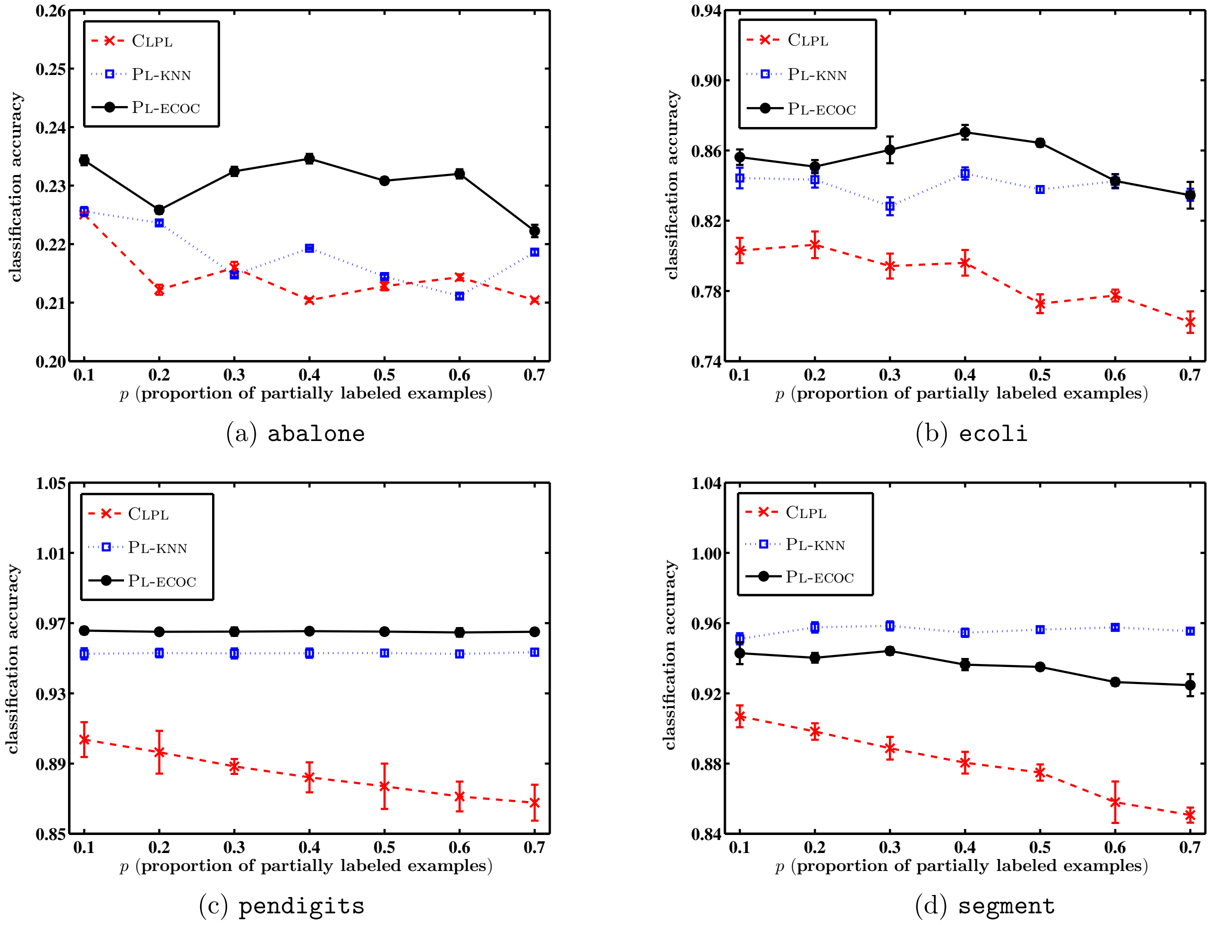
The `Lost` data set[4] contains 1122 faces for 16 persons cropped from the `Lost` TV series, where each face is described by 108 PCA components with candidate labels being names extracted from the associating screenplays [9, 10]. The `BirdSong` data set[5] contains 4998 singing syllables recorded from 13 bird species, where each syllable is described by 38 features with candidate labels being bird species jointly singing during a 10-second period [4, 17]. The `MSRCv2` data set[6] contains 1758 image segmentations from 23 classes of objects, where each image segmentation is described by 48 his-

[3] In this paper, to accommodate for the rare case of endless `while` loop (i.e. being unable to generate a total of L codewords satisfying the thresholding condition), the loop will terminate once a maximum number of 1,000 iterations is reached.

[4] `http://www.timotheecour.com/tv_data/tv_data.html`

[5] `http://web.engr.oregonstate.edu/~briggsf/`

[6] `http://research.microsoft.com/en-us/projects/objectclassrecognition/`



(a) abalone

(b) ecoli

(c) pendigits

(d) segment

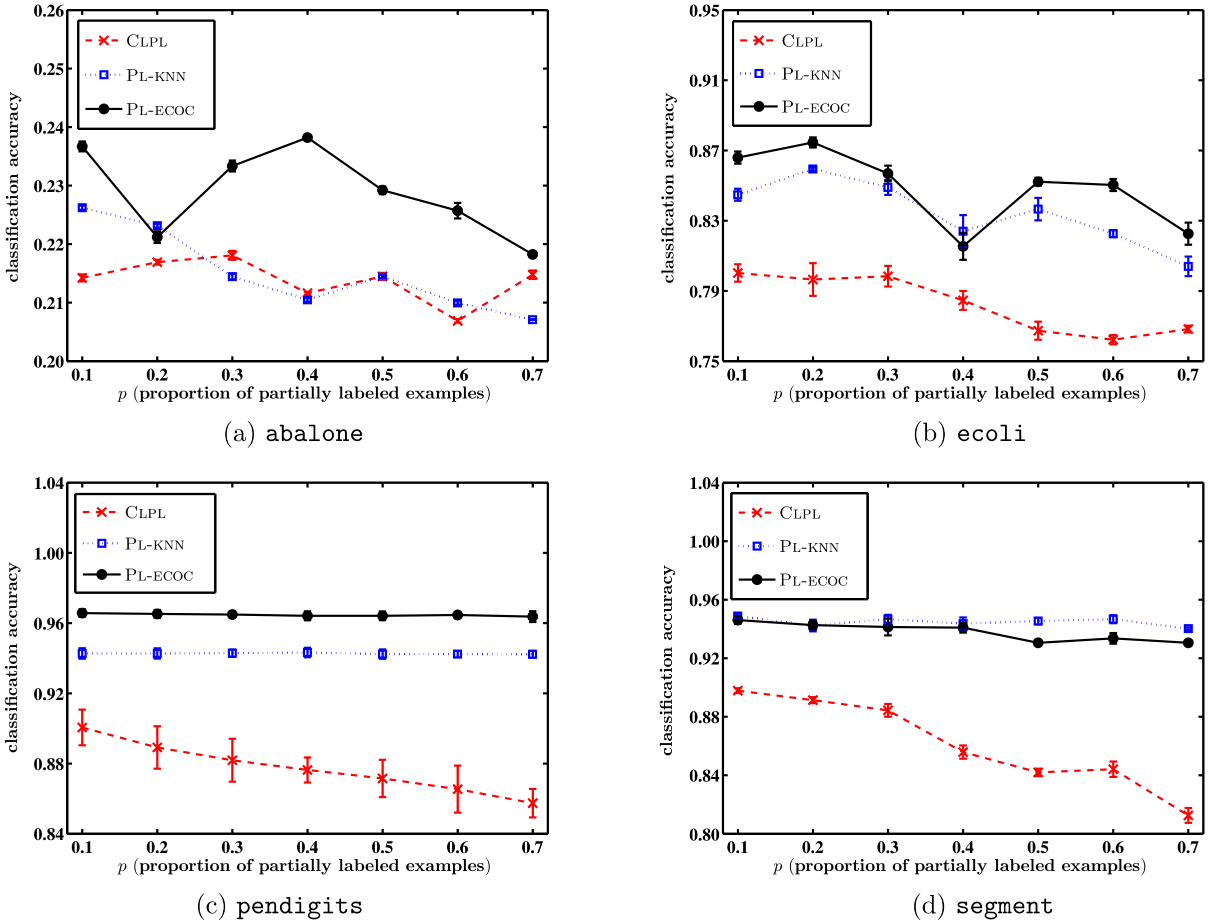
Figure 1: Classification accuracy of each comparing algorithm changes as p (proportion of partially labeled examples) increases with one additional candidate label ($r = 1$).

togram and gradient features with candidate labels being objects appearing within the same image [17, 22]. Table 2 also shows the average number of candidate labels (Avg. # CLs) for each real-world data set.

In this paper, the performance of PL-ECOC is compared with two state-of-the-art partial label learning approaches. One is the parametric approach CLPL (Convex Learning from Partial Labels) [10] and another is the non-parametric approach PL-KNN (Partial Label learning with k Nearest Neighbor) [14]. Parameters suggested in respective literatures are employed to implement CLPL (SVM with squared hinge loss) and PL-KNN ($k = 5$ with weighted voting). For PL-ECOC, the binary learner $\mathfrak{B}$ is set to be Libsvm [5] and the thresholding parameter *thr* is set to be one-tenth of the training set size (i.e. $\frac{1}{10} \cdot |\mathcal{D}|$).

4.2 Controlled UCI Data Sets Figures 1 to 3 illustrate the classification accuracy of each comparing algorithm with different number of additional candidate labels (r= 1, 2, 3) respectively. As usually adopted by ECOC-based techniques [1, 19, 25], the codeword length L is set to be $\lceil 10 \cdot \log_2(q) \rceil$ for PL-ECOC. In each figure, the proportion of partially labeled examples in the data set (i.e. p) is configured to increase from 0.1 to 0.7 with step-size 0.1. For any partially labeled example, r class labels in $\mathcal{Y}$ other than the ground-truth one will be randomly picked up to serve as its additional candidate labels. For each (r, p) configuration, ten-fold cross-validation is conducted on the corresponding data set and the average classification accuracy is recorded.

As illustrated in Figures 1 to 3, the performance of PL-ECOC is superior or comparable to the other algorithms in most cases. Specifically, Table 3 summarizes the win/tie/loss counts between PL-ECOC and the comparing algorithms based on pairwise t-tests at 0.05 significance level. Among the 84 statistical comparisons (21 (r, p)-configurations × 4 data sets), it is



(a) `abalone`

(b) `ecoli`

(c) `pendigits`

(d) `segment`

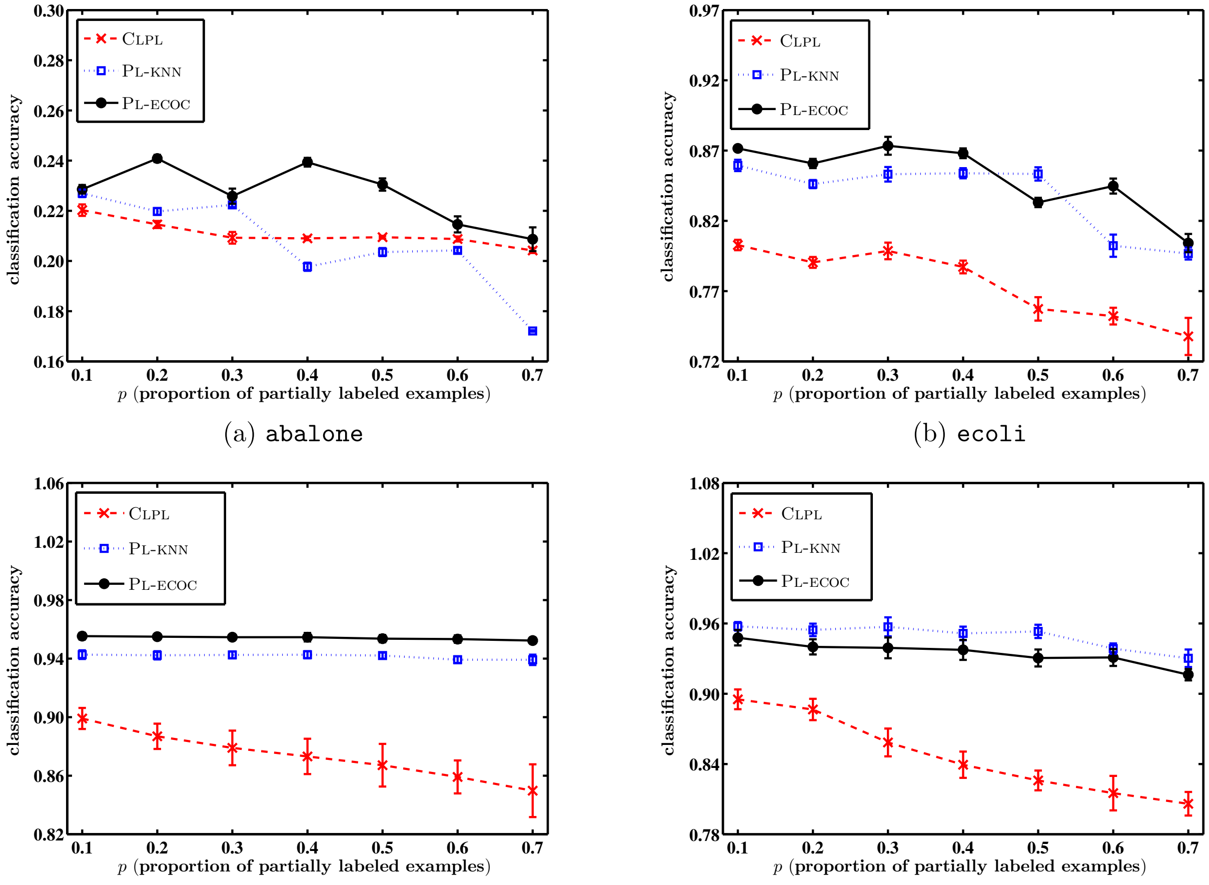
Figure 2: Classification accuracy of each comparing algorithm changes as p (proportion of partially labeled examples) increases with two additional candidate labels ($r = 2$).

Table 3: Win/tie/loss counts (pairwise t-tests at 0.05 significance level) on the classification performance of PL-ECOC against CLPL and PL-KNN.

	PL-ECOC against	
	CLPL	PL-KNN
$r = 1$, varying p	**22/6/0**	**8/14/6**
$r = 2$, varying p	**21/7/0**	**8/17/3**
$r = 3$, varying p	**24/4/0**	**10/13/5**
$p = 1$, $r = 1$, varying ϵ	**21/7/0**	**8/17/3**

shown that: (i) PL-ECOC achieves superior performance against CLPL in 79.7% cases and has not been outperformed by CLPL in any case; (ii) PL-ECOC achieves superior or at least comparable performance against PL-KNN in 83.3% cases and has been outperformed by PL-KNN in only 16.7% cases. Furthermore, it is interesting to find that on the `pendigits` data set (the largest one with 10,992 examples), the performance of PL-ECOC is rather stable as being barely influenced by increasing values of p. This observation suggests that when the number of training examples is abundant, the binary training set identified for each column of the coding matrix (Table 1, Steps 5 to 9) would be sufficiently large for robust classifier induction.

Figure 4 illustrates the classification accuracy of each comparing algorithm as the co-occurring probability between one extra candidate label and the ground-truth label varies. For $p = 1$ and $r = 1$, the co-occurring probability ϵ is configured to increase from 0.1 to 0.7 with step-size 0.1. For each class label $y \in \mathcal{Y}$, a co-occurring label y' is uniquely specified. The co-occurring label to the ground-truth one is chosen as the extra candidate label with probability ϵ, otherwise any other class label would be chosen as candidate label randomly. Similarly, ten-fold cross-validation is conducted



(a) `abalone`

(b) `ecoli`

(c) `pendigits`

(d) `segment`

Figure 3: Classification accuracy of each comparing algorithm changes as p (proportion of partially labeled examples) increases with three additional candidate labels ($r = 3$).

under each configuration of ϵ.

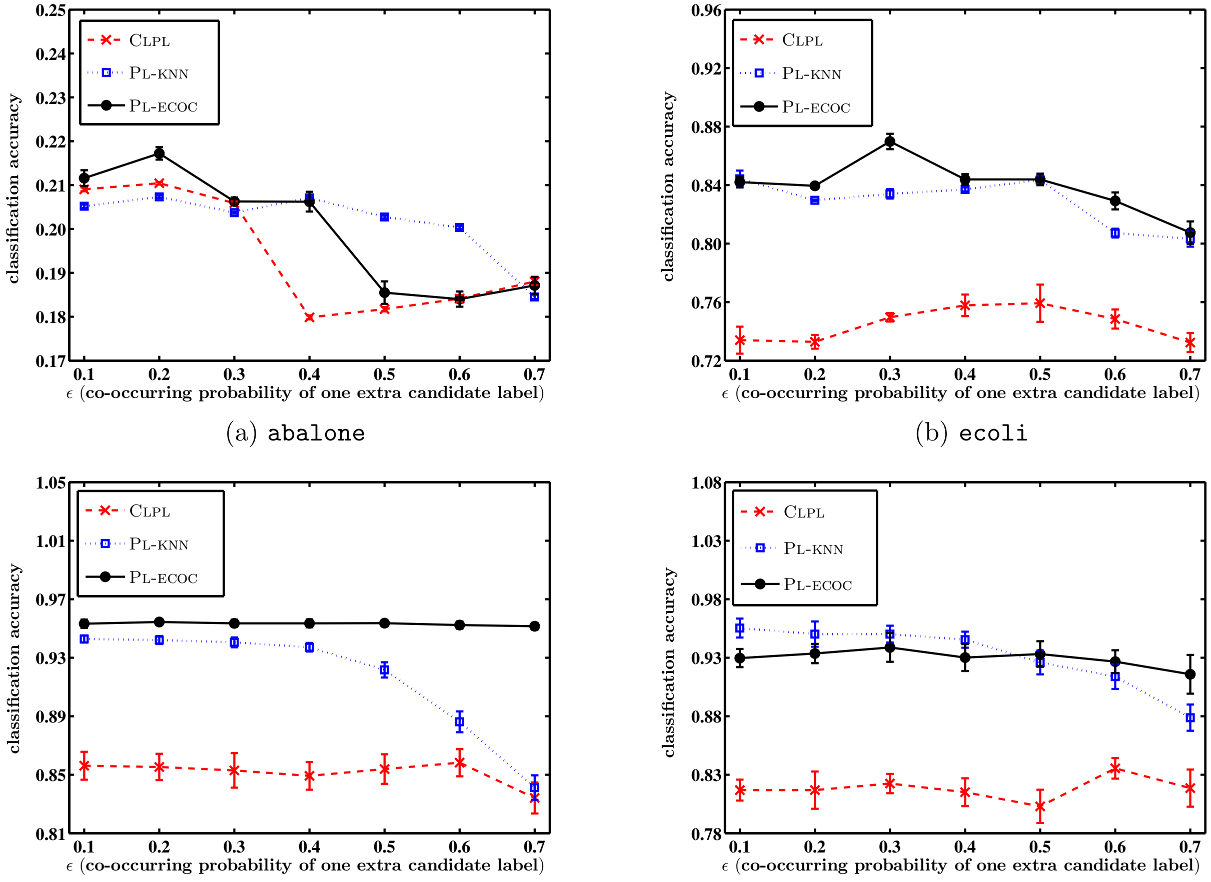
As illustrated in Figure 4, the performance of PL-ECOC is also superior or comparable to the other algorithms in most cases. Among the 28 statistical comparisons (7 ϵ-configuration × 4 data sets), it is shown in Table 3 that: (i) PL-ECOC achieves superior performance against CLPL in 75.0% cases and has not been outperformed by CLPL in any case; (ii) PL-ECOC achieves superior or at least comparable performance against PL-KNN in 89.3% cases and has been outperformed by PL-KNN in only 10.7% cases. Furthermore, on the `pendigits` data set, the performance advantage of PL-ECOC is more pronounced as the co-occurring probability ϵ increases.

4.3 Real-World Data Sets Table 4 reports the classification accuracy of each comparing algorithm on the real-world partial label data sets. For PL-ECOC, we set the codeword length $L = \lceil 100 \cdot \log_2(q) \rceil$ to achieve comparable results. One each data set, ten-fold cross-validation is conducted where the mean accuracy as well as the standard deviation are recorded for comparison. The highest accuracies among the comparing algorithms are shown in boldface.

It is shown in Table 4 that: (i) No algorithm can excel the other algorithms on all three data sets, while PL-ECOC does perform robustly without being ranked last in any case; (ii) Pairwise t-tests at 0.05 significance level show that PL-ECOC performs superiorly on `BirdSong` and `MSRCv2` while performs inferiorly on `Lost` against CLPL. Correspondingly, PL-ECOC performs superiorly on `Lost` and `BirdSong` while performs comparably on `MSRCv2` against PL-KNN.

5 Conclusion

In this paper, the problem of partial label learning is studied where a new disambiguation-free approach named PL-ECOC is proposed. The classical ECOC



(a) `abalone`

(b) `ecoli`

(c) `pendigits`

(d) `segment`

Figure 4: Classification accuracy of each comparing algorithm changes as ϵ (co-occurring probability of one extra candidate label) increases ($p = 1$, $r = 1$).

techniques are adapted toward solving this problem by treating the candidate label set of each PL example as an entirety. The effectiveness of the proposed approach is validated via comparative experiments on controlled UCI data sets as well as three real-world data sets.

Note that some PL training examples might be excluded from PL-ECOC's building process for the binary classifiers (Table 1, Step 13). Therefore, an important future work is to explore effective ways to make full use of those excluded PL examples. Furthermore, it is also interesting to investigate how PL-ECOC would work under various settings, e.g. by increasing the ECOC codeword length L, by incorporating binary learners $\mathfrak{B}$ other than SVM, or by employing distance function $\texttt{dist}(\cdot,\cdot)$ other than hamming distance, etc.

Acknowledgements

The author wishes to thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by the National Science Foundation of China (61175049, 61222309), the MOE Program for New Century Excellent Talents in University (NCET-13-0130), and the Fundamental Research Funds for the Central Universities (the Cultivation Program for Young Faculties of Southeast University).

References

[1] Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of Machine Learning Research **1**(Dec), 113–141 (2000)

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

[3] Bache, K., Lichman, M.: UCI machine learning repository. School of Information and Computer Sciences, University of California, Irvine (2013). URL `[http://archive.ics.uci.edu/ml]`

Table 4: Classification accuracy (mean± std. deviation) of each comparing algorithm on the real-world partial label data sets. The best performance on each data set is shown in bold face.

	Pl-ecoc	Clpl	Pl-knn
Lost	0.651±0.034	**0.742±0.042**	0.448±0.053
BirdSong	**0.743±0.019**	0.620±0.026	0.637±0.022
MSRCv2	**0.445±0.049**	0.386±0.046	0.443±0.027

[4] Briggs, F., Fern, X.Z., Raich, R.: Rank-loss support instance machines for MIML instance annotation. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 534–542. Beijing, China (2012)

[5] Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**(3), Article 27 (2011)

[6] Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-Supervised Learning. MIT Press, Cambridge, MA (2006)

[7] Côme, E., Oukhellou, L., Denoeux, T., Aknin, P.: Mixture model estimation with soft labels. In: D. Dubois, M.A. Lubiano, H. Prade, M.A. Gil, P. Grzegorzewski, O. Hryniewicz (eds.) Advances in Soft Computing 48, pp. 165–174. Springer, Berlin (2008)

[8] Côme, E., Oukhellou, L., Denoeux, T., Aknin, P.: Learning from partially supervised data using mixture models and belief functions. Pattern Recognition **42**(3), 334–348 (2009)

[9] Cour, T., Sapp, B., Jordan, C., Taskar, B.: Learning from ambiguously labeled images. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 919–926. Miami, FL (2009)

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

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

[12] Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple-instance problem with axis-parallel rectangles. Artificial Intelligence **89**(1-2), 31–71 (1997)

[13] Grandvalet, Y., Bengio, Y.: Learning from partial labels with minimum entropy. Tech. rep., Center for Interuniversity Research and Analysis of Organizations, Québec, Canada (2004)

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

[15] Jie, L., Orabona, F.: Learning from candidate labeling sets. In: J. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, A. Culotta (eds.) Advances in Neural Information Processing Systems 23, pp. 1504–1512. MIT Press, Cambridge, MA (2010)

[16] Jin, R., Ghahramani, Z.: Learning with multiple labels. In: S. Becker, S. Thrun, K. Obermayer (eds.) Advances in Neural Information Processing Systems 15, pp. 897–904. MIT Press, Cambridge, MA (2003)

[17] Liu, L., Dietterich, T.: A conditional multinomial mixture model for superset label learning. In: P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (eds.) Advances in Neural Information Processing Systems 25, pp. 557–565. MIT Press, Cambridge, MA (2012)

[18] Nguyen, N., Caruana, R.: Classification with partial labels. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 381–389. Las Vegas, NV (2008)

[19] Pujol, O., Escalera, S., Radeva, P.: An incremental node embedding technique for error correcting output codes. Pattern Recognition **41**(2), 713–725 (2008)

[20] Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: O. Maimon, L. Rokach (eds.) Data Mining and Knowledge Discovery Handbook, pp. 667–686. Springer, Berlin (2010)

[21] Vannoorenberghe, P., Smets, P.: Partially supervised learning by a credal EM approach. In: L. Godo (ed.) Lecture Notes in Computer Science 3571, pp. 956–967. Springer, Berlin (2005)

[22] Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: Proceedings of the 10th IEEE International Conference on Computer Vision, pp. 1800–1807. Beijing, China (2005)

[23] Zeng, Z., Xiao, S., Jia, K., Chan, T.H., Gao, S., Xu, D., Ma, Y.: Learning by associating ambiguously labeled images. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 708–715. Portland, OR (2013)

[24] Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering (in press)

[25] Zhou, Z.H.: Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC, Boca Raton, FL (2012)

[26] Zhu, X., Goldberg, A.B.: Introduction to semi-supervised learning. In: R.J. Brachman, T.G. Dietterich (eds.) Synthesis Lectures to Artificial Intelligence and Machine Learning, pp. 1–130. Morgan & Claypool Publishers, San Francisco, CA (2009)