Cost-Sensitive Reference Pair Encoding for Multi-Label Learning

Yao-Yuan Yang and Kuan-Hao Huang and Chih-Wei Chang and Hsuan-Tien Lin

Department of Computer Science & Information Engineering
National Taiwan University

June 4, 2018

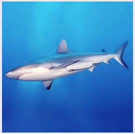
Which animal?

dog

rabbit

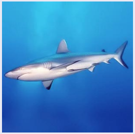
cat

guinea pig



shark

Which animals?

dog	rabbit	cat	guinea pig	shark
1	1	1	1	0

Multi-label Classification (MLC)

Problem definition

- train set $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$
- label vector $\mathbf{y} \in \{0, 1\}^K$ as a binary vector
  - $\mathbf{y}[k] = 1$ if and only if the k-th bit is relevant
- learn a f from $\mathcal{D}$ that maps $\mathbf{x}$ to $\mathbf{y}$
- test data $(\mathbf{x}, \mathbf{y})$, prediction $\hat{\mathbf{y}} = f(\mathbf{x})$
- goal is to make $\hat{\mathbf{y}}$ close to ground truth $\mathbf{y}$

Evaluation

- cost function $C(\mathbf{y}, \hat{\mathbf{y}})$: the cost of predicting $\mathbf{y}$ as $\hat{\mathbf{y}}$
- F1 score, Accuracy score, Hamming loss, Rank loss

Cost-Sensitive Multi-label Classification (CSMLC)

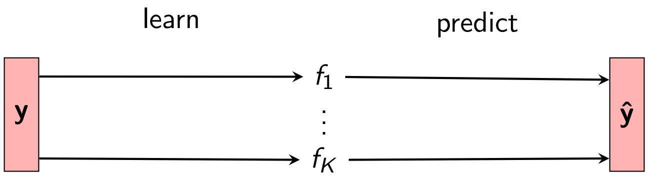
Problem definition

- train set $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$
- label vector $\mathbf{y} \in \{0,1\}^K$ as a binary vector
  - $\mathbf{y}[k] = 1$ if and only if the k-th bit is relevant
- learn a f from $\mathcal{D}$ and cost function C that maps $\mathbf{x}$ to $\mathbf{y}$
- test data $(\mathbf{x}, \mathbf{y})$, prediction $\hat{\mathbf{y}} = f(\mathbf{x})$
- goal is to make $\hat{\mathbf{y}}$ close to ground truth $\mathbf{y}$

Evaluation

- cost function $C(\mathbf{y}, \hat{\mathbf{y}})$: the cost of predicting $\mathbf{y}$ as $\hat{\mathbf{y}}$
- F1 score, Accuracy score, Hamming loss, Rank loss

Naive approach



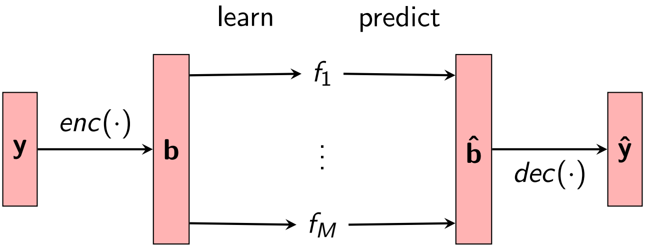
Binary Relevance (BR)

- train independent binary classifier for $\mathbf{y}_1, \ldots, \mathbf{y}_K$

Label space encoding

- add bits to label vector $\mathbf{y}$, prediction error can be "corrected"
- $enc(\cdot) : \{0,1\}^K \to \{0,1\}^M$
- $dec(\cdot) : \{0,1\}^M \to \{0,1\}^K$
- exists no cost-sensitive code

Naive approach



Binary Relevance (BR)

- train independent binary classifier for $\mathbf{y}_1, \ldots, \mathbf{y}_K$

Label space encoding

- add bits to label vector $\mathbf{y}$, prediction error can be "corrected"
- $enc(\cdot) : \{0, 1\}^K \to \{0, 1\}^M$
- $dec(\cdot) : \{0, 1\}^M \to \{0, 1\}^K$
- exists no cost-sensitive code

Cost-sensitive encoding

One versus One (OVO) encoding

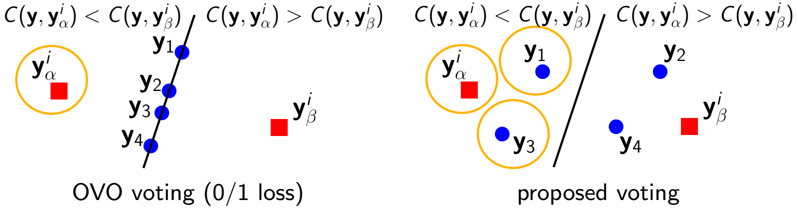
- multi-class classification OVO reduction
- consider each possible label vector $\mathbf{y}$ as an independent class
- $\mathbf{y}^i_\alpha, \mathbf{y}^i_\beta \in \{0,1\}^K$ as the reference label vector

$$enc_{ovo}(\mathbf{y})[i] = \begin{cases} 1 & \text{if } \mathbf{y} = \mathbf{y}^i_\alpha \\ 0 & \text{if } \mathbf{y} = \mathbf{y}^i_\beta \\ 0.5 & \text{otherwise} \end{cases}$$

Cost-Sensitive encoding

- multi-class classification for accuracy $\Rightarrow$ to 0/1 loss in MLC

$$enc_{cs}(\mathbf{y})[i] = \begin{cases} 1 & \text{if } C(\mathbf{y}, \mathbf{y}^i_\alpha) < C(\mathbf{y}, \mathbf{y}^i_\beta) \\ 0 & \text{if } C(\mathbf{y}, \mathbf{y}^i_\alpha) > C(\mathbf{y}, \mathbf{y}^i_\beta) \\ 0.5 & \text{otherwise}(C(\mathbf{y}, \mathbf{y}^i_\alpha) = C(\mathbf{y}, \mathbf{y}^i_\beta)) \end{cases}$$



- equivalent to finding the nearest neighbor under Hamming distance in the encoding space
- encode cost information into distance between label vectors

$$dec_{cs}(\hat{\mathbf{b}}) = \operatorname*{argmin}_{\mathbf{y}\in\{0,1\}^K} d_{ham}(\hat{\mathbf{b}}, enc_{cs}(\mathbf{y}))$$

Speedup

Sampling code

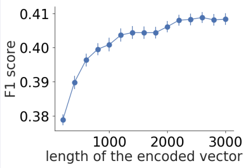
- the full code length is as large as $\binom{2^K}{2}$
- redundancy
  - consider the two bits (i and j)
    - $\mathbf{y}_\alpha^i = (1, 1, 1, 0)$, $\mathbf{y}_\beta^i = (1, 1, 0, 1)$
    - $\mathbf{y}_\alpha^j = (1, 0, 1, 0)$, $\mathbf{y}_\beta^j = (1, 0, 0, 1)$
  - learning similar things (last two labels are $(1, 0)$ or $(0, 1)$?)
- uniform sampling works well

Candidate set

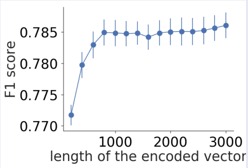
- infeasible to search the full label space $\{0, 1\}^K$
- search only in a subset (candidate set) of the label vectors
- reasonable choice is all distinct label vectors in training set

Convergence

F1 score (↑)

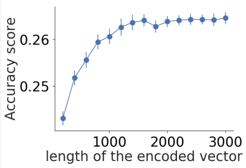


CAL500

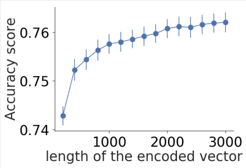


scene

Accuracy score (↑)

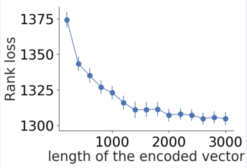


CAL500

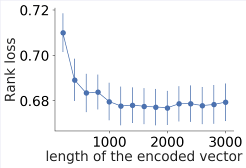


scene

Rank loss (↓)



CAL500



scene

CSRPE converges steadily with the increase of code length

Compare with other encoding algorithms

Data set	F1 score ↑				Accuracy score ↑			
	REP	RREP	HAMR	CSRPE	REP	RREP	HAMR	CSRPE
Corel5k	.0683	.1028	.0608	**.2455**	.0471	.0696	.0408	**.1664**
CAL500	.3388	.3527	.3152	**.4083**	.2097	.2179	.1925	**.2645**
bibtex	.3636	.3761	.3658	**.4663**	.3063	.3103	.3094	**.3926**
enron	.5441	.5336	.5459	**.5911**	.4303	.4215	.4344	**.4772**
medical	.7883	.7757	.7877	**.8203**	.7559	.7431	.7604	**.7939**
genbase	**.9897**	.9893	.9896	.9878	**.9859**	.9852	.9856	.9835
yeast	.6119	.6130	.6171	**.6670**	.5047	.5065	.5120	**.5653**
flags	.6954	.6965	.7005	**.7222**	.5849	.5860	.5913	**.6056**
scene	.5895	.5926	.6365	**.7860**	.5791	.5816	.6258	**.7620**
emotions	.5968	.5773	.6100	**.6655**	.5179	.4959	.5320	**.5775**
Data set	**Rank loss ↓**				**Hamming loss ↓**			
	REP	RREP	HAMR	CSRPE	REP	RREP	HAMR	CSRPE
Corel5k	618.1	597.2	623.5	**490.2**	.0095	.0097	**.0094**	.0108
CAL500	1500.	1477.	1537.	**1305.**	.1522	**.1416**	.1490	.1651
bibtex	132.6	124.1	131.5	**104.9**	**.0124**	.0130	**.0124**	.0134
enron	43.39	44.06	43.40	**34.32**	.0489	.0499	**.0485**	.0500
medical	5.454	5.733	5.601	**5.330**	.0104	.0107	.0102	**.0100**
genbase	.2461	**.2422**	.2525	.3863	.0012	**.0011**	**.0011**	.0014
yeast	9.609	9.565	9.443	**8.451**	.1941	.1933	.1932	**.1891**
flags	3.123	3.139	3.078	**3.010**	.2591	.2591	.2599	**.2585**
scene	1.136	1.149	1.031	**0.679**	.0914	.0970	.0848	**.0821**
emotions	1.789	1.906	1.764	**1.591**	.1966	.2110	**.1953**	.1994

- under Hamming loss, algorithms perform competitively
- CSRPE is able to generalize better across cost functions

Cost-Sensitive Multi-label Active learning (CSMLAL)

Active learning setting

- labeled pool $\mathcal{D}_l = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N_l}$
- unlabeled pool $\mathcal{D}_u = \{\mathbf{x}^{(n)}\}_{n=1}^{N_u}$
- MLC classifier f_t trained on $\mathcal{D}_l$
- cost function C

CSMLAL

- for iterations $t = 1, \ldots, T$
  1. consider $\mathcal{D}_u$, $\mathcal{D}_l$, f_t, C, query $\mathbf{x}_t \in \mathcal{D}_u$ with label vector $\mathbf{y}_t$
  2. $\mathcal{D}_u = \mathcal{D}_u - \{\mathbf{x}_t\}$
  3. $\mathcal{D}_l = \mathcal{D}_l + \{(\mathbf{x}_t, \mathbf{y}_t)\}$
  4. train f_{t+1} on $\mathcal{D}_l$
- the goal is to minimize the average cost of f_t on the testing instances evaluated on C
- let classifier perform better with less data labeled

CSRPE for CSMLAL

Uncertainty sampling

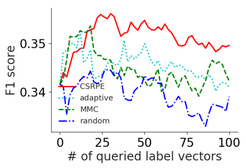
- calculating a value for each instance in $\mathcal{D}_u$
- greedily choose the most uncertain instance
- various way of evaluating the uncertainty

CSRPE

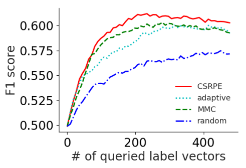
- encoded vector of the prediction: $\bar{\mathbf{b}} = enc_{cs}(\mathbf{f_t}(\mathbf{x}))$
- predicted encoded vector from CSRPE: $\hat{\mathbf{b}} = h(\mathbf{x})$
- nearest encoded vector of $\hat{b}$: $\tilde{\mathbf{b}} = enc_{cs}(dec_{cs}(\hat{\mathbf{b}}))$
- Cost estimation uncertainty
  - $d_{ham}(\hat{\mathbf{b}}, \tilde{\mathbf{b}})$
  - how well CSRPE estimates the cost between encoded vectors
- Cost utility uncertainty
  - $d_{ham}(\hat{\mathbf{b}}, \bar{\mathbf{b}})$
  - how uncertain f_t is under the current cost function

Compare with active learning algorithms

F1 score (↑)

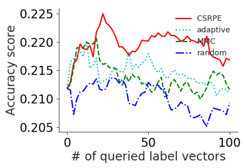


CAL500

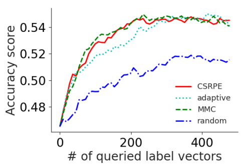


scene

Accuracy score (↑)

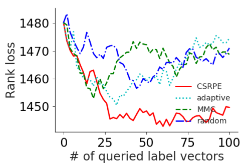


CAL500

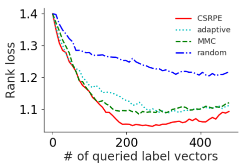


scene

Rank loss (↓)



CAL500



scene

CSRPE performs the best across different criteria

Conclusion

Cost-sensitive code

- derived cost-sensitive encoding from OVO code
- captures cost information in distance of encoded vectors

Multi-lable classification

- exploit the redundancy between classifiers by uniform sampling
- nearest-neighbor-based decoding on a candidate set
- generalize better across different cost functions

Active learning

- the encoding provides better estimation of uncertainty
- generalize better than other active learning algorithms

Thank you for listening. Any question?