Disambiguation-Free Partial Label Learning

(非消歧偏标记学习)

Min-Ling Zhang

School of Computer Science and Engineering,

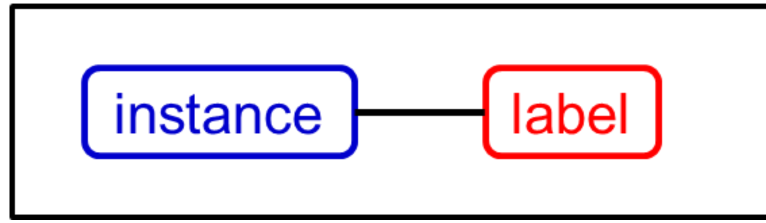
MOE Key Lab. of Computer Network & Information Integration

Southeast University, China

April 21, Xiamen

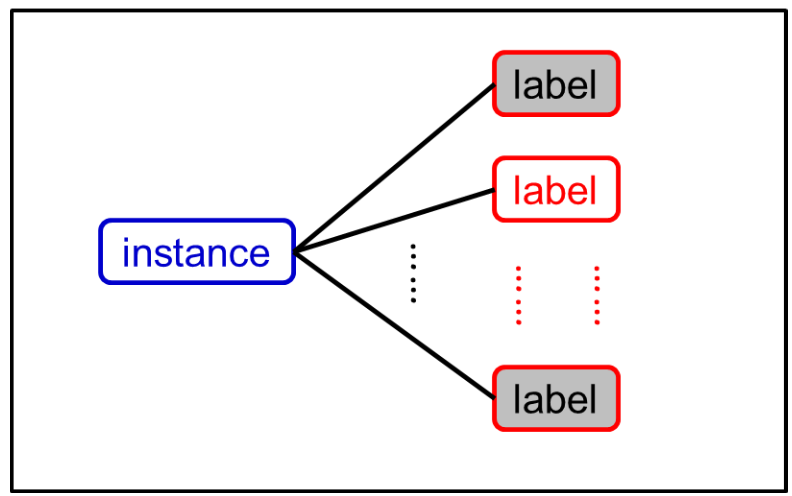
Partial Label (PL) Learning

object



Traditional Supervised Learning

object



Partial Label Learning (PLL)

- Multiple candidate labels
- Only one valid (but unknown)

The Problem

Difficulty: weak supervision

The ground-truth label of the PL training example is **concealed** in its candidate label set

Common strategy: disambiguation

Try to **disambiguate** the set of candidate labels

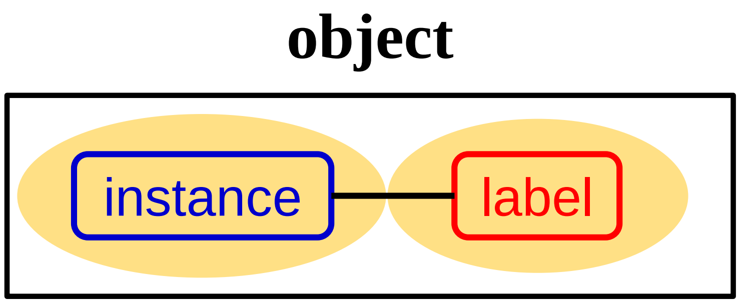
➔ Prone to be misled by the *false positive* label(s)

Question: Are there other strategies of learning from PL examples without relying on disambiguating the candidate label set?

Outline

- Introduction
- The PL-ECOC Approach
- Experiments
  - Controlled UCI Data Sets
  - Real-world Data Sets
- Conclusion

Traditional Supervised Learning

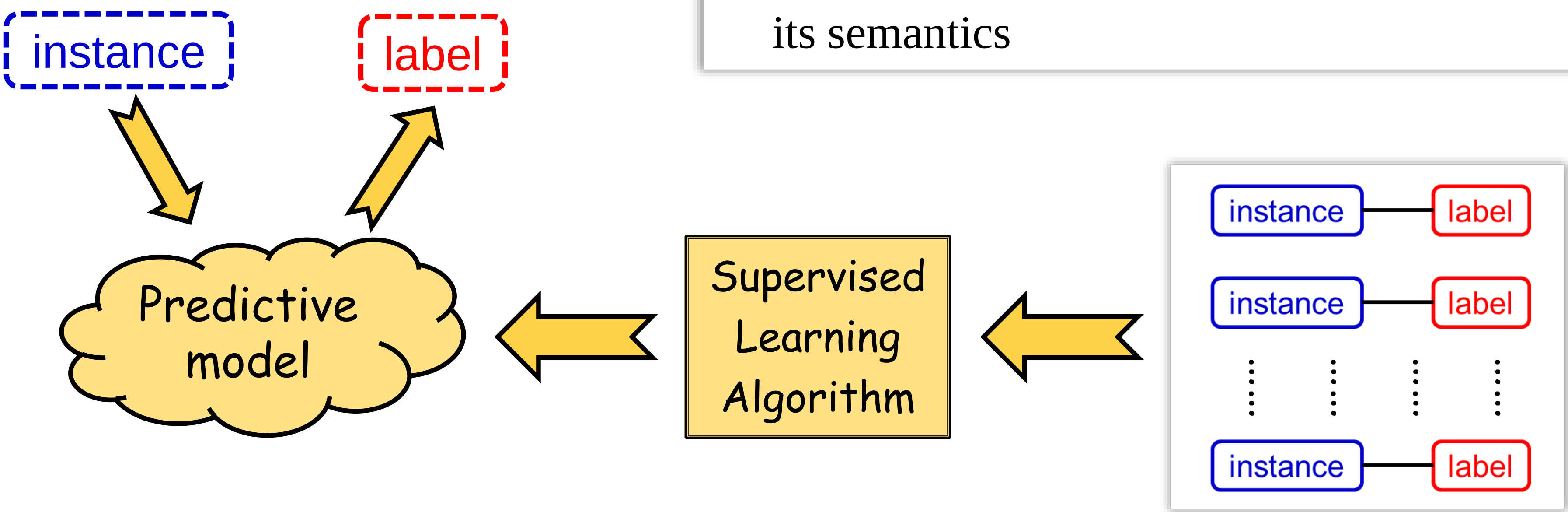


Input Space

represented by a single instance (feature vector) characterizing its properties

Output Space

associated with a single label characterizing its semantics



Basic Assumption: Strong Supervision

label ⟺ supervision information

Key factor for successful learning

(encoding *semantics* and *regularities* for the learning problem)

Strong supervision assumption

- **Sufficient labeling**
  abundant labeled training data are available

- **Explicit labeling**
  object labeling is unique and unambiguous

But, Supervision Is Usually Weak

Difficult to have!

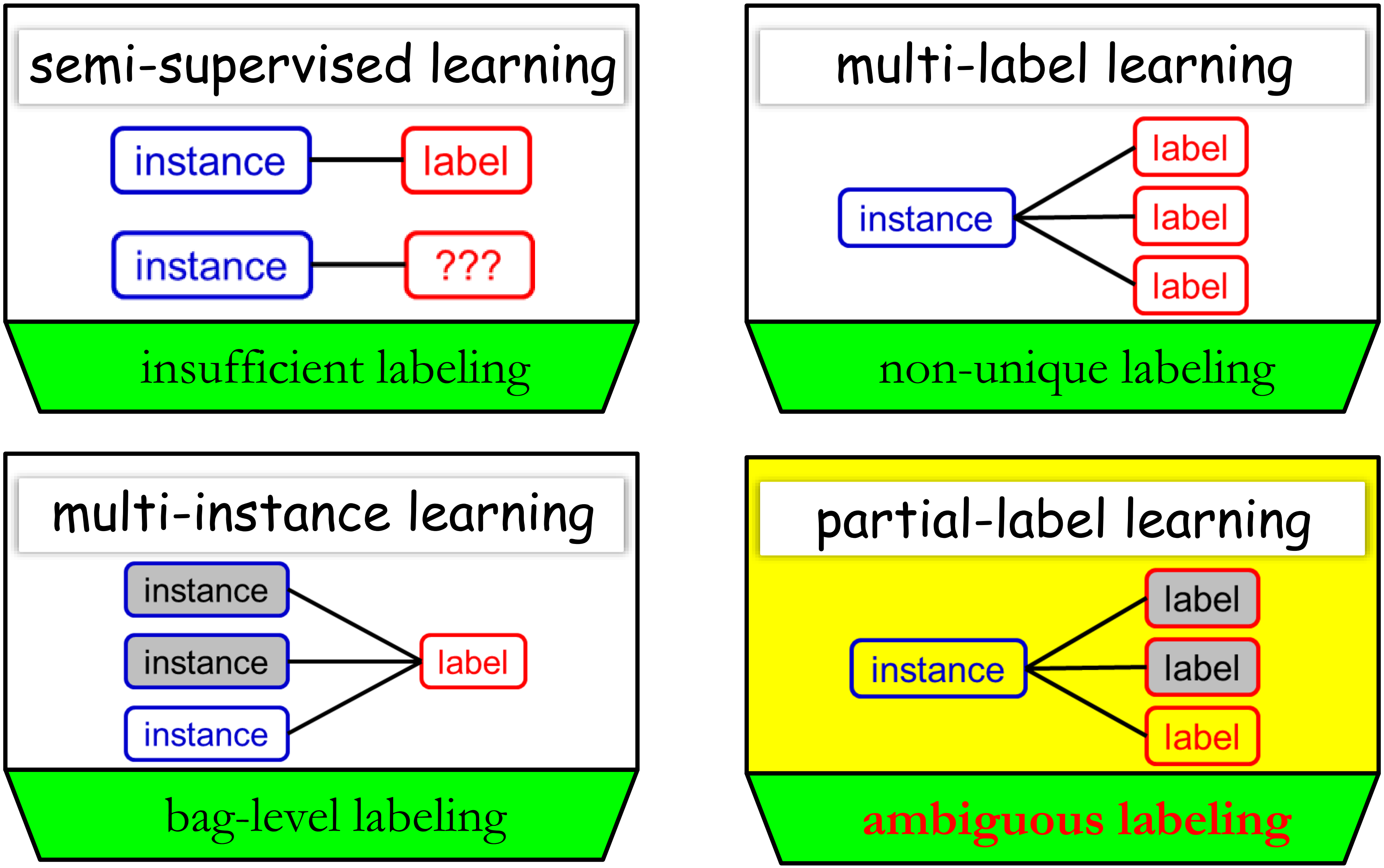
Constrained by:

- Limited resources
- Physical environment
- Problem properties
- ……

Strong supervision (sufficient & explicit) → **Strong generalization ability**

In practice, we usually have to learn with weak supervision

For Example…


semi-supervised learning
instance
label
instance
???
insufficient labeling
multi-label learning
instance
label
label
label
non-unique labeling
multi-instance learning
instance
instance
instance
label
bag-level labeling
partial-label learning
instance
label
label
label
ambiguous labeling

Partial Label

Appreciator A ------→ Picasso style ×

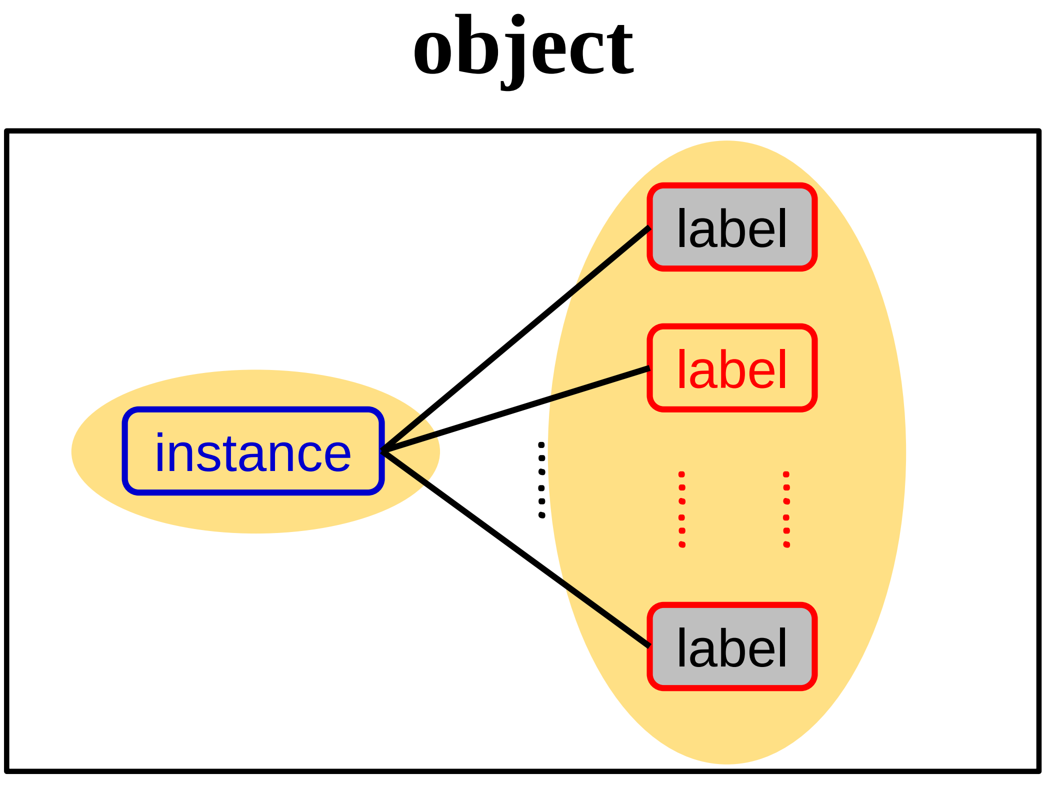
Appreciator B ------→ Monet style ×

Appreciator C ------→ van Gogh style √

Widely exist in real-world applications

- Computer vision [Cour et al., JMLR11] [Tang & Zhang, AAAI'17]
- Image classification [Zeng et al., CVPR'13] [Chen et al., CVPR'13]
- Learning from crowds [Raykar et al., JMLR10]
- Ecoinformatics [Liu & Dietterich, NIPS'12] [Zhang & Yu, IJCAI'15]
- ……

Partial-Label Learning (PLL)



- Each object is associated with multiple candidate labels
- Only one of the candidate label is the unknown ground-truth label

Partial-Label Learning (PLL)

Partial Label vs. Unlabel/Multi-Label

Partial-label vs. Unlabel

Commonness: ground-truth label is unknown

Difference: ground-truth label is confined

Partial-label vs. Multi-label

Commonness: multiple labels being assigned

Difference: only one assigned label being valid

Outline

- Introduction
- The PL-ECOC Approach
- Experiments
  - Controlled UCI Data Sets
  - Real-world Data Sets
- Conclusion

Formal Definition of PLL

Settings

$\mathcal{X}$: d-dimensional feature space $\mathbb{R}^d$

$\mathcal{Y}$: label space with q labels $\{y_1, y_2, \cdots, y_q\}$

Inputs

$\mathcal{D}$: training set with m examples $\{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq m\}$

$\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector $(\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \cdots, \boldsymbol{x}_{id})^{\mathrm{T}}$

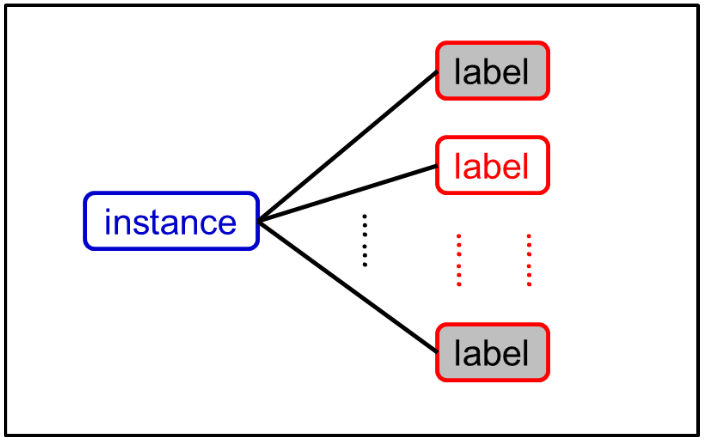
$Y_i \subseteq \mathcal{Y}$ is the candidate label set for $\boldsymbol{x}_i$, with its (unknown) ground-truth label $y_i \in Y_i$

Outputs

h : multi-class predictor $\mathcal{X} \rightarrow \mathcal{Y}$

Key Challenge

object



Ambiguous labeling

ground-truth label not accessible by the learning algorithm

Common strategy: Disambiguation

- Disambiguation by ground-truth label identification
- Disambiguation by candidate label averaging

Existing Approaches

Disambiguation by Identification

[Jin & Ghahramani, NIPS'03] [Nguyen & Caruana, KDD'08]
[Liu & Dietterich, NIPS'12] [Chen et al., CVPR'13] [Zhang et al., KDD'16]

treating the ground-truth label as latent variable → identified via iterative refining procedure such as EM

Potential weakness: the identified label may turn out to be the false positive label

Existing Approaches

Disambiguation by Averaging

[Hullermeier & Beringer, IDA06] [Cour et al., CVPR'09]
[Cour et al., JMLR11] [Zhang & Yu, IJCAI'15]

treating all the candidate labels in an equal manner → make final prediction by averaging their modeling outputs

Potential weakness: ground-truth output overwhelmed by false positive outputs

The PL-ECOC Approach

Goal of PLL Induce a multi-class predictor $h: \mathcal{X} \rightarrow \mathcal{Y}$

Popular Binary Decomposition

- ☐ **One-vs-Rest (#classifiers: q)**
- ☐ **One-vs-One (#classifiers: $q(q$-1)/2)**

Not applicable due to the unknown ground-truth label

PL-ECOC (Partial-label Learning with Error-Correcting Output Codes)

Two major advantages

- ☐ Naturally enable binary decomposition
- ☐ Disambiguation-free

The PL-ECOC Approach (Cont.)

Illustrative procedure of ECOC

	h_1	h_2	h_3	h_4	h_5	h_6	h_7
y_1	+1	-1	+1	+1	+1	-1	-1
y_2	+1	+1	-1	-1	+1	-1	+1
y_3	-1	+1	-1	+1	+1	+1	-1
y_4	+1	-1	-1	+1	-1	-1	+1
y_5	-1	-1	+1	-1	-1	+1	+1
$\boldsymbol{x}^*$	+1	-1	-1	+1	-1	+1	-1

For each multi-class example $(\boldsymbol{x}_i, y_i)$

- $h_1(\boldsymbol{x}_i) = +1$ if $y_i \in \{y_1, y_2, y_4\}$
- $h_1(\boldsymbol{x}_i) = -1$ if $y_i \in \{y_3, y_5\}$

Identify the class with closest codeword to test instance x^*

The PL-ECOC Approach (Cont.)

Illustrative procedure of PL-ECOC

	h_1	h_2	h_3	h_4	h_5	h_6	h_7
y_1	+1	-1	+1	+1	+1	-1	-1
y_2	+1	+1	-1	-1	+1	-1	+1
y_3	-1	+1	-1	+1	+1	+1	-1
y_4	+1	-1	-1	+1	-1	-1	+1
y_5	-1	-1	+1	-1	-1	+1	+1
$\boldsymbol{x}^*$	-1	+1	-1	+1	+1	-1	-1

For each partial-label example $(\boldsymbol{x}_i, Y_i)$

- $h_1(\boldsymbol{x}_i) = +1$ if $Y_i \subseteq \{y_1, y_2, y_4\}$
- $h_1(\boldsymbol{x}_i) = -1$ if $Y_i \subseteq \{y_3, y_5\}$
- ignored w.r.t. h_1 otherwise

make prediction in the same way as ECOC

The PL-ECOC Approach (Cont.)

Complete Pipeline of PL-ECOC

- Coding matrix generation

randomly generate a q-bits column vector $\boldsymbol{v}$ → derive the binary training set $B_{\boldsymbol{v}}$ from the PL training set $\xrightarrow{|B_v| \geq \tau}$ set $\boldsymbol{v}$ as a column coding of the coding matrix

Repeat until reaching the ECOC coding length *L*

- Binary classifier induction

  induce a total of L binary classifiers, one for each column coding

- Make prediction for unseen instance

  identify the class whose codeword is closest to the classifiers' outputs on unseen instance

Outline

- Introduction
- The PL-ECOC Approach
- Experiments
  - Experimental Setup
  - Controlled UCI Data Sets
  - Real-world Data Sets
- Conclusion

Experimental Setup

Comparing Algorithms

PL-ECOC : Coding length $= \lceil 10 \cdot \log_2(q) \rceil$; Base learner: Libsvm

averaging-based disambiguation

CLPL: Base learner: SVM with squared hinge loss

PL-*k*NN: # nearest neighbors = 5

identification-based disambiguation

PL-SVM: Regularization parameter pool $\{10^{-3},\ldots,10^{3}\}$

LSB-CMM: # mixture components = q

Experimental Protocol

Ten-fold cross-validation + Pairwise *t*-test

Controlled UCI Data Sets

Data set	# Examples	# Features	# Class Labels
Ecoli	336	7	8
Dermatology	364	23	6
Vehicle	846	18	4
Segment	2,310	18	7
Abalone	4,177	7	29
Satimage	6,435	36	7
Usps	9,298	256	10
Pendigits	10,992	16	10
Letter	20,000	16	26

Generating an **artificial** PL data set from an UCI data set with three controlling parameters p, r, ϵ

Controlled UCI Data Sets

Controlled UCI Data Sets

Data set	# Examples	# Features	# Class Labels
Ecoli	336	7	8
Dermatology	364	23	6
Vehicle	846	18	4
Segment	2,310	18	7
Abalone	4,177	7	29
Satimage	6,435	36	7
Usps	9,298	256	10
Pendigits	10,992	16	10
Letter	20,000	16	26

Generating an **artificial** PL data set from an UCI data set with three controlling parameters p, r, ϵ

p : Proportion of examples which are partially labeled $(|S_i| \neq 1)$

r : # false positive labels in candidate label set $(|S_i| = r + 1)$

ϵ : Co-occurring probability for one extra candidate label

Fix r (=1, 2, 3), varying $p \in \{0.1, \ldots, 0.7\}$

Fix r (=1), p (=1), varying $\epsilon \in \{0.1, \ldots, 0.7\}$

28 configurations per UCI data set

Controlled UCI Data Sets (Cont.)

TABLE 3
Win/tie/loss counts (pairwise t-test at 0.05 significance level) on the classification performance of PL-ECOC against each comparing algorithm on the controlled UCI data sets.

PL-ECOC against		Eco.	Der.	Veh.	Seg.	Aba.	Sat.	Usp.	Pen.	Let.	Subtotal	In Total
		Data Sets (names in abbreivation)										
PL-KNN	[Figure 1]	0/7/0	1/6/0	7/0/0	3/4/0	7/0/0	0/7/0	7/0/0	5/2/0	7/0/0	**37/26/0**	**156/96/0**
	[Figure 2]	0/7/0	3/4/0	7/0/0	2/5/0	7/0/0	0/7/0	7/0/0	7/0/0	5/2/0	**38/25/0**	
	[Figure 3]	0/7/0	2/5/0	7/0/0	4/3/0	7/0/0	1/6/0	7/0/0	7/0/0	5/2/0	**40/23/0**	
	[Figure 4]	2/5/0	3/4/0	7/0/0	2/5/0	7/0/0	3/4/0	7/0/0	6/1/0	4/3/0	**41/22/0**	
CLPL	[Figure 1]	0/7/0	0/7/0	6/1/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	**48/15/0**	**181/71/0**
	[Figure 2]	0/7/0	0/7/0	3/4/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	**45/18/0**	
	[Figure 3]	0/7/0	0/7/0	3/4/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	**45/18/0**	
	[Figure 4]	0/7/0	1/6/0	0/7/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	**43/20/0**	
PL-SVM	[Figure 1]	0/7/0	0/7/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	**49/14/0**	**195/57/0**
	[Figure 2]	0/7/0	0/7/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	**49/14/0**	
	[Figure 3]	0/7/0	0/7/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	**49/14/0**	
	[Figure 4]	0/7/0	0/7/0	6/1/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	**48/15/0**	
LSB-CMM	[Figure 1]	7/0/0	0/7/0	1/6/0	7/0/0	0/7/0	7/0/0	7/0/0	7/0/0	7/0/0	**43/20/0**	**179/73/0**
	[Figure 2]	7/0/0	0/7/0	1/6/0	7/0/0	0/7/0	7/0/0	7/0/0	7/0/0	7/0/0	**43/20/0**	
	[Figure 3]	7/0/0	0/7/0	4/3/0	7/0/0	0/7/0	7/0/0	7/0/0	7/0/0	7/0/0	**46/17/0**	
	[Figure 4]	7/0/0	2/5/0	1/6/0	7/0/0	2/5/0	7/0/0	7/0/0	7/0/0	7/0/0	**47/16/0**	

Out of 252 statistical tests (28 configurations x 9 UCI data sets)

- None of the comparing algorithms significantly outperformed PL-ECOC
- PL-ECOC outperforms PL-KNN and CLPL in 61.9% and 71.8% cases respectively
- PL-ECOC outperforms PL-SVM and LSB-CMM in 77.3% and 71.0% cases respectively

Real-World Data Sets

Real-World Data Sets

Data set	# Examples	# Features	# Class Labels	Avg. # CLs	Domain
Lost	1,122	108	16	2.23	*automatic face naming* [11]
MSRCv2	1,758	48	23	3.16	*object classification* [21]
BirdSong	4,998	38	13	2.18	*bird song classification* [4]
Soccer Player	17,472	279	171	2.09	*automatic face naming* [27]
LYN 10	18,313	163	11	2.02	*automatic face naming* [17]
LYN 20	19,027	163	21	2.01	
LYN 50	20,308	163	54	1.97	
LYN 100	21,390	163	101	1.94	
LYN 200	22,991	163	219	1.91	

automatic face naming

instance: face cropped from image/video

candidate labels: names extracted from associated captions/subtitles

object classification

instance: image segmentation

candidate labels: objects appearing within the same image

bird song classification

instance: singing syllable of the bird

candidate labels: bird species jointly singing within 10-seconds period

URL: http://cse.seu.edu.cn/PersonalPage/zhangml/Resources.htm#partial data

Real-World Data Sets (Cont.)

TABLE 4

Predictive accuracy (mean±std) of each comparing algorithm on the real-world PL data sets. In addition, •/∘ indicates whether the performance of PL-ECOC is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 0.05 significate level).

	PL-ECOC	PL-KNN	CLPL	PL-SVM	LSB-CMM
Lost	0.703±0.052	0.424±0.041•	0.742±0.038∘	0.729±0.040	0.707±0.055
MSRCv2	0.505±0.027	0.448±0.037•	0.413±0.039•	0.482±0.043	0.456±0.031•
BirdSong	0.740±0.016	0.614±0.024•	0.632±0.017•	0.663±0.032•	0.717±0.024•
Soccer Player	0.537±0.020	0.497±0.014•	0.368±0.010•	0.443±0.014•	0.525±0.015
LYN 10	0.694±0.010	0.460±0.012•	0.605±0.013•	0.692±0.009	0.703±0.010∘
LYN 20	0.697±.0.012	0.469±0.015•	0.585±0.010•	0.686±0.011•	0.702±0.011
LYN 50	0.694±0.008	0.472±0.014•	0.540±0.012•	0.666±0.002•	0.679±0.007•
LYN 100	0.680±0.012	0.459±0.010•	0.507±0.011•	0.655±0.010•	0.673±0.010
LYN 200	0.662±0.010	0.457±0.014•	0.462±0.009•	0.636±0.010•	0.648±0.007•

- On ***BirdSong***, ***LYN 50*** and ***LYN 200***, PL-ECOC is superior to all the comparing algorithms
- On ***Soccer Player***, ***LYN 20***, ***LYN 100*** and ***MSRCv2***, PL-ECOC is superior or at least comparable to all the comparing algorithms
- On ***Lost*** and ***LYN 10***, PL-ECOC is inferior to the comparing algorithms in only two cases (CLPL on *Lost*; LSB-CMM on *LYN 10*)

Sensitivity Analysis for Coding Length

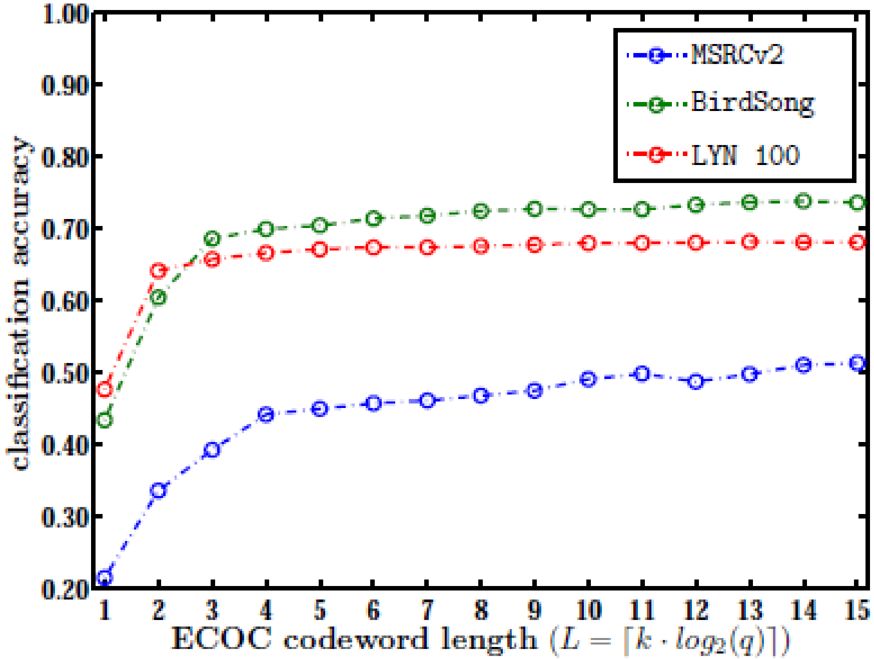


Fig. 5. Classification accuracy of PL-ECOC changes as the codeword length L increases from $\lceil log_2(q) \rceil$ to $\lceil 15 \cdot log_2(q) \rceil$ with step-size $\lceil log_2(q) \rceil$.

- Accuracy improves as the coding length increases
- Becomes stable as coding length approaches $\lceil 10 \cdot \log_2(q) \rceil$

Outline

- Introduction
- The PL-ECOC Approach
- Experiments
  - Controlled UCI Data Sets
  - Real-world Data Sets
- Conclusion

Conclusion

Main Contribution

Propose a new strategy to learn from partial label data, which is free of disambiguation

Key Technique

Treat the candidate label set as an entirety, and then adapt the ECOC procedure

Future Work

Investigate variants of PL-ECOC, new strategies to learn from partial label data, etc.