Multi-label Learning via Codewords

2018 ICTAI, Volos, Greece, November 5–7, 2018.

Mahlagha Sedghi [1] Yinjie Huang [1] Michael Georgiopoulos [1]
Georgios C. Anagnostopoulos [2]

[1]Machine Learning Laboratory, University of Central Florida, US

[2]ICE Laboratory, Florida Institute of Technology, US

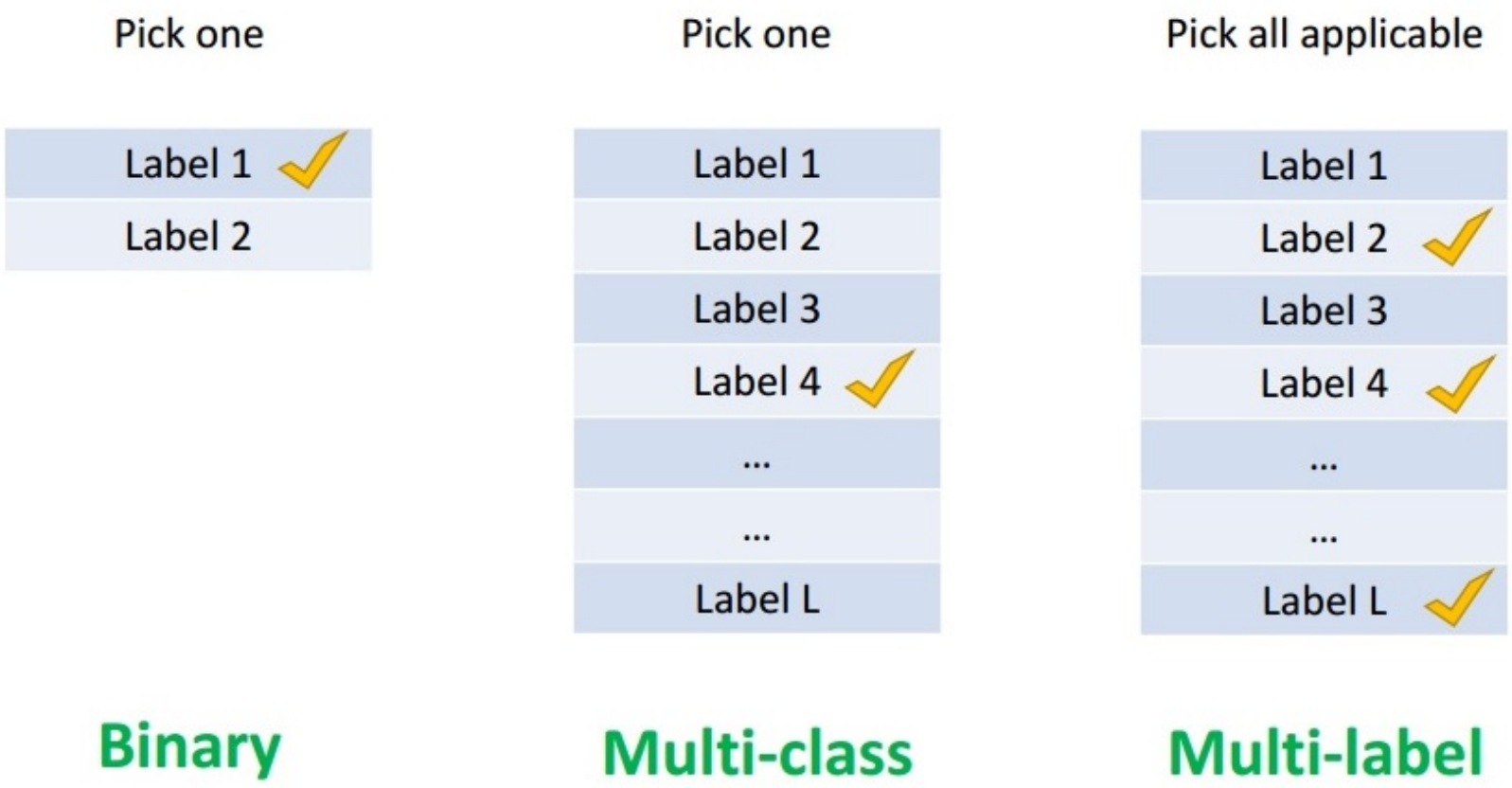
Novemeber 05th, 2018

Table of Contents

Section 1

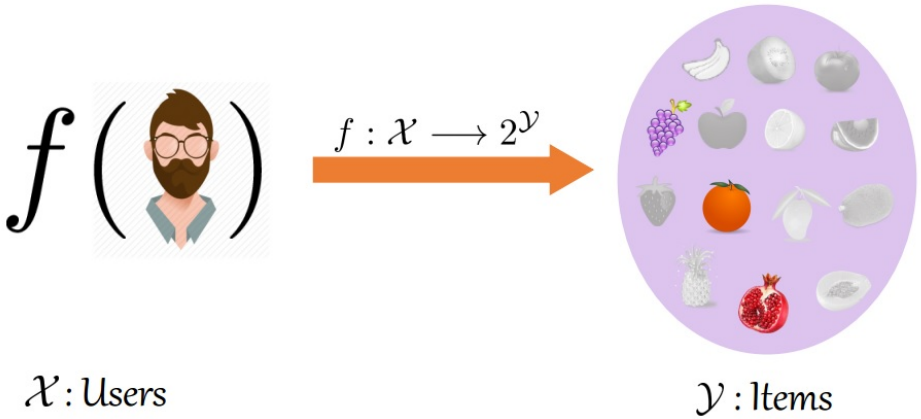
Introduction

Classification Paradigms



Multi-label Classification Examples

What items would this user buy?



Who are present in this selfie?

What tags are related to this image?

Label Correlations

- In most realistic settings, presence or absence of the labels are correlated.
  - In image annotation, if an image features a *fish* tag, it will most likely also feature an *ocean* tag, rather than a *sky* tag.
- Structured Output Prediction (SOP) approaches can be used to capture these correlations.
  - Exact SOP training algorithms applicable to Multi-label problems suffer from computationally intensive training time. ($\mathcal{O}(2^L)$, with L as number of labels)

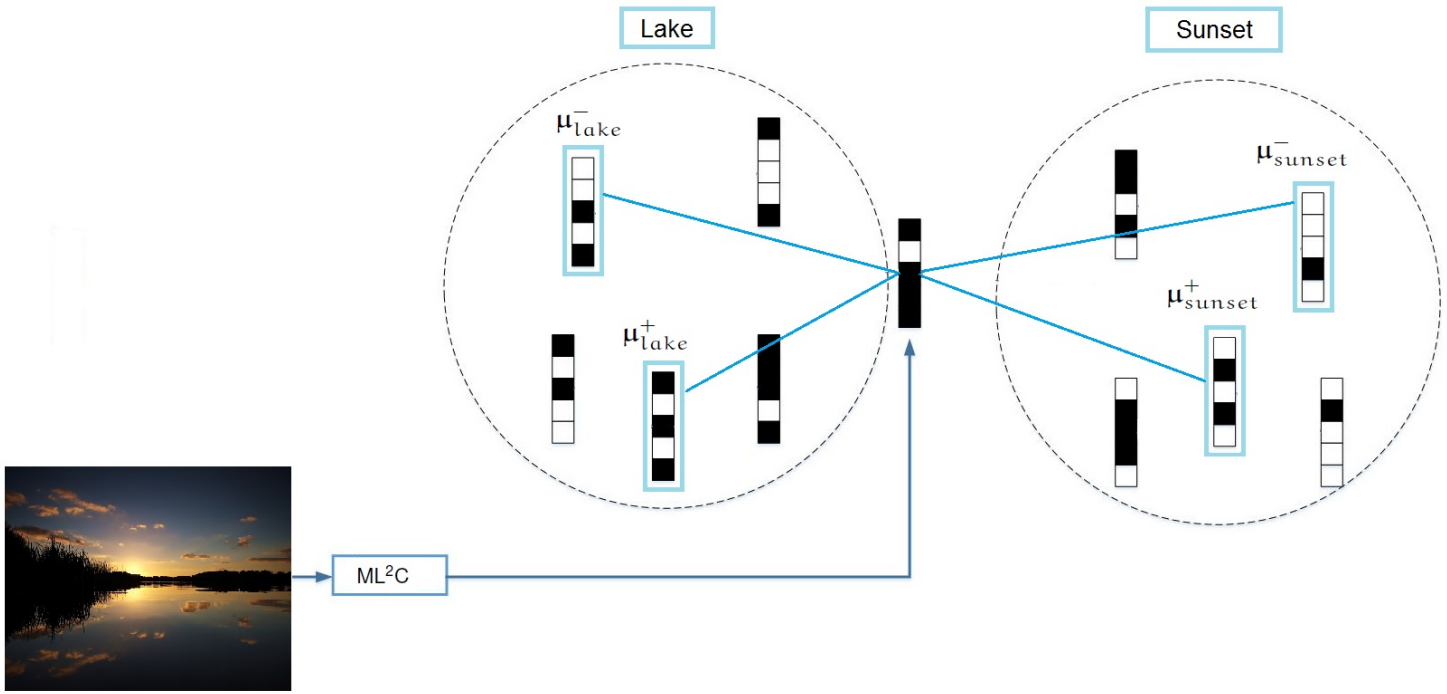
Section 2

Formulation

Formulation

Multi-label Learning via Codewords (ML^{2}C) utilizes

- Hash function/encoders to map the input data into a common B-bit code space.
- L pairs of codewords in the same space, where each pair is used for presence/absence of a particular label.
- The prediction of each label is based on the Hamming distance of the input's hash code with the codeword pair.



Formulation

- Data: $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n\in\mathbb{N}_N}$, where $\boldsymbol{x}_n \in \mathcal{X} \triangleq \mathbb{R}^D$ and $\boldsymbol{y}_n \in \mathbb{H} \triangleq \{-1, 1\}^L$
- SOPs are performed in the hash space.
  - True labels are represented by codewords set $\boldsymbol{M} \triangleq [\boldsymbol{\mu}_1(1), \boldsymbol{\mu}_1(-1), ..., \boldsymbol{\mu}_L(1), \boldsymbol{\mu}_L(-1)] \in \mathbb{R}^{B\times 2L}$
  - Weight parameters: $\boldsymbol{W} \triangleq [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_B] \in \mathbb{R}^{D\times B}$, where $\boldsymbol{w}_b \in \mathbb{R}^D$ corresponds to each bit.
- Hash code:

$$\boldsymbol{h}(\boldsymbol{x}) = \underset{\boldsymbol{y}\in\mathbb{H}^L}{\arg\max}\, f(\boldsymbol{x}, \boldsymbol{y}) \tag{1}$$

- Scoring function for each label $l \in \mathbb{N}_L$:

$$f_b(\boldsymbol{x}_n, h_b; \boldsymbol{w}_b) = \langle \boldsymbol{w}_b, \phi_b(\boldsymbol{x}_n, h_b)\rangle \ \forall b \in \mathbb{N}_B$$
$$f(\boldsymbol{x}_n, \boldsymbol{h}; \boldsymbol{W}) = \boldsymbol{x}_n^T \boldsymbol{W}\boldsymbol{h} \tag{2}$$

- Feature map for each label $l \in \mathbb{N}_L$:

$$\phi_b(\boldsymbol{x}_n, h_b) = h_b \boldsymbol{x}_n$$
$$\phi(\boldsymbol{x}_n, h) = \boldsymbol{x}_n \boldsymbol{h}^T \tag{3}$$

Formulation

- By adjusting parameters W and M, ML^{2}C attempts to reduce the distortion measure:

$$E(\boldsymbol{W}, \boldsymbol{M}) \triangleq \sum_{n \in \mathbb{N}_{\mathcal{N}}} \sum_{l \in \mathbb{N}_L} [y_{nl} = 1] d_H(\boldsymbol{\mu}_l(1), \boldsymbol{h}(\boldsymbol{x}_n)) + [y_{nl} = -1] d_H(\boldsymbol{\mu}_l(-1), \boldsymbol{h}(\boldsymbol{x}_n)) \quad (4)$$

where d_H is the Hamming distance defined as

$$d_H(\boldsymbol{\mu}_l(y_{nl}), \boldsymbol{h}(\boldsymbol{x}_n)) \triangleq \sum_b [\mu_{lb}(y_{nl}) h_b(\boldsymbol{x}_n) < 0], b \in \mathbb{N}_B \quad (5)$$

- Label prediction in the original space

$$\hat{y}_l(\boldsymbol{x}_n) = \underset{y_l \in \{-1,1\}}{\arg\min} \; d_H(\boldsymbol{\mu}_l(y_l), \boldsymbol{h}(\boldsymbol{x}_n)), \; \forall l \in \mathbb{N}_L \quad (6)$$

Section 3

Algorithm

Algorithm - Observations

- We decompose the error of the final predicted label $\hat{\boldsymbol{y}}$ to that of each label l: $Err(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{l \in \mathbb{N}_L} err(y_l, \hat{y}_l)$
- Based on 4, the label-wise error can be defined as

$$err(y_l, \hat{y}_l) \equiv \Delta_l(\boldsymbol{\mu}_l(y_l), \hat{\boldsymbol{h}}) = [y_l = 1] d_H(\boldsymbol{\mu}_l(1), \hat{\boldsymbol{h}}) + [y_l = -1] d_H(\boldsymbol{\mu}_l(-1), \hat{\boldsymbol{h}}) \tag{7}$$

- This problem is non-convex, and hence difficult to optimize directly.
- Upperbounding the error function with a convex loss over parameters could be helpful.
- A two-step Block Coordinate Descent algorithm would optimize the surrogate loss.

Algorithm - Surrogate Loss

- The error defined in Eq. (7) has the following surrogate loss:

$$\Delta_l(\boldsymbol{\mu}_l(y_l), \hat{\boldsymbol{h}}) \leq \sup_{\boldsymbol{h}' \in \mathbb{H}^B} \Delta_l(\boldsymbol{\mu}_l(y_l), \boldsymbol{h}') + f(\boldsymbol{x}_n, \boldsymbol{h}'; \boldsymbol{W}) - f(\boldsymbol{x}_n, \boldsymbol{\mu}_l(y_l); \boldsymbol{W}) \quad (8)$$

- Considering a regularization term on $\boldsymbol{W}$, one can alternatively solve the following problem:

$$\min_{\boldsymbol{W},\boldsymbol{M}} \sum_{n \in \mathbb{N}_\mathcal{N}} \sum_{l \in \mathbb{N}_L} \sup_{\boldsymbol{h}' \in \mathbb{H}^B} \{\Delta_l(\boldsymbol{\mu}_l(y_l), \boldsymbol{h}') + f(\boldsymbol{x}_n, \boldsymbol{h}'; \boldsymbol{W}) - f(\boldsymbol{x}_n, \boldsymbol{\mu}_l(y_l); \boldsymbol{W}) + \tfrac{\lambda}{2} \|\boldsymbol{W}\|_F^2\} \quad (9)$$

Algorithm - First block minimization

- Prob. (10) is equivalent to Prob. (9) through
  - Defining slack variables ξ_{nl} which represent maximal value of the surrogate loss,
  - Reforming the surrogate loss as constraints

$$\min_{\boldsymbol{W}} \sum_{n\in\mathbb{N}_{\mathcal{N}}} \sum_{l\in\mathbb{N}_L} \xi_{nl} + \frac{\lambda}{2} \|\boldsymbol{W}\|_F^2 \tag{10}$$

$$s.t.\ f(\boldsymbol{x}_n, \boldsymbol{\mu}_l(y_l); \boldsymbol{W}) - f(\boldsymbol{x}_n, \boldsymbol{h}'; \boldsymbol{W}) \geq \Delta_l(\boldsymbol{\mu}_l(y_l), \boldsymbol{h}') - \xi_{nl}\ \forall\ \boldsymbol{h}' \in \mathbb{H}^B$$

- Prob. (10) is easily recognized as the standard formulation of Structured Support Vector Machine (SSVM) ($\mathcal{O}(L2^B)$).

Algorithm - Second block minimization

- Codewords Optimization Step:

$$\min_{\boldsymbol{M}} \sum_{n\in\mathbb{N}_{\mathcal{N}}} \sum_{l\in\mathbb{N}_L} \sup_{\boldsymbol{h}'\in\mathbb{H}^B} \{\Delta_l(\boldsymbol{\mu}_l(y_l), \boldsymbol{h}') + f(\boldsymbol{x}_n, \boldsymbol{h}'; \boldsymbol{W}) - f(\boldsymbol{x}_n, \boldsymbol{\mu}_l(y_l); \boldsymbol{W})\} \quad (11)$$

- Having W fixed, the supremum value of the surrogate loss w.r.t. $\boldsymbol{h}'$ is attained by $\tilde{\boldsymbol{h}}$, calculated in the previous block.
- The problem is decomposable to L separate optimization problems in terms of $\boldsymbol{\mu}_l(y_l)$, which can be solved in parallel:

$$\min_{\boldsymbol{\mu}_l(y_l)} \sum_{n\in\mathbb{N}_{\mathcal{N}}} [y_{nl}=1][\mu_l(1)\tilde{h}_n < 0] + [y_{nl}=-1][\mu_l(-1)\tilde{h}_n < 0]$$
$$+ \boldsymbol{w}_b^T \boldsymbol{x}_n(\tilde{h}_n - [y_{nl}=1]\mu_l(1) - [y_{nl}=-1]\mu_l(-1)) \quad (12)$$

- Prob. (12) can be further simplified in terms of bit-wise quantities, and is solved by simple substitution ($\mathcal{O}(LB)$).

Algorithm - Optimization of Prob. (4)

Input: Codeword Bit Length B, Training Samples X with multi-labels Y.
Output: $\boldsymbol{W}$, $\boldsymbol{M}$.
01. Initialize $\boldsymbol{W}$, $\boldsymbol{M}$.
02. **While Not Converged**
03. Weight Optimization Step:
04. **For each label**
05. Score Function $\leftarrow$ *Eq*. (2).
06. Feature Map $\leftarrow$ *Eq*. (3).
07. Cost Function $\leftarrow$ *Eq*. (7).
08. Solve SSVM with `SSVM-MATLAB`.
09. **End For**
10. Codewords Optimization Step: Optimize over Prob. (11).
11. **End While**

The algorithm has $\mathcal{O}(L2^B)$ overall complexity, which is computationally superior to the state-of-the-art.

Section 4

Experiments

Datasets

A subset of following datasets are used for different experiments.

Dataset	#Inst	#Dim	#L	Card	Den	Dist	Dom
Birds	645	260	19	1.014	0.053	133	Audio
CAL500	502	68	174	26.044	0.150	502	Music
Emotions	593	72	6	1.869	0.311	27	Music
Scene	2407	294	6	1.074	0.179	15	Image
Yeast	2417	103	14	4.237	0.303	198	Biology
Rcv1v2-s1	6000	47236	101	2.880	0.029	1028	Text
Rcv1v2-s2	6000	47236	101	2.634	0.026	954	Text
Rcv1v2-s3	6000	47236	101	2.614	0.026	939	Text
Rcv1v2-s4	6000	47236	101	2.484	0.025	816	Text
Rcv1v2-s5	6000	47236	101	2.642	0.026	946	Text

Training Time Comparison

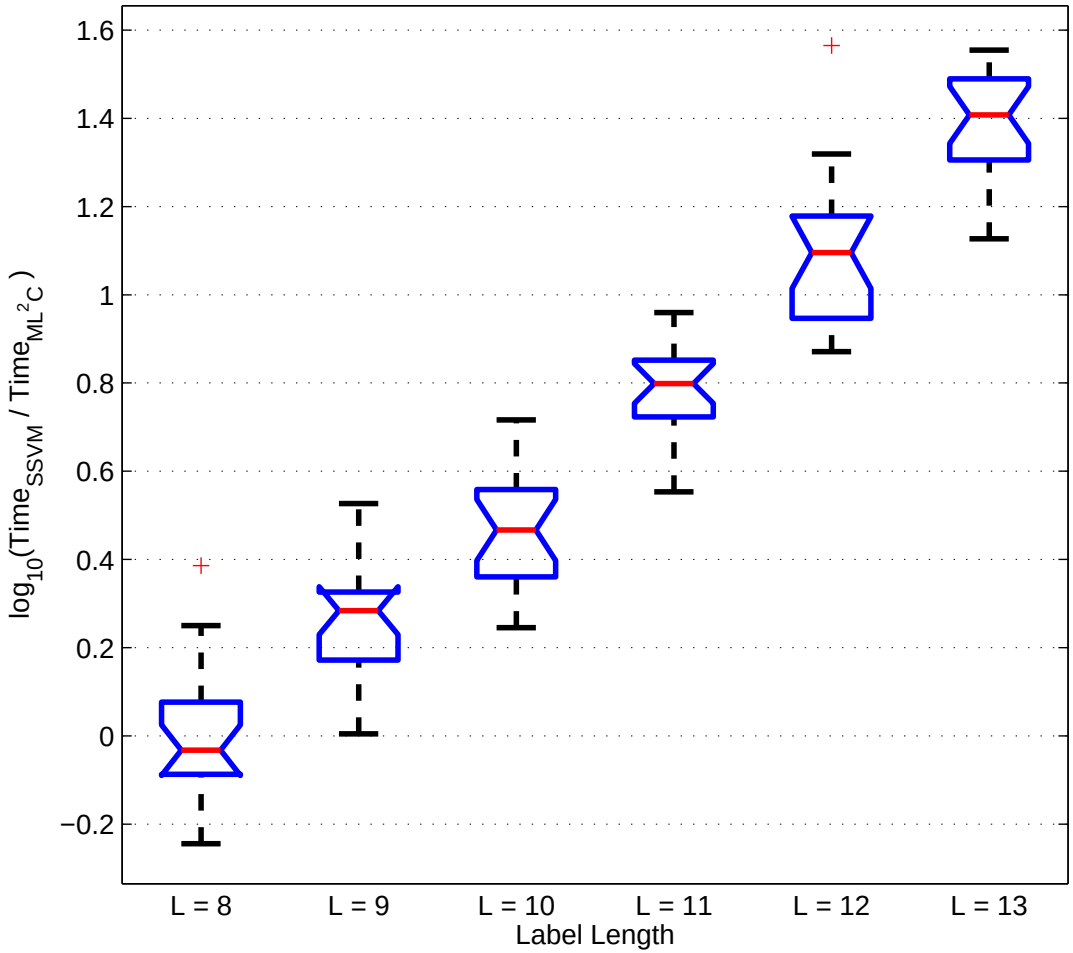


Figure: Comparison of training time for ML^{2}C and SSVM on the *Birds* data set. ML^{2}C used a code word length of $B = 5$.

Performance Comparison - Methods and Measures

- Methods to compare with:
  - Classifier Chains (CC) [Read et al., 2011]
  - Calibrated Label Ranking (CLR) [Fürnkranz et al., 2008]
  - Ensemble of Classifier Chains (ECC) [Read et al., 2011]
  - Multi-Label K Nearest Neighbor (ML-KNN) [Zhang and Zhou, 2007]
  - Binary Relevance - Support Vector Machine (BR-SVM) [Godbole and Sarawagi, 2004]
- Evaluation Measures:
  - Hamming Loss $\triangleq \frac{1}{N}\sum_{n\in\mathbb{N}_N}|\boldsymbol{y}_n \cap \hat{\boldsymbol{y}}_n|$
  - $micro - F_1 \triangleq \frac{\sum_{l\in\mathbb{N}_L} 2TP_l}{\sum_{l\in\mathbb{N}_L} 2TP_l+FP_l+FN_l}$
  - $macro - F_1 \triangleq \frac{1}{L}\sum_{l\in\mathbb{N}_L} \frac{2TP_l}{2TP_l+FP_l+FN_l}$

Performance Comparison - Hamming Loss

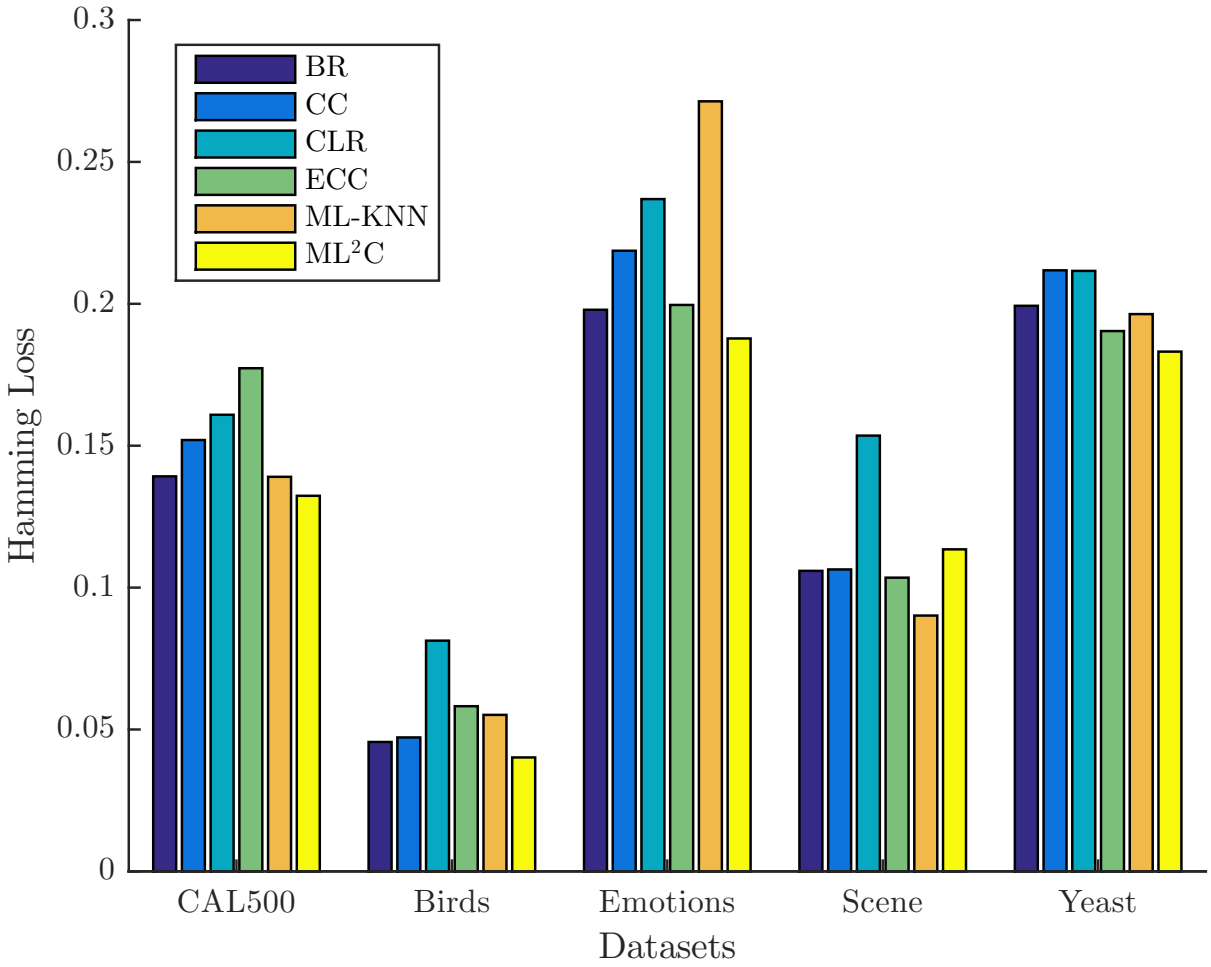


Figure: Average Hamming Loss of 6 algorithms over 5 medium-size datasets. The smaller the Hamming Loss is, the better the algorithms perform.

Performance Comparison - *micro* $- F_1$

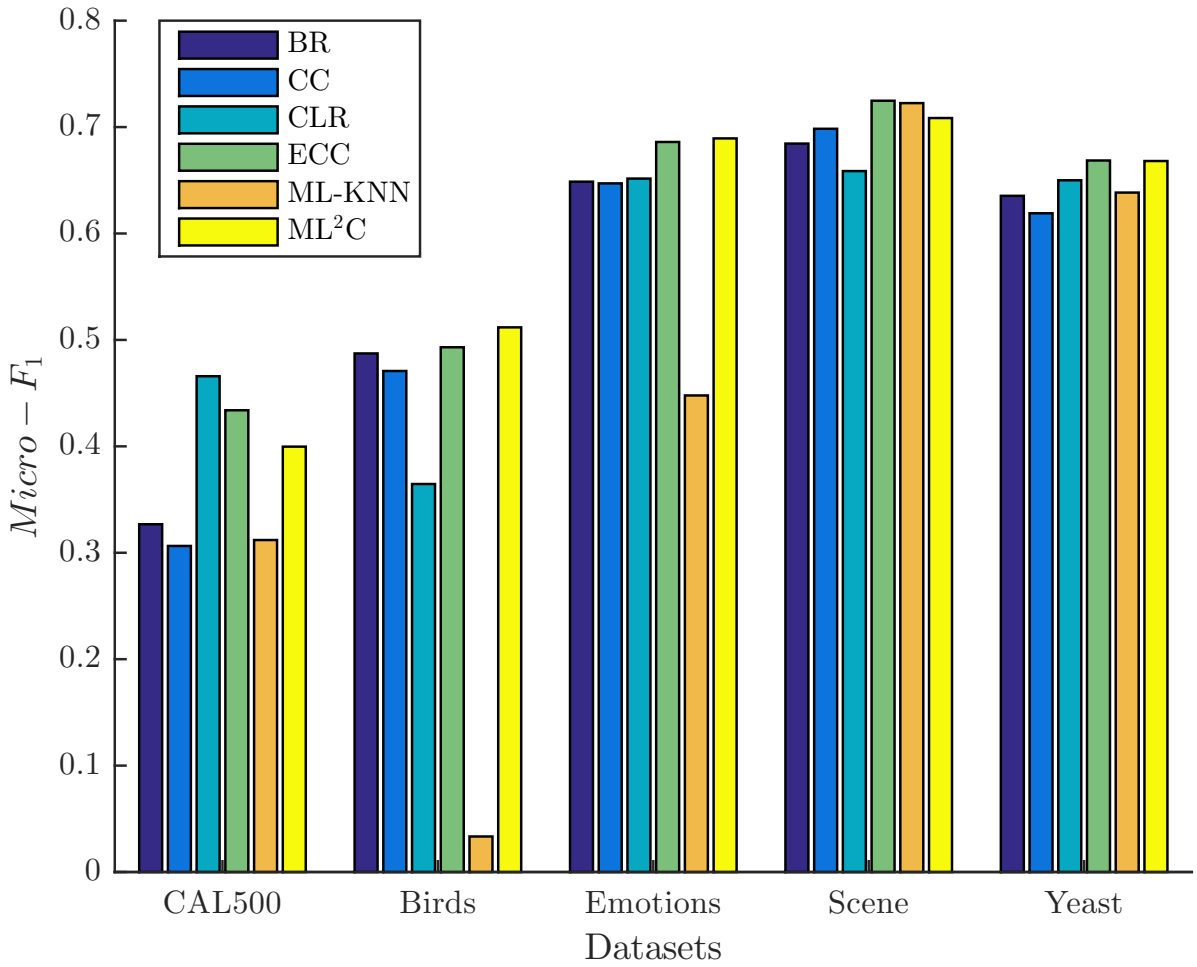


Figure: Average *micro* $- F_1$ of 6 algorithms over 5 medium-size datasets. The larger the F_1 measures are, the better the algorithms perform.

Performance Comparison - *macro* $- F_1$

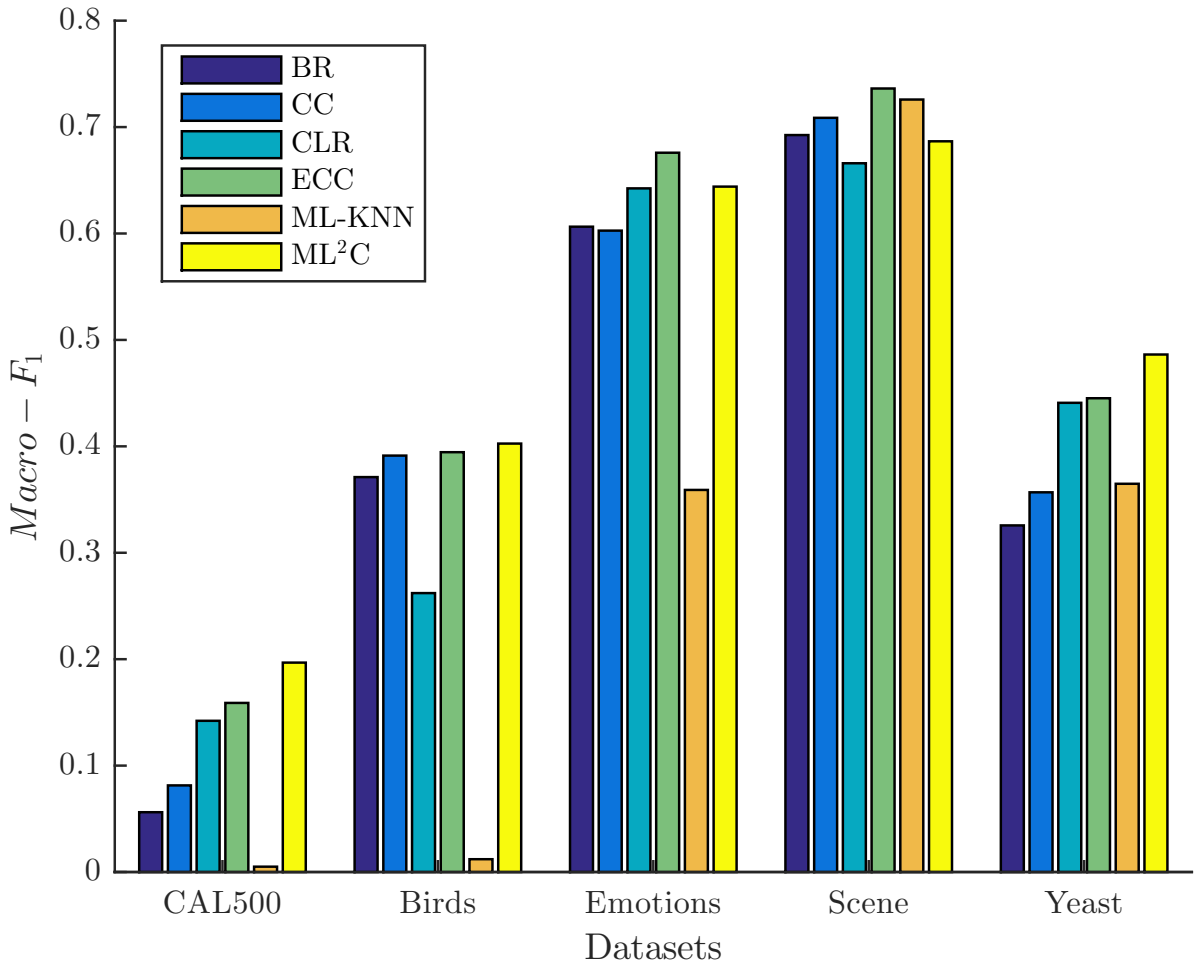


Figure: Average *macro* $- F_1$ of 6 algorithms over 5 medium-size datasets. The larger the F_1 measures are, the better the algorithms perform.

Performance Comparison - Ranking

Table: Ranking results of all algorithms on 5 benchmark datasets of medium size. For each dataset, the statistically best and comparable results for a family-wise error significance level of 0.05 are highlighted in boldface. All algorithms are ranked from best to worst. If the performance is statistically comparable, the algorithms share the same rank.

Dataset	ML^{2}C	BR-SVM	ML-KNN	ECC	CLR	CC
CAL500	**1st**	*2nd*	*2nd*	*5th*	*4th*	*3rd*
Birds	**1st**	*2nd*	*2nd*	**1st**	*3rd*	**1st**
Emotions	**1st**	*2nd*	*5th*	*2nd*	*4th*	*3rd*
Scene	*3rd*	**1st**	*2nd*	*2nd*	**1st**	*2nd*
Yeast	**1st**	*3rd*	*2nd*	**1st**	*4th*	*4th*

Performance Comparison - Hamming Loss

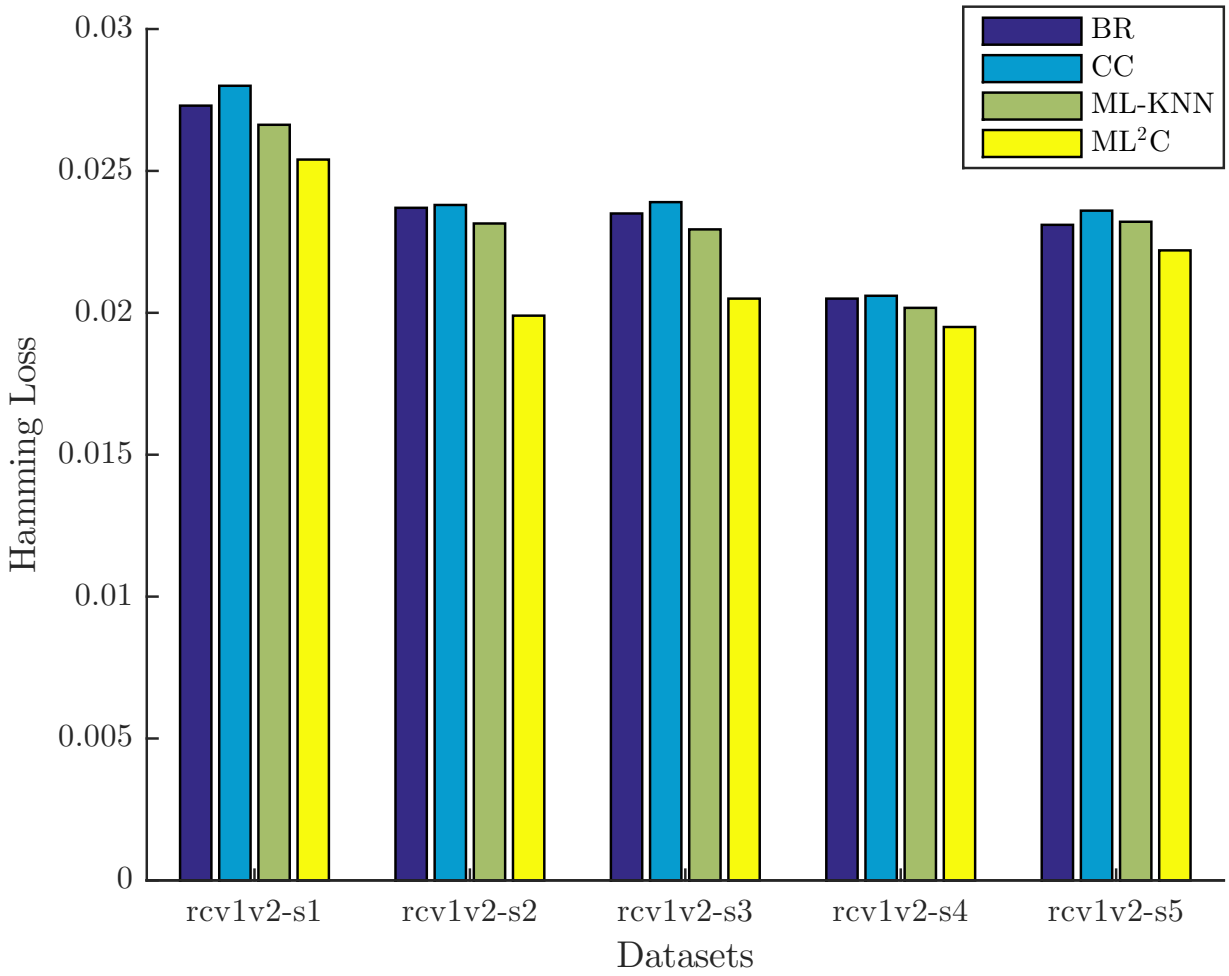


Figure: Average Hamming Loss of 6 algorithms over 5 large-size datasets. The smaller the Hamming Loss is, the better the algorithms perform.

Performance Comparison - *micro* $- F_1$

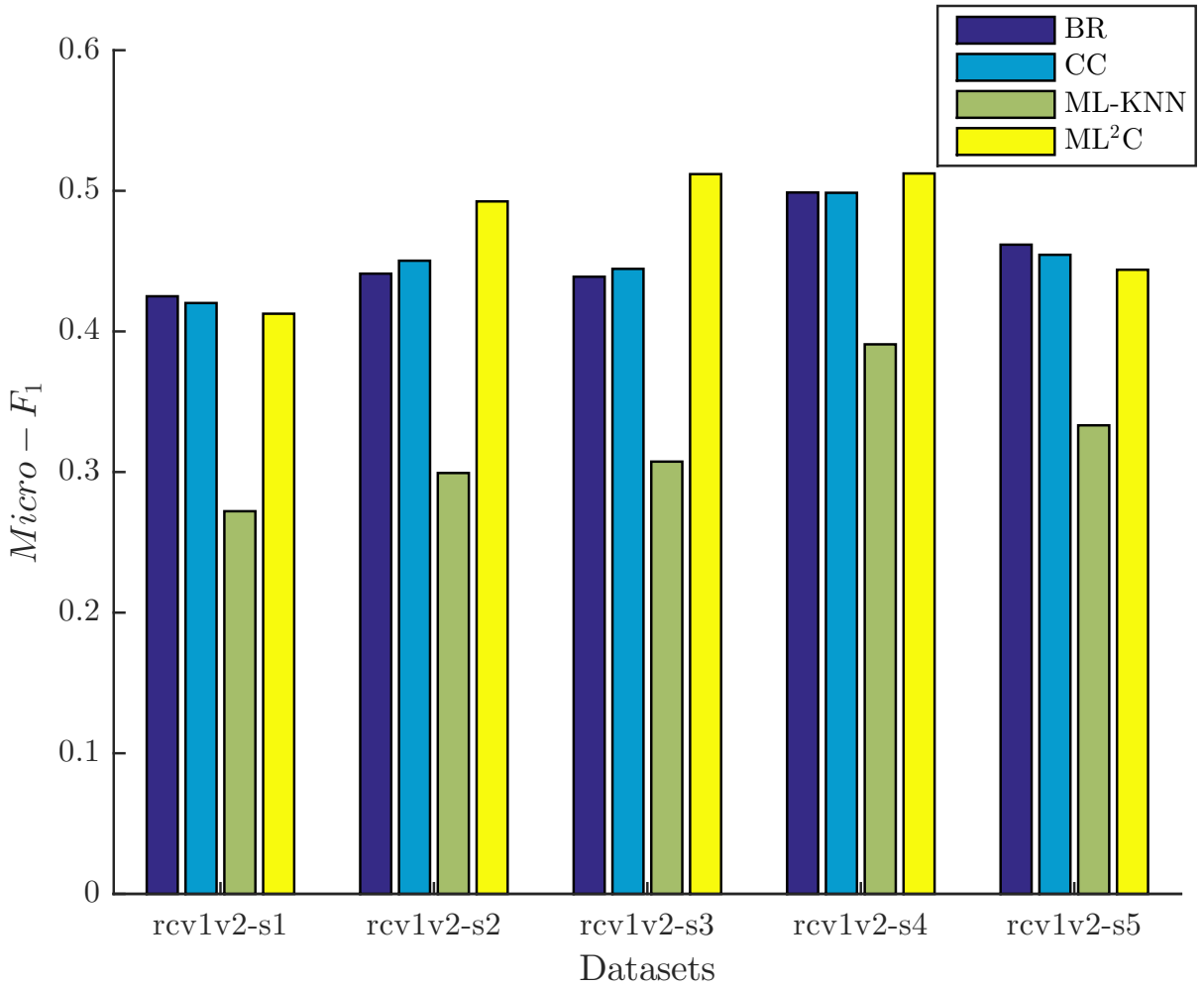


Figure: Average *micro* $- F_1$ of 6 algorithms over 5 large-size datasets. The larger the F_1 measures are, the better the algorithms perform.

Performance Comparison - *macro* − F_1

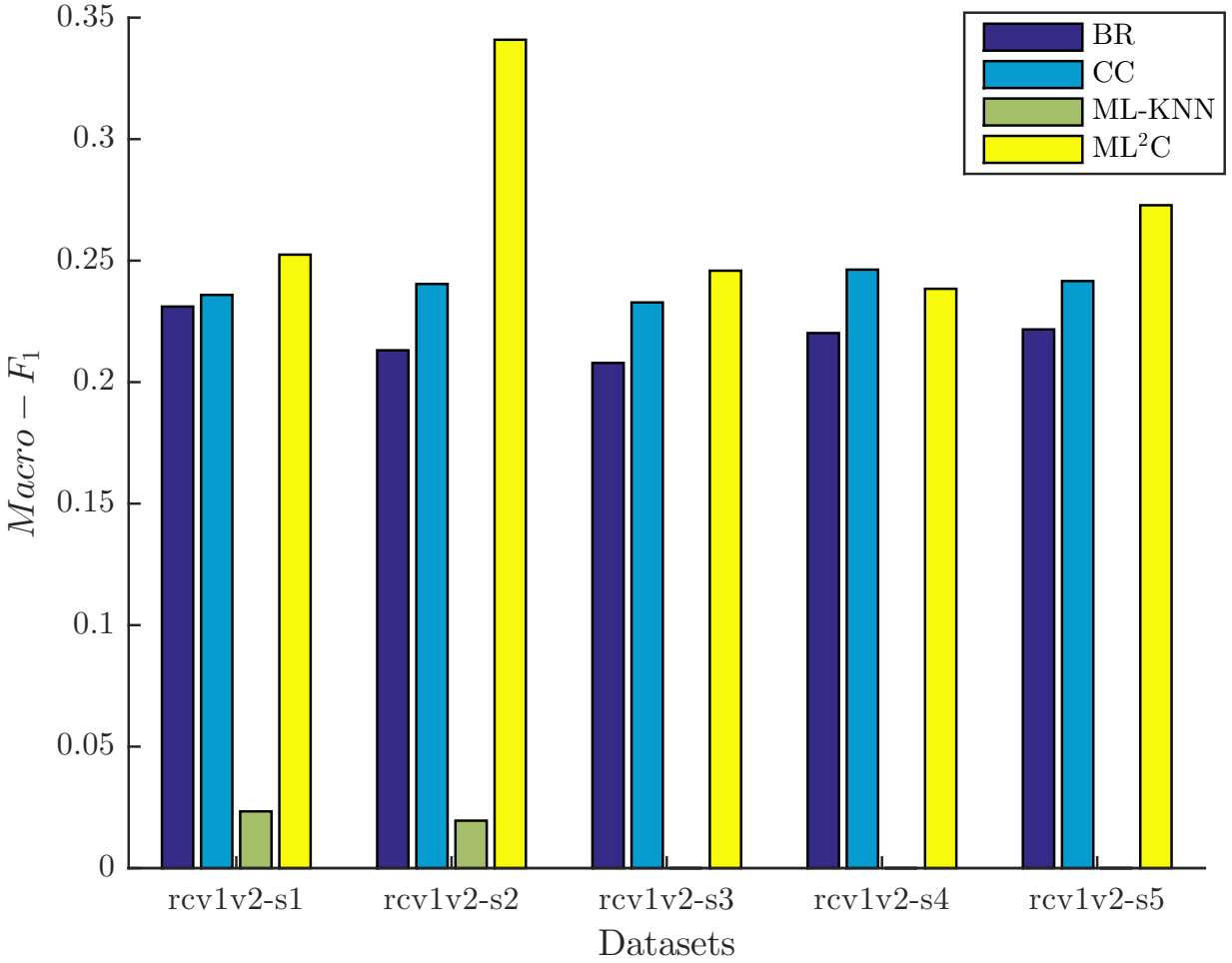


Figure: Average *macro* − F_1 of 6 algorithms over 5 large-size datasets. The larger the F_1 measures are, the better the algorithms perform.

Performance Comparison - Ranking

Table: Ranking results of algorithms on 5 benchmark datasets of large size. For each dataset, the statistically best and comparable results for a family-wise error significance level of 0.05 are highlighted in boldface. All algorithms are ranked from best to worst. If the performance is statistically comparable, the algorithms share the same rank.

Data Sets	ML^{2}C	BR-SVM	ML-KNN	CC
Rcv1v2-s1	**1st**	*3rd*	*2nd*	*3rd*
Rcv1v2-s2	**1st**	*3rd*	*2nd*	*3rd*
Rcv1v2-s3	**1st**	*3rd*	*2nd*	*3rd*
Rcv1v2-s4	**1st**	*2nd*	**1st**	*2nd*
Rcv1v2-s5	**1st**	*2nd*	*2nd*	*3rd*

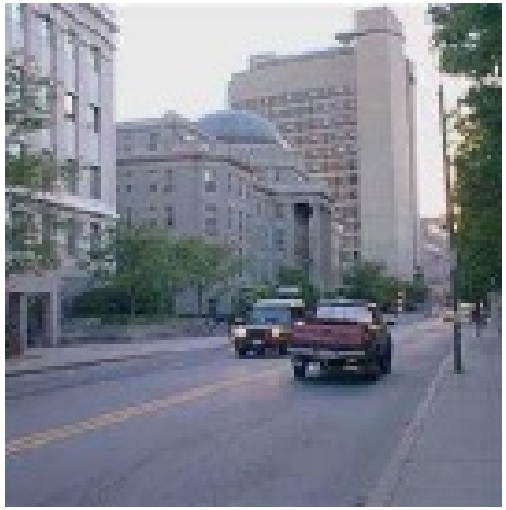
Image Annotation

Table: *Street* scene of the *SUN* dataset. Wrong annotations (red), missed annotations (inside parentheses).

			
Ground Truth	Car, Building, Sidewalk, Road, Sky, Tree	Car, Building, Sidewalk, Road, Sky, Tree	Car, Building, Sidewalk, Road, Person, Sky, Tree
ML^{2}C	Car, Building, Sidewalk, Road, Sky, Tree	Car, Building, Sidewalk, Road, Sky, Tree	Car, Building, Sidewalk, Road, Person, Sky, Tree
BR-SVM	Car, Building, Sidewalk, Road, Person, Sky, (Tree)	Car, Building, Sidewalk, Road, Person, Sky, (Tree)	Car, Building, Sidewalk, Road, Person, Sky, (Tree)
ML-KNN	Car, Building, Sidewalk, Road, Person, Sky, (Tree)	Car, Building, Sidewalk, Road, Person, Sky, (Tree)	Car, Building, Sidewalk, Road, Person, Sky, (Tree)
ECC	Car, Building, (Tree), Window,Sidewalk, Road, Sky	Car, Building, Sidewalk, Road, Sky, (Tree)	Car, Building, Sidewalk, Road, (Person), Sky, Tree
CLR	Car, Building, Tree, Window,Sidewalk, Road, Sky	Car, Building, Sidewalk, Road, Sky, (Tree)	Car, Building, Sidewalk, Road, (Person), Sky, Tree
CC	Car, Building, Sidewalk, Road, Window, Sky, (Tree)	Car, Building, Sidewalk, Road, Person, Sky, (Tree)	Car, Building, Sidewalk, Road, (Person), Sky, Tree

Image Annotation

Table: *Mountain* scene of the *SUN* dataset. Wrong annotations (red), missed annotations (inside parentheses).

			
Ground Truth	Mountain, Sky	Mountain, Sky, Snow	Mountain, Sky
ML^{2}C	Mountain, Sky	Mountain, Sky, (Snow)	Mountain, Sky
BR-SVM	Mountain, Sky, Snow	Mountain, Sky, (Tree), Snow	Mountain, Sky, Tree, Snow
ML-KNN	Mountain, Sky, Tree	Mountain, Sky, (Tree), (Snow)	Mountain, Sky, Tree
ECC	Mountain, Sky, Tree, Snow	Mountain, Sky, Tree, (Snow)	Mountain, Sky, Tree
CLR	Mountain, Sky	Mountain, Sky, (Snow)	Mountain, Sky, Tree
CC	Mountain, Sky, Snow	Mountain, Sky, Tree, (Snow)	Mountain, Sky, Tree

Section 5

Summary

Summary

- A novel multi-label learning framework - Multi-label Learning via Codewords (ML^{2}C) is proposed.
- ML^{2}C reduces the training time complexity from typical Structured Output Prediction (SOP) approaches ($\mathcal{O}(2^L)$) to $\mathcal{O}(L2^B)$.
- Its training algorithm is simple to implement.
- Experiments on 10 benchmark datasets, compared with 5 other state-of-art methods, show ML^{2}C is superior.

Thank You!

Thanks for your time.

Section 6

References

References I


Fürnkranz, J., Hüllermeier, E., LozaMencía, E., and Brinker, K. (2008).
Multilabel classification via calibrated label ranking.
Machine Learning, 73(2):133–153.


Godbole, S. and Sarawagi, S. (2004).
Discriminative methods for multi-labeled classification.
In *Pacific-Asia conference on knowledge discovery and data mining*, pages 22–30. Springer.


Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011).
Classifier chains for multi-label classification.
Machine Learning, 85:333–359.


Zhang, M.-l. and Zhou, Z.-h. (2007).
Ml-knn: A lazy learning approach to multi-label learning.
Pattern Recognition, 40:2007.