Cost-Sensitive Label Embedding for Multi-Label Classification

Kuan-Hao Huang and Hsuan-Tien Lin

Department of Computer Science & Information Engineering
National Taiwan University

ECML PKDD, September 20, 2017

Multi-Label Classification (MLC)

Multi-Label Classification

- an extension of the multi-class classification
- allow instance with multiple associated classes

Multi-Label Classification (MLC)

Multi-Label Classification

- an extension of the multi-class classification
- allow instance with multiple associated classes

Example: Image with Animals (dog, cat, rabbit, shark)

image				
class	{ dog, cat }	{ dog, cat, rabbit }	{ dog }	{ shark }
label	$(1, 1, 0, 0)$	$(1, 1, 1, 0)$	$(1, 0, 0, 0)$	$(0, 0, 0, 1)$

Multi-Label Classification (MLC)

Notation

- feature vector (image): $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$
- label vector (classes): $\mathbf{y} \in \mathcal{Y} \subseteq \{0, 1\}^K$

Multi-Label Classification (MLC)

Notation

- feature vector (image): $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$
- label vector (classes): $\mathbf{y} \in \mathcal{Y} \subseteq \{0, 1\}^K$

Multi-Label Classification

- given training instances $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$
- learn a **predictor** h from $\mathcal{D}$
- for testing instance $(\mathbf{x}, \mathbf{y})$, prediction $\tilde{\mathbf{y}} = h(\mathbf{x})$
- let the prediction $\tilde{\mathbf{y}}$ be close to ground truth $\mathbf{y}$

Multi-Label Classification (MLC)

Notation

- feature vector (image): $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$
- label vector (classes): $\mathbf{y} \in \mathcal{Y} \subseteq \{0, 1\}^K$

Multi-Label Classification

- given training instances $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$
- learn a **predictor** h from $\mathcal{D}$
- for testing instance $(\mathbf{x}, \mathbf{y})$, prediction $\tilde{\mathbf{y}} = h(\mathbf{x})$
- let the prediction $\tilde{\mathbf{y}}$ be close to ground truth $\mathbf{y}$

Evaluation of Closeness

- cost function $c(\mathbf{y}, \tilde{\mathbf{y}})$: the penalty of predicting $\mathbf{y}$ as $\tilde{\mathbf{y}}$
- Hamming loss, 0/1 loss, Rank loss, F1 score(loss), Accuracy score(loss)

Cost-Sensitive Multi-Label Classification (CSMLC)

Notation

- feature vector (image): $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$
- label vector (classes): $\mathbf{y} \in \mathcal{Y} \subseteq \{0, 1\}^K$

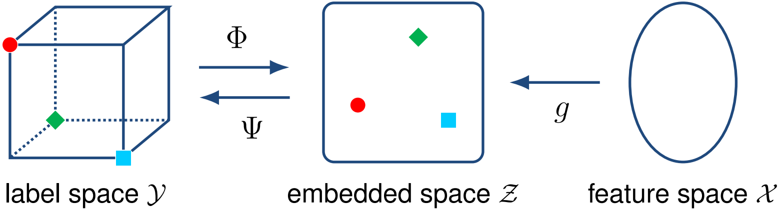
Cost-Sensitive Multi-Label Classification (CSMLC)

- given training instances $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$ and cost function c
- learn a **predictor** h from both $\mathcal{D}$ and c
- for testing instance $(\mathbf{x}, \mathbf{y})$, prediction $\tilde{\mathbf{y}} = h(\mathbf{x})$
- let the prediction $\tilde{\mathbf{y}}$ be close to ground truth $\mathbf{y}$

Evaluation of Closeness

- cost function $c(\mathbf{y}, \tilde{\mathbf{y}})$: the penalty of predicting $\mathbf{y}$ as $\tilde{\mathbf{y}}$
- Hamming loss, 0/1 loss, Rank loss, F1 score(loss), Accuracy score(loss)

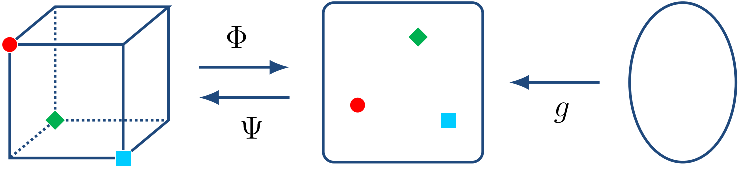
Label Embedding



Training Stage

- embedding function Φ: label vector $\mathbf{y} \rightarrow$ embedded vector $\mathbf{z}$
- learn a regressor g from $\{(\mathbf{x}^{(n)}, \mathbf{z}^{(n)})\}_{n=1}^{N}$

Label Embedding



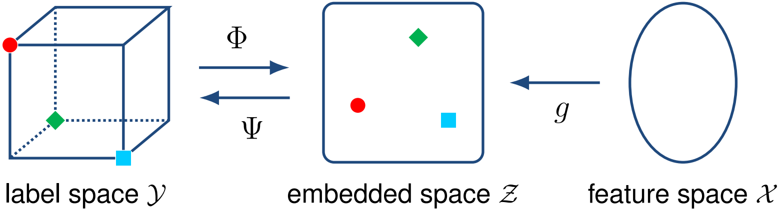
Training Stage

- embedding function Φ: label vector $\mathbf{y} \to$ embedded vector $\mathbf{z}$
- learn a regressor g from $\{(\mathbf{x}^{(n)}, \mathbf{z}^{(n)})\}_{n=1}^{N}$

Predicting Stage

- for testing instance $\mathbf{x}$, predicted embedded vector $\tilde{\mathbf{z}} = g(\mathbf{x})$
- decoding function Ψ: predicted embedded vector $\tilde{\mathbf{z}} \to$ predicted label vector $\tilde{\mathbf{y}}$

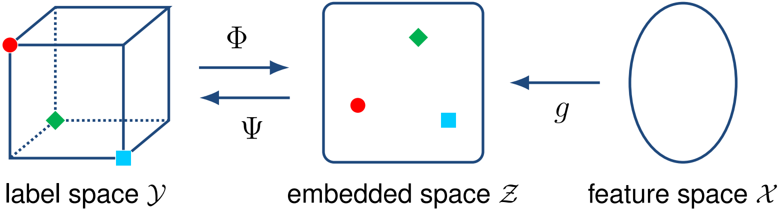
Cost-Sensitive Label Embedding



Existing Works

- **label embedding**: PLST, FaIE, RAkEL, ECC-based [Tai et al., 2012; Lin et al., 2014; Tsoumakas et al., 2011; Ferng et al., 2013]
- **cost-sensitivity**: CFT, PCC [Li et al., 2014; Dembczynski et al., 2010]
- **cost-sensitivity + label embedding**: no existing works

Cost-Sensitive Label Embedding



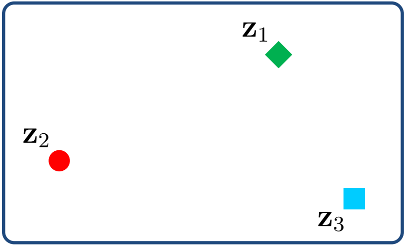
Existing Works

- **label embedding**: PLST, FaIE, RAkEL, ECC-based [Tai et al., 2012; Lin et al., 2014; Tsoumakas et al., 2011; Ferng et al., 2013]
- **cost-sensitivity**: CFT, PCC [Li et al., 2014; Dembczynski et al., 2010]
- **cost-sensitivity + label embedding**: no existing works

Cost-Sensitive Label Embedding

- consider cost function c when designing embedding function Φ and decoding function Ψ (cost-sensitive embedded vectors $\mathbf{z}$)

Cost-Sensitive Embedding

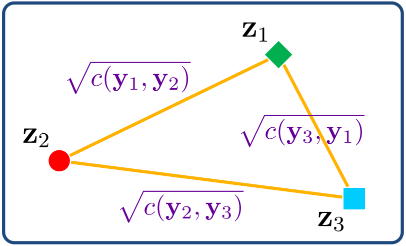


embedded space $\mathcal{Z}$

Training Stage

- distances between embedded vectors $\Leftrightarrow$ cost information
- larger (smaller) distance $d(\mathbf{z}_i, \mathbf{z}_j) \Leftrightarrow$ higher (lower) cost $c(\mathbf{y}_i, \mathbf{y}_j)$

Cost-Sensitive Embedding

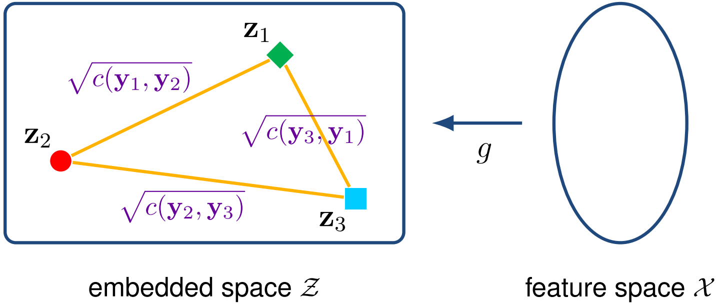


embedded space $\mathcal{Z}$

Training Stage

- distances between embedded vectors $\Leftrightarrow$ cost information
- larger (smaller) distance $d(\mathbf{z}_i, \mathbf{z}_j) \Leftrightarrow$ higher (lower) cost $c(\mathbf{y}_i, \mathbf{y}_j)$
- $d(\mathbf{z}_i, \mathbf{z}_j) \approx \sqrt{c(\mathbf{y}_i, \mathbf{y}_j)}$ by multidimensional scaling (manifold learning)

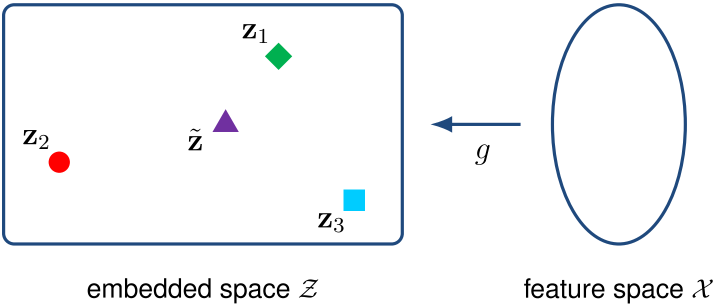
Cost-Sensitive Embedding



Training Stage

- distances between embedded vectors $\Leftrightarrow$ cost information
- larger (smaller) distance $d(\mathbf{z}_i, \mathbf{z}_j) \Leftrightarrow$ higher (lower) cost $c(\mathbf{y}_i, \mathbf{y}_j)$
- $d(\mathbf{z}_i, \mathbf{z}_j) \approx \sqrt{c(\mathbf{y}_i, \mathbf{y}_j)}$ by multidimensional scaling (manifold learning)

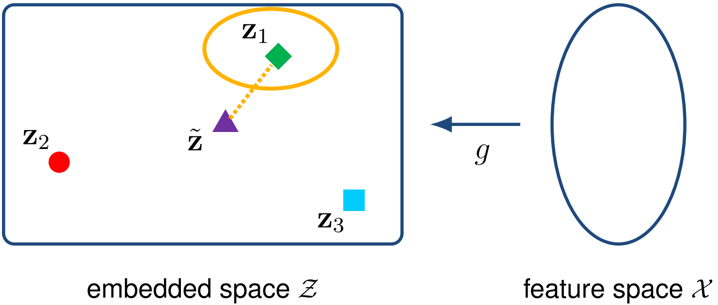
Cost-Sensitive Decoding



Predicting Stage

- for testing instance $\mathbf{x}$, predicted embedded vector $\tilde{\mathbf{z}} = g(\mathbf{x})$

Cost-Sensitive Decoding



Predicting Stage

- for testing instance $\mathbf{x}$, predicted embedded vector $\tilde{\mathbf{z}} = g(\mathbf{x})$
- find nearest embedded vector $\mathbf{z}_q$ of $\tilde{\mathbf{z}}$
- cost-sensitive prediction $\tilde{\mathbf{y}} = \mathbf{y}_q$

Theoretical Explanation

Theorem

$$c(\mathbf{y}, \tilde{\mathbf{y}}) \leq 5\left(\underbrace{\left(d(\mathbf{z}, \mathbf{z}_q) - \sqrt{c(\mathbf{y}, \mathbf{y}_q)}\right)^2}_{\text{embedding error}} + \underbrace{\|\mathbf{z} - g(\mathbf{x})\|^2}_{\text{regression error}}\right)$$

Theoretical Explanation

Theorem

$$c(\mathbf{y}, \tilde{\mathbf{y}}) \leq 5\left(\underbrace{\left(d(\mathbf{z}, \mathbf{z}_q) - \sqrt{c(\mathbf{y}, \mathbf{y}_q)}\right)^2}_{\text{embedding error}} + \underbrace{\|\mathbf{z} - g(\mathbf{x})\|^2}_{\text{regression error}}\right)$$

Optimization

- embedding error $\rightarrow$ multidimensional scaling
- regression error $\rightarrow$ regressor g

Theoretical Explanation

Theorem

$$c(\mathbf{y}, \tilde{\mathbf{y}}) \leq 5\left(\underbrace{\left(d(\mathbf{z}, \mathbf{z}_q) - \sqrt{c(\mathbf{y}, \mathbf{y}_q)}\right)^2}_{\text{embedding error}} + \underbrace{\|\mathbf{z} - g(\mathbf{x})\|^2}_{\text{regression error}}\right)$$

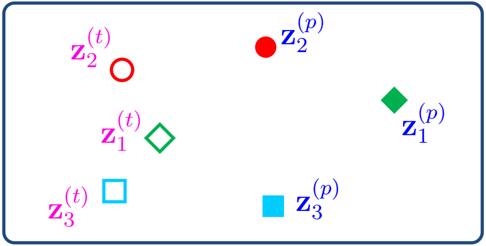
Optimization

- embedding error $\rightarrow$ multidimensional scaling
- regression error $\rightarrow$ regressor g

Challenge

- **asymmetric cost function** vs. **symmetric distance**?
- $c(\mathbf{y}_i, \mathbf{y}_j) \neq c(\mathbf{y}_j, \mathbf{y}_i)$ vs. $d(\mathbf{z}_i, \mathbf{z}_j)$

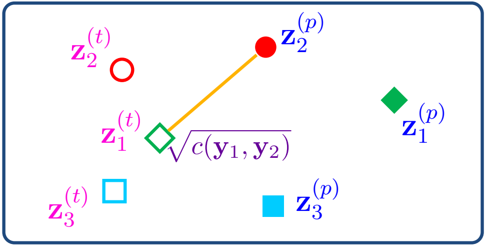
Mirroring Trick



embedded space $\mathcal{Z}$

- two roles of $\mathbf{y}_i$: **ground truth role $\mathbf{y}_i^{(t)}$** and **prediction role $\mathbf{y}_i^{(p)}$**

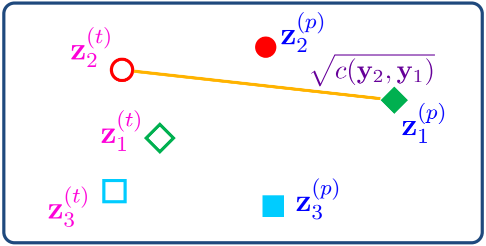
Mirroring Trick



embedded space $\mathcal{Z}$

- two roles of $\mathbf{y}_i$: **ground truth role** $\mathbf{y}_i^{(t)}$ and **prediction role** $\mathbf{y}_i^{(p)}$
- $\sqrt{c(\mathbf{y}_i, \mathbf{y}_j)} \Rightarrow$ predict $\mathbf{y}_i$ as $\mathbf{y}_j \Rightarrow$ for $\mathbf{z}_i^{(t)}$ and $\mathbf{z}_j^{(p)}$

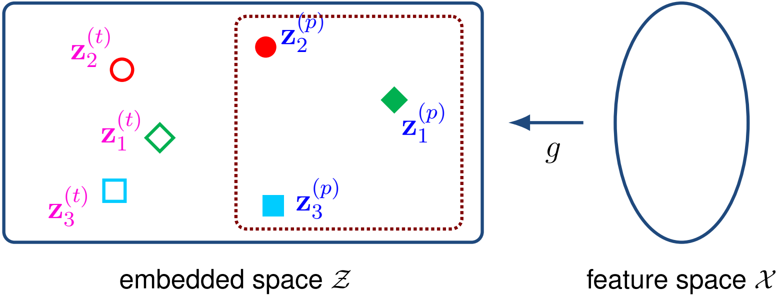
Mirroring Trick



embedded space $\mathcal{Z}$

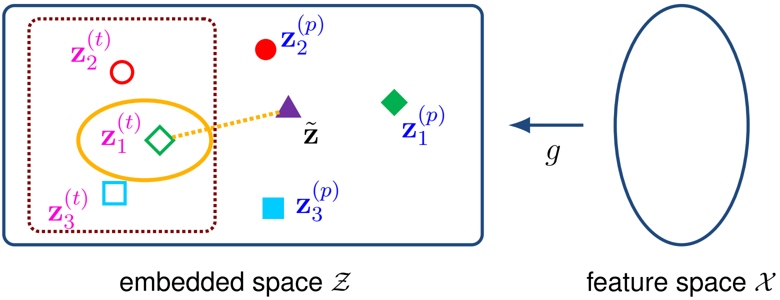
- two roles of $\mathbf{y}_i$: **ground truth role** $\mathbf{y}_i^{(t)}$ and **prediction role** $\mathbf{y}_i^{(p)}$
- $\sqrt{c(\mathbf{y}_i, \mathbf{y}_j)} \Rightarrow$ predict $\mathbf{y}_i$ as $\mathbf{y}_j \Rightarrow$ for $\mathbf{z}_i^{(t)}$ and $\mathbf{z}_j^{(p)}$
- $\sqrt{c(\mathbf{y}_j, \mathbf{y}_i)} \Rightarrow$ predict $\mathbf{y}_j$ as $\mathbf{y}_i \Rightarrow$ for $\mathbf{z}_i^{(p)}$ and $\mathbf{z}_j^{(t)}$

Mirroring Trick



- two roles of $\mathbf{y}_i$: **ground truth role** $\mathbf{y}_i^{(t)}$ and **prediction role** $\mathbf{y}_i^{(p)}$
- $\sqrt{c(\mathbf{y}_i, \mathbf{y}_j)} \Rightarrow$ predict $\mathbf{y}_i$ as $\mathbf{y}_j \Rightarrow$ for $\mathbf{z}_i^{(t)}$ and $\mathbf{z}_j^{(p)}$
- $\sqrt{c(\mathbf{y}_j, \mathbf{y}_i)} \Rightarrow$ predict $\mathbf{y}_j$ as $\mathbf{y}_i \Rightarrow$ for $\mathbf{z}_i^{(p)}$ and $\mathbf{z}_j^{(t)}$
- learn **regressor** g from $\mathbf{z}_i^{(p)}, \mathbf{z}_2^{(p)}, \ldots, \mathbf{z}_L^{(p)}$

Mirroring Trick



- two roles of $\mathbf{y}_i$: **ground truth role** $\mathbf{y}_i^{(t)}$ and **prediction role** $\mathbf{y}_i^{(p)}$
- $\sqrt{c(\mathbf{y}_i, \mathbf{y}_j)} \Rightarrow$ predict $\mathbf{y}_i$ as $\mathbf{y}_j \Rightarrow$ for $\mathbf{z}_i^{(t)}$ and $\mathbf{z}_j^{(p)}$
- $\sqrt{c(\mathbf{y}_j, \mathbf{y}_i)} \Rightarrow$ predict $\mathbf{y}_j$ as $\mathbf{y}_i \Rightarrow$ for $\mathbf{z}_i^{(p)}$ and $\mathbf{z}_j^{(t)}$
- learn **regressor** g from $\mathbf{z}_i^{(p)}, \mathbf{z}_2^{(p)}, \ldots, \mathbf{z}_L^{(p)}$
- find **nearest embedded vector** of $\tilde{\mathbf{z}}$ from $\mathbf{z}_1^{(t)}, \mathbf{z}_2^{(t)}, \ldots, \mathbf{z}_L^{(t)}$

Cost-Sensitive Label Embedding with Multidimensional Scaling

Training Stage of **CLEMS**

- given training instances $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$ and cost function c
- determine two roles of embedded vectors $\mathbf{z}_i^{(t)}$ and $\mathbf{z}_i^{(p)}$ for label vector $\mathbf{y}_i$
- embedding function $\Phi\colon \mathbf{y}_i \to \mathbf{z}_i^{(p)}$
- learn a regressor g from $\{(\mathbf{x}^{(n)}, \Phi(\mathbf{y}^{(n)}))\}_{n=1}^{N}$

Cost-Sensitive Label Embedding with Multidimensional Scaling

Training Stage of **CLEMS**

- given training instances $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$ and cost function c
- determine two roles of embedded vectors $\mathbf{z}_i^{(t)}$ and $\mathbf{z}_i^{(p)}$ for label vector $\mathbf{y}_i$
- embedding function $\Phi\colon \mathbf{y}_i \to \mathbf{z}_i^{(p)}$
- learn a regressor g from $\{(\mathbf{x}^{(n)}, \Phi(\mathbf{y}^{(n)}))\}_{n=1}^{N}$

Predicting Stage of **CLEMS**

- given the testing instance $\mathbf{x}$
- obtain the predicted embedded vector by $\tilde{\mathbf{z}} = g(\mathbf{x})$
- decoding $\Psi(\cdot) = \Phi^{-1}$(nearest neighbor) $= \Phi^{-1}\big(\operatorname{argmin} d(\mathbf{z}_i^{(t)}, \cdot)\big)$
- prediction $\tilde{\mathbf{y}} = \Psi(\tilde{\mathbf{z}})$

Experiments

Settings

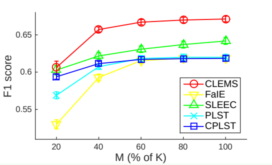
- 12 public datasets
- 50% for training, 25% for validation, and 25% for testing
- tune parameters by validation
- evaluation criteria
  - **F1 score** $\frac{2\|\mathbf{y}\cap\tilde{\mathbf{y}}\|_1}{\|\mathbf{y}\|_1+\|\tilde{\mathbf{y}}\|_1}$ $(\uparrow)$
  - **Accuracy score** $\frac{\|\mathbf{y}\cap\tilde{\mathbf{y}}\|_1}{\|\mathbf{y}\cup\tilde{\mathbf{y}}\|_1}$ $(\uparrow)$
  - **Rank loss** $\sum_{\mathbf{y}[i]>\mathbf{y}[j]}(\llbracket\tilde{\mathbf{y}}[i]<\tilde{\mathbf{y}}[j]\rrbracket+\frac{1}{2}\llbracket\tilde{\mathbf{y}}[i]=\tilde{\mathbf{y}}[j]\rrbracket)$ $(\downarrow)$
- average results of 20 experiments

Competitors

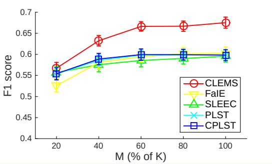
- label embedding algorithms
- cost-sensitive algorithms

Comparison with Label Embedding Algorithms

F1 score (↑)

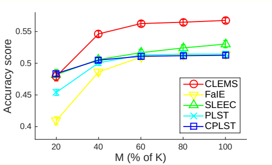


yeast

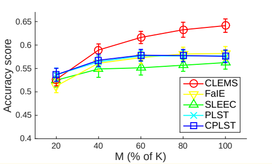


birds

Accuracy score (↑)

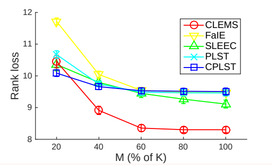


yeast

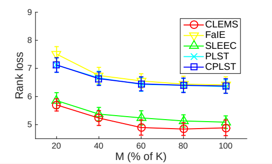


birds

Rank loss (↓)



yeast



birds

CLEMS is the best across different criteria and dimensions

Comparison with Cost-Sensitive Algorithms

Table: Performance across different evaluation criteria

data	F1 score (↑)			Accuracy score (↑)			Rank loss (↓)		
	CLEMS	CFT	PCC	CLEMS	CFT	PCC	CLEMS	CFT	PCC
emot.	**0.676**	0.640	0.643	**0.589**	0.557	–	1.484	1.563	**1.467**
scene	**0.770**	0.703	0.745	**0.760**	0.656	–	0.672	0.723	**0.645**
yeast	**0.671**	0.649	0.614	**0.568**	0.543	–	**8.302**	8.566	8.469
birds	**0.677**	0.601	0.636	**0.642**	0.586	–	4.886	4.908	**3.660**
med.	**0.814**	0.635	0.573	**0.786**	0.613	–	5.170	5.811	**4.234**
enron	**0.606**	0.557	0.542	**0.491**	0.448	–	29.40	26.64	**25.11**
lang.	**0.375**	0.168	0.247	**0.327**	0.164	–	31.03	34.16	**19.11**
flag	**0.731**	0.692	0.706	**0.615**	0.588	–	2.930	3.075	**2.857**
slash	**0.568**	0.429	0.503	**0.538**	0.402	–	4.986	5.677	**4.472**
CAL.	**0.419**	0.371	0.391	**0.273**	0.237	–	1247	1120	**993**
arts	**0.492**	0.334	0.349	**0.451**	0.281	–	9.865	10.07	**8.467**
EUR.	**0.670**	0.456	0.483	**0.650**	0.450	–	89.52	129.5	**43.28**

- **generality for CSMLC**: CLEMS $=$ CFT $>$ PCC
  - PCC requires an efficient inference rule
- **performance**: CLEMS $\approx$ PCC $>$ CFT
- **speed**: CLEMS $\approx$ PCC $>$ CFT

Conclusion

- **algorithm design:** cost-sensitive label embedding algorithm (CLEMS)
  - embed the cost information in distance by multidimensional scaling
  - nearest-neighbor based decoding function
  - mirroring trick for asymmetric cost functions
- **theoretical explanation:**
  - prove the upper bound of the predicted cost for CLEMS
- **empirical performance:**
  - CLEMS outperforms existing label embedding algorithms
  - CLEMS is better than state-of-the-art cost-sensitive algorithms

Conclusion

- **algorithm design:** cost-sensitive label embedding algorithm (CLEMS)
  - embed the cost information in distance by multidimensional scaling
  - nearest-neighbor based decoding function
  - mirroring trick for asymmetric cost functions
- **theoretical explanation:**
  - prove the upper bound of the predicted cost for CLEMS
- **empirical performance:**
  - CLEMS outperforms existing label embedding algorithms
  - CLEMS is better than state-of-the-art cost-sensitive algorithms

Thank you! Any question?