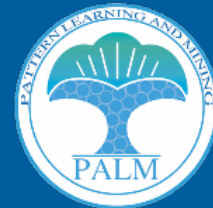


# Labeling Information Enhancement for Multi-label Learning with Low-rank Subspace

---

An Tao\*, Ning Xu, and Xin Geng

Southeast University, China



# Outline



- 1 **Introduction**
- 2 **The LIEMML Algorithm**
- 3 **Experiments**
- 4 **Conclusion**

# ① Introduction

# Introduction

---

## In traditional multi-label learning:



Features: each pixels in the picture

Label set:  $\mathcal{Y} =$   
 $\{sky, water, cloud, beach, plant, house\}$

Labeling information in multi-label learning is categorical in essence.

- Each label is regarded to be either relevant or irrelevant to the instance.

We call such label as *logical label*.

# Introduction

---

However...



Same logical label set  $\mathcal{Y} = \{sky, water, cloud, beach, plant, house\}$

Q: The two pictures can't be differed with only logical labels.

A: To describe the pictures better, we extend the logical label to be numerical.

This new label is called *numerical label*.

# Introduction

---

## Specification of *logical label*

Use  $\mathbf{y}_i \in \{-1, 1\}^t$  to denote the logical label vector.

- Label element of  $\mathbf{y}_i = 1$  : relevant to the instance.
- Label element of  $\mathbf{y}_i = -1$  : irrelevant to the instance.

## Specification of *numerical label*

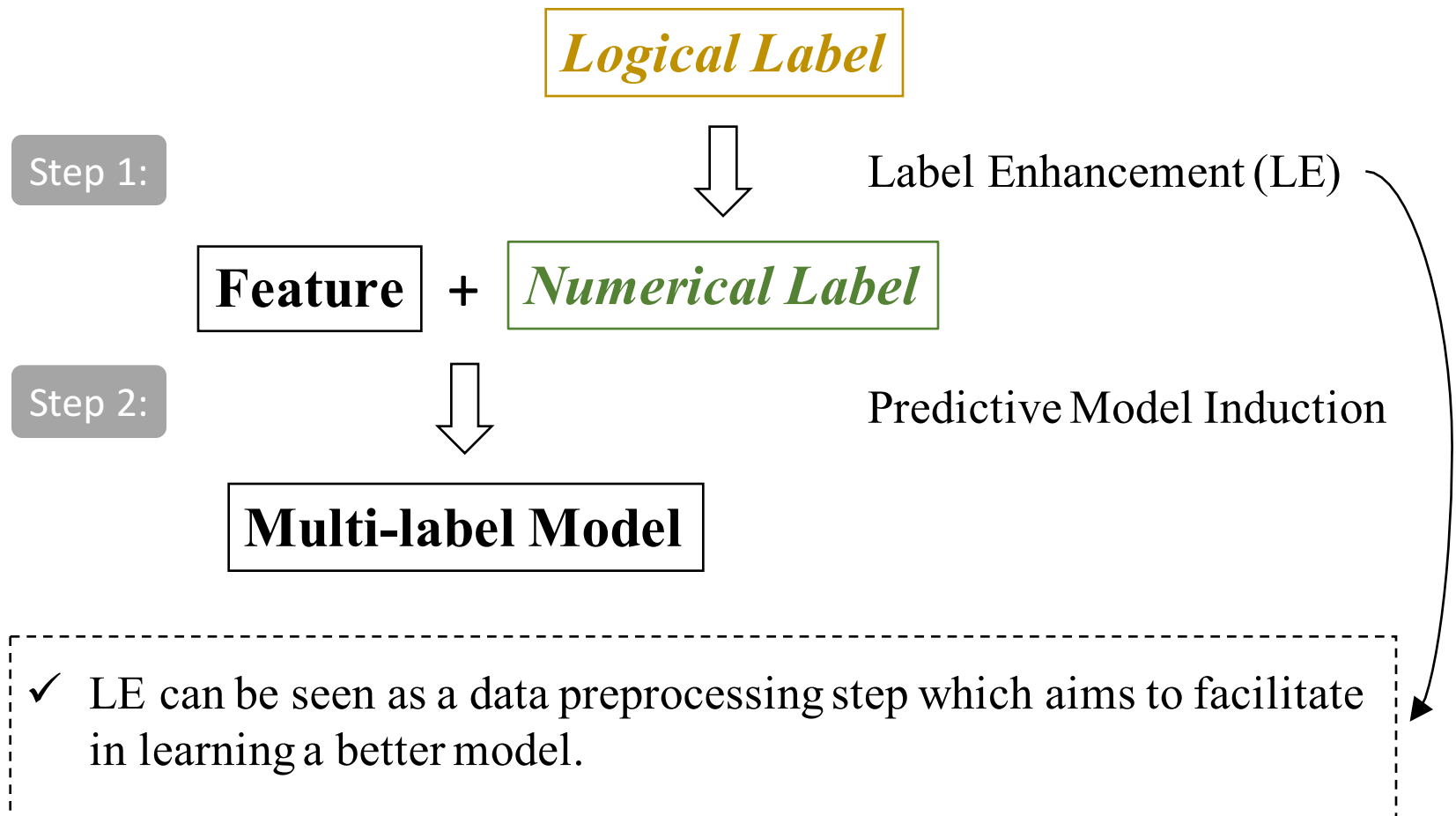
Use  $\mathbf{u}_i \in \mathbb{R}^t$  to denote the numerical label vector.

- Label element of  $\mathbf{u}_i > 0$  : relevant to the instance.
- Label element of  $\mathbf{u}_i < 0$  : irrelevant to the instance.
- Absolute value of label element of  $\mathbf{u}_i$  : reflects the degree to which the label describes the instance.

# Introduction

---

## Overview of our LIEML algorithm for multi-label learning



## ② The LIEMML Algorithm



# The LIEML Algorithm

---

## Symbol Definition:

- $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) | 1 \leq i \leq n\}$  : Dataset
- $\mathbf{x}_i \in \mathbb{R}^d$  :  $i$ -th instance vector
- $\mathbf{y}_i \in \{-1, 1\}^t$  :  $i$ -th logical label vector
- $\mathbf{u}_i \in \mathbb{R}^t$  :  $i$ -th numerical label vector

## Label Enhancement

Linear Model:

$$\mathbf{u}_i = \mathbf{W}^\top \mathbf{x}_i + \mathbf{b}$$

- $\mathbf{W} = [\mathbf{w}^1, \dots, \mathbf{w}^t]$  is a weight matrix.
- $\mathbf{b} \in \mathbb{R}^t$  is a bias vector.

# The LIEML Algorithm

---

## Label Enhancement

Linear Model:

$$u_i = \mathbf{W}^\top \mathbf{x}_i + b$$

- For convenient describing, we set  $\hat{\mathbf{W}} = [\mathbf{W}^\top, b]$ .

The model becomes:

$$\mathbf{U} = \hat{\mathbf{W}}[\mathbf{X}; \mathbf{1}^\top]$$

We then construct a stacked matrix  $\mathbf{Z}$ :

Target Matrix

$\mathbf{Z} = [\mathbf{Y}; \mathbf{X}; \mathbf{1}^\top]$

Label Enhancement

→

$[U; X'; \mathbf{1}^\top]$

# The LIEMML Algorithm

---

## Label Enhancement

$$\mathbf{Z} = [\mathbf{Y}, \mathbf{X}; \mathbf{1}^\top] \longrightarrow [\mathbf{U}, \mathbf{X}'; \mathbf{1}^\top]$$

The optimization problem of LE becomes:

$$\begin{aligned} \underset{\mathbf{Z} \in \mathbb{R}^{(t+d+1) \times n}}{\operatorname{argmin}} \quad & \boxed{L(\mathbf{Z})} + \boxed{R(\mathbf{Z})} \\ \text{s.t.} \quad & z_{(t+d+1)\cdot} = \mathbf{1}^\top \end{aligned}$$

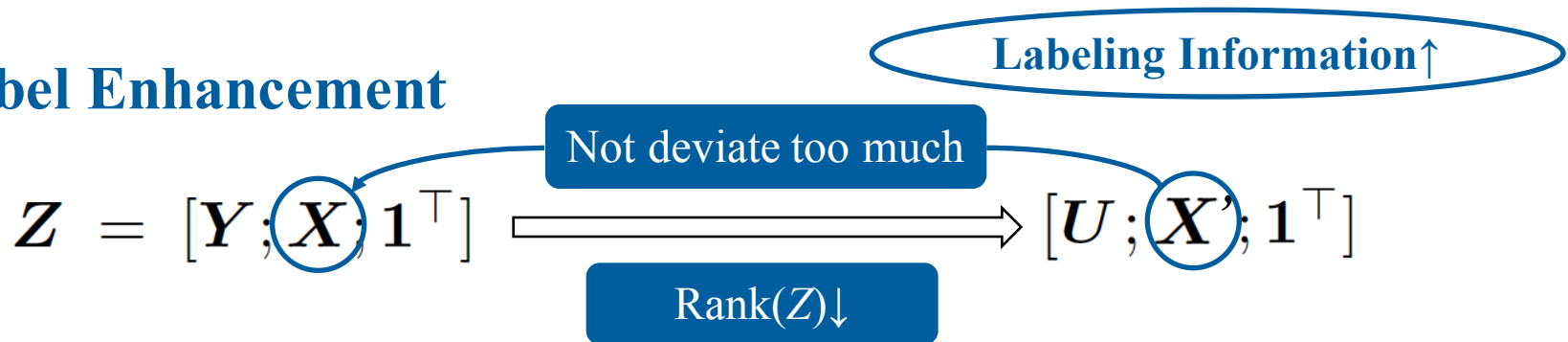
- $L(\mathbf{Z})$ : logistic loss function

$$L(\mathbf{Z}) = \frac{\lambda}{tn} \sum_{i=1}^t \sum_{j=1}^n \log(1 + e^{-y_{ij} z_{ij}})$$

✓  $L(\mathbf{Z})$  prevents the label values in  $\mathbf{Z}$  from deviating the original values too much.

# The LIEMML Algorithm

## Label Enhancement



- $R(Z)$ : nuclear norm  $\|Z\|_*$  and squared function

$$R(Z) = \boxed{\mu \|Z\|_*} + \boxed{\frac{1}{dn} \sum_{i=1}^d \sum_{j=1}^n \frac{1}{2} (z_{(i+t)j} - x_{ij})^2}$$

### Low-rank Assumption

Q: Why construct the stacked matrix  $Z$ ?

A: We assume that the stacked matrix  $Z$  belongs to an underlying low-rank subspace.

✓ The stacked matrix  $Z$  is therefore an underlying low-rank matrix.

# The LIEML Algorithm

---

## Label Enhancement



The target function  $T_1$  for optimization is yielded as:

$$\begin{aligned}
 & \underset{\mathbf{Z} \in \mathbb{R}^{(t+d+1) \times n}}{\operatorname{argmin}} \quad \boxed{L(\mathbf{Z})} + \boxed{R(\mathbf{Z})} \\
 & \text{s.t. } z_{(t+d+1)\cdot} = \mathbf{1}^\top
 \end{aligned}$$

$\Downarrow$

$$T_1(\mathbf{Z}) = \frac{\lambda}{tn} \sum_{i=1}^t \sum_{j=1}^n \log(1 + e^{-y_{ij} z_{ij}}) + \mu \|\mathbf{Z}\|_* + \frac{1}{dn} \sum_{i=1}^d \sum_{j=1}^n \frac{1}{2} (z_{(i+t)j} - x_{ij})^2$$


---


# The LIEML Algorithm

---

## Predictive Model Induction

We build the learning model through an adapted regressor based on MSVR.

The target function  $T_2(\Theta, \mathbf{m})$  we wish to minimize is:

$$T_2(\Theta, \mathbf{m}) = \frac{1}{2} \sum_{j=1}^t \|\boldsymbol{\theta}_j\|^2 + \gamma_1 \sum_{i=1}^n \boxed{\Omega_1(r_i)} + \gamma_2 \sum_{i=1}^n \sum_{j=1}^t \boxed{\Omega_2(q_{ij})}$$


$$\Omega_1(r) = \begin{cases} 0, & r < \varepsilon \\ r^2 - 2r\varepsilon + \varepsilon^2, & r \geq \varepsilon \end{cases}$$

$$\Omega_2(q) = -q\sigma(-q) = \begin{cases} 0, & q > 0 \\ -q, & q \leq 0 \end{cases}$$

- $r_i = \|\mathbf{e}_i\| = \sqrt{\mathbf{e}_i^\top \mathbf{e}_i}$ ,  $\mathbf{e}_i = \mathbf{u}_i - \varphi(\mathbf{x}_i)^\top \Theta - \mathbf{m}$ ,  $q_{ij} = y_{ij}(\varphi(\mathbf{x}_i)^\top \boldsymbol{\theta}_j + m_j)$ ,  
 $\Theta = [\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_t]$ ,  $\mathbf{m} = [m_1, \dots, m_t]$ .
- $\varphi(\mathbf{x})$  is a nonlinear transformation of  $\mathbf{x}$  to a higher dimensional feature space.

# The LIEML Algorithm

---

---

## Algorithm 1. LIEML

---

**Input:** The training set  $\mathcal{D} = \{(x_i, y_i) | 1 \leq i \leq n\}$ , parameters  $\mu_{min}, \eta_\mu, \lambda, \gamma_1, \gamma_2$ , step size  $\tau$ , convergence criterion  $\varepsilon$ ;

**Output:** Model parameters  $\beta$  and  $\mathbf{m}$ ;

```
1 Initial the stacked matrix  $Z^{(0)}$ ;
2 Determin  $\mu_1 > \mu_2 > \dots > \mu_L = \mu_{min} > 0$ ;
3 for each  $\mu \leftarrow \mu_1, \mu_2, \dots, \mu_L$  do
4    $k \leftarrow 0$ ;
5   repeat
6     Compute  $A^{(k)} \leftarrow Z^{(k)} - \tau g(Z^{(k)})$ ;
7     Compute SVD of  $U^{(k)} A^{(k)} (V^{(k)})^\top \leftarrow A^{(k)}$ ;
8     Compute  $Z^{(k+1)} \leftarrow U^{(k)} \max(A^{(k)} - \tau \mu, 0) (V^{(k)})^\top$ ;
9     Project  $Z^{(k+1)}$  to feasible region  $z_{(t+d+1):} \leftarrow \mathbf{1}^\top$ ;
10     $k \leftarrow k + 1$ ;
11  until  $|T_1(Z^{(k)}) - T_1(Z^{(k-1)})| < \varepsilon$ ;
12 end
13  $U \leftarrow z_{(1:t):}$ ;
14 Initial the model parameter  $\beta^{(0)}$  and  $\mathbf{m}^{(0)}$ ;
15 Compute target function  $T_2(\beta^{(0)}, \mathbf{m}^{(0)})$  by Eq. 8;
16  $k \leftarrow 0$ ;
17 repeat
18   Compute the descending directions  $\beta'$  and  $\mathbf{m}'$  by Eq. 15;
19   repeat
20     Compute the model parameter  $\beta^{(k+1)}$  through a line search algorithm
        combining  $\beta^{(k)}$  and  $\beta'$ ;
21     Compute the model parameter  $\mathbf{m}^{(k+1)}$  through a line search algorithm
        combining  $\mathbf{m}^{(k)}$  and  $\mathbf{m}'$ ;
22     Compute  $T_2(\beta^{(k+1)}, \mathbf{m}^{(k+1)})$  by Eq. 8;
23   until  $T_2(\beta^{(k+1)}, \mathbf{m}^{(k+1)}) < T_2(\beta^{(k)}, \mathbf{m}^{(k)})$ ;
24    $k \leftarrow k + 1$ ;
25 until  $|T_2(\beta^{(k)}, \mathbf{m}^{(k)}) - T_2(\beta^{(k-1)}, \mathbf{m}^{(k-1)})| < \varepsilon$ ;
```

---

# 3 Experiments



# Experiments

---

## Experiment Configuration

Ten benchmark multi-label data sets:

Datasets	$ S $	$\dim(S)$	$L(S)$	$F(S)$	$LCard(S)$	$LDen(S)$	$DL(S)$	$PDL(S)$	Domain
cal500	502	68	174	numeric	26.044	0.150	502	1.000	audio
emotion	593	72	6	numeric	1.868	0.311	27	0.046	audio
medical	978	1449	45	nominal	1.245	0.028	94	0.096	text
llog	1460	1004	75	nominal	1.180	0.016	304	0.208	text
enron	1702	1001	53	nominal	3.378	0.064	753	0.442	text
image	2000	294	5	numeric	1.236	0.247	20	0.010	image
scene	2407	294	5	numeric	1.074	0.179	15	0.006	image
yeast	2417	103	14	numeric	4.237	0.303	198	0.082	biology
slashdot	3782	1079	22	nominal	1.181	0.054	156	0.041	text
corel5k	5000	499	374	nominal	3.522	0.009	3175	0.635	image

Six well-established multi-label learning algorithms:

- BR, CLR, ECC, RAKEL, LP, and  $ML^2$

Five evaluation metrics widely-used in multi-label learning:

- Ranking-loss, One-error, Hamming-loss, Coverage, and Average precision

# Experiments

## Experimental Results

Comparing algorithm	Ranking-loss ↓				
	cal500	emotion	medical	llog	enron
LIEML	<b>0.177±0.002(1)</b>	0.221±0.011(2)	<b>0.026±0.005(1)</b>	0.144±0.008(2)	<b>0.076±0.002(1)</b>
BR	0.258±0.003(6)	0.233±0.016(6)	0.091±0.005(5)	0.328±0.007(6)	0.312±0.009(7)
CLK	0.239±0.026(5)	0.222±0.014(3)	0.123±0.026(7)	0.190±0.015(5)	0.089±0.002(2)
ECC	0.205±0.004(4)	0.227±0.017(4)	0.032±0.007(2)	0.154±0.009(3)	0.120±0.004(5)
RAKEL	0.444±0.005(7)	0.254±0.020(7)	0.095±0.033(6)	0.412±0.010(7)	0.241±0.005(6)
LP	0.181±0.003(2)	<b>0.182±0.012(1)</b>	0.034±0.006(4)	<b>0.125±0.005(1)</b>	0.091±0.003(4)
ML <sup>2</sup>	0.188±0.002(3)	0.231±0.012(5)	0.032±0.005(2)	0.158±0.005(4)	0.090±0.012(3)

Comparing algorithm	Ranking-loss ↓				
	image	scene	yeast	slashdot	corel5k
LIEML	<b>0.143±0.006(1)</b>	0.065±0.003(2)	0.170±0.002(2)	<b>0.093±0.002(1)</b>	0.121±0.002(2)
BR	0.314±0.014(7)	0.229±0.010(7)	0.190±0.004(4)	0.240±0.008(6)	0.416±0.003(6)
CLK	0.294±0.009(5)	0.127±0.003(4)	0.198±0.003(6)	0.260±0.007(7)	<b>0.114±0.002(1)</b>
ECC	0.276±0.005(4)	0.151±0.005(5)	0.190±0.003(4)	0.123±0.004(3)	0.292±0.003(5)
RAKEL	0.311±0.010(6)	0.205±0.008(6)	0.245±0.004(7)	0.190±0.005(5)	0.627±0.004(7)
LP	0.181±0.008(3)	0.087±0.006(3)	0.174±0.004(3)	0.132±0.005(4)	0.145±0.002(3)
ML <sup>2</sup>	<b>0.143±0.007(1)</b>	<b>0.064±0.003(1)</b>	<b>0.168±0.003(1)</b>	0.095±0.003(2)	0.163±0.003(4)

Comparing algorithm	One-error ↓				
	cal500	emotion	medical	llog	enron
LIEML	<b>0.119±0.014(1)</b>	0.350±0.023(2)	<b>0.166±0.011(1)</b>	0.766±0.020(3)	<b>0.225±0.011(1)</b>
BR	0.921±0.025(7)	0.375±0.027(6)	0.297±0.036(6)	0.884±0.011(6)	0.648±0.019(7)
CLK	0.331±0.111(6)	0.356±0.030(5)	0.688±0.143(7)	0.900±0.019(7)	0.376±0.017(4)
ECC	0.191±0.021(4)	0.353±0.040(4)	0.182±0.019(3)	0.785±0.009(4)	0.424±0.013(6)
RAKEL	0.286±0.039(5)	0.392±0.035(7)	0.208±0.071(4)	0.838±0.014(5)	0.412±0.016(5)
LP	0.120±0.015(2)	<b>0.303±0.027(1)</b>	0.213±0.021(5)	0.748±0.011(2)	0.311±0.013(3)
ML <sup>2</sup>	0.141±0.016(3)	0.352±0.021(3)	0.179±0.019(2)	<b>0.683±0.018(1)</b>	0.258±0.090(2)

Comparing algorithm	One-error ↓				
	image	scene	yeast	slashdot	corel5k
LIEML	<b>0.271±0.010(1)</b>	0.197±0.006(2)	<b>0.226±0.009(1)</b>	0.387±0.009(2)	0.650±0.006(2)
BR	0.538±0.019(7)	0.475±0.014(7)	0.285±0.008(7)	0.734±0.017(6)	0.919±0.006(7)
CLK	0.514±0.014(5)	0.371±0.008(4)	0.270±0.007(6)	0.979±0.003(7)	0.721±0.007(4)
ECC	0.486±0.018(4)	0.373±0.008(5)	0.256±0.007(5)	0.481±0.014(4)	0.699±0.006(3)
RAKEL	0.515±0.017(6)	0.444±0.012(6)	0.251±0.008(4)	0.453±0.005(3)	0.819±0.010(6)
LP	0.353±0.017(3)	0.270±0.016(3)	0.241±0.011(3)	0.558±0.009(5)	0.755±0.005(5)
ML <sup>2</sup>	0.272±0.009(2)	<b>0.194±0.008(1)</b>	0.228±0.009(2)	<b>0.382±0.009(1)</b>	<b>0.647±0.007(1)</b>

# Experiments

## Experimental Results

Comparing algorithm	<i>Hamming-loss</i> ↓				
	cal500	emotion	medical	llog	enron
LIEML	<b>0.136±0.001(1)</b>	0.251±0.006(3)	0.015±0.001(3)	<b>0.015±0.000(1)</b>	<b>0.047±0.001(1)</b>
BR	0.214±0.004(7)	0.265±0.013(5)	0.022±0.003(5)	0.052±0.003(7)	0.105±0.003(7)
CLK	0.165±0.005(5)	0.270±0.011(7)	0.024±0.002(6)	0.019±0.002(5)	0.072±0.002(6)
ECC	0.146±0.002(4)	0.254±0.013(4)	0.013±0.001(2)	0.016±0.000(2)	0.064±0.001(5)
RAKEL	0.138±0.002(2)	0.269±0.011(6)	<b>0.010±0.003(1)</b>	0.017±0.001(4)	0.058±0.001(3)
LP	0.167±0.004(6)	<b>0.223±0.007(1)</b>	0.017±0.001(4)	0.016±0.000(2)	0.063±0.003(4)
ML <sup>2</sup>	0.138±0.002(2)	0.243±0.010(2)	0.283±0.027(7)	0.021±0.001(6)	0.051±0.001(2)
Comparing algorithm	<i>Hamming-loss</i> ↓				
	image	scene	yeast	slashdot	corel5k
LIEML	0.160±0.003(2)	0.083±0.002(2)	<b>0.195±0.003(1)</b>	<b>0.040±0.001(1)</b>	<b>0.009±0.000(1)</b>
BR	0.287±0.008(6)	0.184±0.005(7)	0.219±0.003(6)	0.130±0.003(7)	0.027±0.000(7)
CLK	0.305±0.005(7)	0.181±0.004(6)	0.222±0.002(7)	0.058±0.001(5)	0.011±0.001(3)
ECC	0.244±0.005(4)	0.133±0.002(4)	0.216±0.002(5)	0.049±0.001(4)	0.015±0.001(5)
RAKEL	0.286±0.007(5)	0.171±0.005(5)	0.202±0.003(3)	0.048±0.001(3)	0.012±0.001(4)
LP	0.190±0.005(3)	0.127±0.005(3)	0.214±0.004(4)	0.060±0.002(6)	0.024±0.000(6)
ML <sup>2</sup>	<b>0.156±0.004(1)</b>	<b>0.076±0.003(1)</b>	0.196±0.003(2)	0.043±0.001(2)	0.010±0.001(2)
Comparing algorithm	<i>Coverage</i> ↓				
	cal500	emotion	medical	llog	enron
LIEML	<b>0.744±0.007(1)</b>	0.347±0.010(2)	<b>0.041±0.006(1)</b>	<b>0.149±0.007(1)</b>	<b>0.226±0.006(1)</b>
BR	0.852±0.014(6)	0.363±0.015(6)	0.118±0.007(6)	0.377±0.008(6)	0.601±0.014(7)
CLK	0.794±0.010(5)	0.351±0.016(3)	0.143±0.030(7)	0.225±0.016(5)	0.243±0.006(3)
ECC	0.788±0.008(4)	0.356±0.013(4)	0.048±0.009(2)	0.192±0.010(4)	0.300±0.009(5)
RAKEL	0.971±0.001(7)	0.381±0.019(7)	0.117±0.040(5)	0.459±0.011(7)	0.523±0.008(6)
LP	0.747±0.007(2)	<b>0.318±0.031(1)</b>	0.052±0.001(4)	0.159±0.006(2)	0.242±0.005(2)
ML <sup>2</sup>	0.780±0.008(3)	0.357±0.009(5)	0.048±0.008(2)	0.162±0.008(3)	0.256±0.017(4)
Comparing algorithm	<i>Coverage</i> ↓				
	image	scene	yeast	slashdot	corel5k
LIEML	<b>0.168±0.006(1)</b>	<b>0.068±0.003(1)</b>	0.454±0.004(2)	<b>0.109±0.002(1)</b>	0.276±0.004(2)
BR	0.301±0.012(7)	0.207±0.009(7)	0.474±0.005(4)	0.259±0.009(6)	0.758±0.003(6)
CLK	0.286±0.008(5)	0.120±0.007(3)	0.492±0.006(6)	0.272±0.007(7)	<b>0.267±0.004(1)</b>
ECC	0.272±0.005(4)	0.141±0.004(4)	0.476±0.004(5)	0.139±0.004(3)	0.562±0.007(5)
RAKEL	0.298±0.010(6)	0.186±0.006(6)	0.558±0.006(7)	0.212±0.005(5)	0.886±0.004(7)
LP	0.198±0.007(3)	0.171±0.009(5)	<b>0.451±0.005(1)</b>	0.148±0.005(4)	0.328±0.005(3)
ML <sup>2</sup>	<b>0.168±0.007(1)</b>	<b>0.067±0.003(1)</b>	0.454±0.004(2)	0.112±0.003(2)	0.372±0.006(4)

# Experiments

## Experimental Results

Comparing algorithm	Average precision $\uparrow$				
	cal500	emotion	medical	llog	enron
LIEML	<b>0.512<math>\pm</math>0.003(1)</b>	0.745 $\pm$ 0.011(2)	<b>0.872<math>\pm</math>0.011(1)</b>	0.347 $\pm$ 0.014(3)	<b>0.698<math>\pm</math>0.008(1)</b>
BR	0.300 $\pm$ 0.005(7)	0.730 $\pm$ 0.015(6)	0.762 $\pm$ 0.022(5)	0.215 $\pm$ 0.009(5)	0.381 $\pm$ 0.009(7)
CLK	0.395 $\pm$ 0.042(5)	0.742 $\pm$ 0.016(3)	0.400 $\pm$ 0.062(7)	0.194 $\pm$ 0.018(7)	0.610 $\pm$ 0.008(4)
ECC	0.463 $\pm$ 0.006(4)	0.740 $\pm$ 0.021(4)	0.860 $\pm$ 0.015(3)	0.342 $\pm$ 0.009(4)	0.559 $\pm$ 0.008(5)
RAKEL	0.353 $\pm$ 0.006(6)	0.717 $\pm$ 0.023(7)	0.700 $\pm$ 0.234(6)	0.197 $\pm$ 0.013(6)	0.539 $\pm$ 0.006(6)
LP	0.496 $\pm$ 0.005(3)	<b>0.779<math>\pm</math>0.012(1)</b>	0.837 $\pm$ 0.018(4)	0.390 $\pm$ 0.009(2)	0.661 $\pm$ 0.007(3)
ML <sup>2</sup>	0.501 $\pm$ 0.003(2)	0.737 $\pm$ 0.013(5)	0.865 $\pm$ 0.014(2)	<b>0.405<math>\pm</math>0.013(1)</b>	0.681 $\pm$ 0.053(2)

Comparing algorithm	Average precision $\uparrow$				
	image	scene	yeast	slashdot	corel5k
LIEML	<b>0.824<math>\pm</math>0.006(1)</b>	0.884 $\pm$ 0.004(2)	<b>0.766<math>\pm</math>0.005(1)</b>	0.708 $\pm$ 0.006(2)	<b>0.305<math>\pm</math>0.003(1)</b>
BR	0.649 $\pm$ 0.012(7)	0.692 $\pm$ 0.010(7)	0.734 $\pm$ 0.004(5)	0.427 $\pm$ 0.014(6)	0.123 $\pm$ 0.003(6)
CLK	0.666 $\pm$ 0.008(5)	0.778 $\pm$ 0.004(4)	0.730 $\pm$ 0.003(6)	0.250 $\pm$ 0.007(7)	0.274 $\pm$ 0.002(3)
ECC	0.685 $\pm$ 0.008(4)	0.766 $\pm$ 0.005(5)	0.741 $\pm$ 0.004(4)	0.628 $\pm$ 0.009(3)	0.264 $\pm$ 0.003(4)
RAKEL	0.661 $\pm$ 0.010(6)	0.713 $\pm$ 0.008(6)	0.720 $\pm$ 0.005(7)	0.617 $\pm$ 0.004(4)	0.122 $\pm$ 0.004(7)
LP	0.775 $\pm$ 0.009(3)	0.842 $\pm$ 0.009(3)	0.753 $\pm$ 0.006(3)	0.579 $\pm$ 0.009(5)	0.241 $\pm$ 0.002(5)
ML <sup>2</sup>	<b>0.824<math>\pm</math>0.006(1)</b>	<b>0.885<math>\pm</math>0.004(1)</b>	0.765 $\pm$ 0.005(2)	<b>0.711<math>\pm</math>0.005(1)</b>	0.297 $\pm$ 0.002(2)

**LIEML ranks 1<sup>st</sup> in the most cases!**

The model of LE in LIEML is linear, but nonlinear in ML<sup>2</sup>, it is uneasy for LIEML to beat ML<sup>2</sup> with the less efficient linear way.

- ✓ The results of the experiment validate the effectiveness of our LIEML algorithm for multi-label learning.

# ④ Conclusion

# Conclusion

---

## Major Contribution

This paper proposes a novel multi-label learning method named LIEMML, which enhances the labeling information by extending logical labels into numerical labels.

The labeling information is enhanced by leveraging the underlying low-rank structure in the stacked matrix.

## More Information

Our lab website:



<http://palm.seu.edu.cn/>

My personal website:



<https://antao.netlify.com/>





**Thank You!**

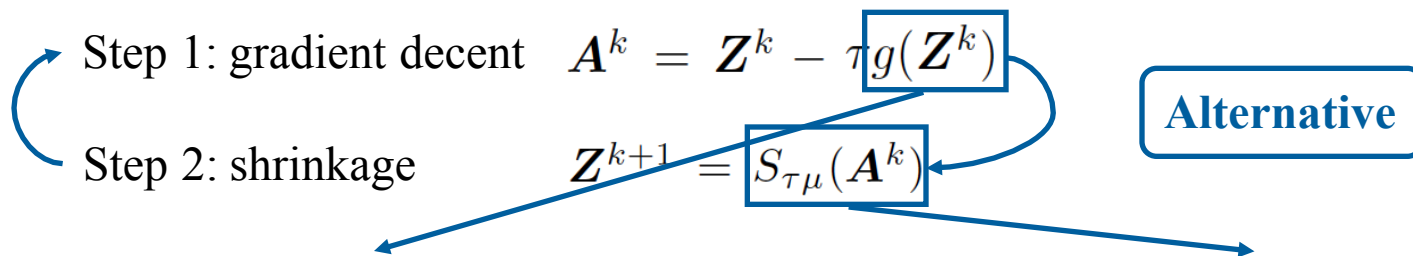


# The LIEML Algorithm

## Label Enhancement

$$T_1(\mathbf{Z}) = \frac{\lambda}{tn} \sum_{i=1}^t \sum_{j=1}^n \log(1 + e^{-y_{ij} z_{ij}}) + \mu \|\mathbf{Z}\|_* + \frac{1}{dn} \sum_{i=1}^d \sum_{j=1}^n \frac{1}{2} (z_{(i+t)j} - x_{ij})^2$$

To optimize the target function  $T_1$  :



$$g(\mathbf{z}_{ij}) = \begin{cases} \frac{\lambda}{tn} \frac{-y_{ij}}{1+e^{y_{ij} z_{ij}}}, & i \leq t \\ \frac{1}{dn} (z_{ij} - x_{(i-t)j}), & t < i \leq t + d \\ 0, & i = t + d + 1 \end{cases}$$

$$\mathbf{A}^k = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^\top$$
$$S_{\tau\mu}(\mathbf{A}^k) = \mathbf{U} \max(\mathbf{\Lambda} - \tau\mu, 0) \mathbf{V}^\top$$

- ✓ To improve the speed of convergence, we begin with a large value  $\mu_1$  for  $\mu$ , and decay  $\mu$  as  $\mu_1 > \mu_2 > \dots > \mu_L = \mu_{min} > 0$ .



# The LIEMML Algorithm

## Predictive Model Induction

To minimize  $T_2(\Theta, \mathbf{m})$ , we use an iterative quasi-Newton method called Iterative Re-Weighted Least Square (IRWLS).

$$\begin{aligned} T_2(\Theta, \mathbf{m}) &= \frac{1}{2} \sum_{j=1}^t \|\theta_j\|^2 + \gamma_1 \sum_{i=1}^n \Omega_1(r_i) + \gamma_2 \sum_{i=1}^n \sum_{j=1}^t \Omega_2(q_{ij}) \\ &\approx T_2''(\Theta, \mathbf{m}) = \frac{1}{2} \sum_{j=1}^t \|\theta_j\|^2 + \frac{1}{2} \gamma_1 \sum_{i=1}^n a_i r_i^2 + \gamma_2 \sum_{i=1}^n \sum_{j=1}^t q_{ij} \sigma(-q_{ij}) + \nu \end{aligned}$$

The quadratic problem can be solved as:

$$\begin{bmatrix} \gamma_1 \Phi^\top D_a \Phi + I & \gamma_1 \Phi^\top \mathbf{a} \\ \gamma_1 \mathbf{a}^\top \Phi & \gamma_1 \mathbf{1}^\top \mathbf{a} \end{bmatrix} \begin{bmatrix} \theta'_j \\ m'_j \end{bmatrix} = \begin{bmatrix} \gamma_1 \Phi^\top D_a \mathbf{u}_j + \gamma_2 \Phi^\top D_j \mathbf{y}_j \\ \gamma_1 \mathbf{a}^\top \mathbf{u}_j + \gamma_2 (\sigma_j)^\top \mathbf{y}_j \end{bmatrix}$$

•  $\Phi = [\varphi(\mathbf{x}_1), \dots, \varphi(\mathbf{x}_n)]^\top$ ,  $\mathbf{a} = [a_1, \dots, a_n]^\top$ ,  $(D_a)_{ik} = a_i \delta_{ik}$ ,  $(D_j)_{ik} = \sigma(-q_{ij}) \delta_{ik}$ ,  $\sigma_j = [\sigma(-q_{1j}), \dots, \sigma(-q_{nj})]^\top$ ,  $\mathbf{y}_j = [y_{1j}, \dots, y_{nj}]^\top$ .

The solution for the next iteration ( $\Theta^{(k+1)}$  and  $\mathbf{m}^{(k+1)}$ ) of  $T_2(\Theta, \mathbf{m})$  is obtained via a line search algorithm along  $(\Theta'$  and  $\mathbf{m}')$ .

# Experiments

---

## Experiment Configuration

Let  $\mathcal{S} = \{(\mathbf{x}_i, Y_i) \mid 1 \leq i \leq p\}$  denote the multi-label test set, and  $h(\cdot)$  (or equivalently  $f(\cdot, \cdot)$ ) denote the learned multi-label predictor. Typical example-based measures include:

- *Subset Accuracy*:  $\frac{1}{p} \sum_{i=1}^p \llbracket h(\mathbf{x}_i) = Y_i \rrbracket$ . This measure evaluates the proportion of test examples whose predicted label set coincides with the ground-truth label set. Here,  $\llbracket \pi \rrbracket$  returns 1 if predicate  $\pi$  holds, and 0 otherwise.
- *Hamming Loss*:  $\frac{1}{p} \sum_{i=1}^p \frac{1}{q} |h(\mathbf{x}_i) \Delta Y_i|$ . This measure evaluates the proportion of misclassified instance-label pairs, i.e., a relevant label is missed or an irrelevant label is predicted. Here,  $\Delta$  stands for the symmetric difference between two sets and  $|\cdot|$  measures the cardinality of a set.
- *One-error*:  $\frac{1}{p} \sum_{i=1}^p \llbracket \arg \max_{y \in \mathcal{Y}} f(\mathbf{x}_i, y) \notin Y_i \rrbracket$ . This measure evaluates the proportion of test examples whose top-1 predicted label fails to be a relevant label.
- *Coverage*:  $\frac{1}{p} \sum_{i=1}^p \max_{y \in Y_i} \text{rank}_f(\mathbf{x}_i, y) - 1$ . This measure evaluates the number of steps needed to move down the ranked label list so as to cover all relevant labels of the test example. Here,  $\text{rank}_f(\mathbf{x}, y)$  returns the rank of class label  $y$  within label space  $\mathcal{Y}$  according to the descending order specified by  $f(\mathbf{x}, \cdot)$ .

# Experiments

---

## Experiment Configuration

- *Ranking Loss*:  $\frac{1}{p} \sum_{i=1}^p \frac{1}{|Y_i||\bar{Y}_i|} |\{(y', y'') | f(\mathbf{x}, y') \leq f(\mathbf{x}_i, y''), (y', y'') \in Y_i \times \bar{Y}_i\}|$ . This measure evaluates the proportion of incorrectly ordered label pairs, i.e., an irrelevant label yields larger output value than a relevant label. Here,  $\bar{Y}_i$  is the complementary set of  $Y_i$  in  $\mathcal{Y}$ .
- *Average Precision*:  $\frac{1}{p} \sum_{i=1}^p \frac{1}{|Y_i|} \sum_{y \in Y_i} \frac{|\{y' | \text{rank}_f(\mathbf{x}_i, y') \leq \text{rank}_f(\mathbf{x}_i, y), y' \in Y_i\}|}{\text{rank}_f(\mathbf{x}_i, y)}$ . This measure evaluates the average proportion of labels ranked higher than a relevant label  $y \in Y_i$  that are also relevant.

For *hamming loss*, *one-error*, *coverage* and *ranking loss*, the smaller the value, the better the generalization performance. For the other example-based measures, the larger the value, the better the performance.

# Introduction

---

**In traditional multi-label learning:**

