Learning to Calibrate and Rerank Multi-label Predictions

Cheng Li, Virgil Pavlu, Javed Aslam, Bingyu Wang, Kechen Qin

Khoury College of Computer Sciences
Northeastern University

Multi-label Classification: example

Flickr Image Tagging

- ☐ airport
- ☐ animal
- ☑ clouds
- ☐ book
- ☑ lake
- ☑ sunset
- ☑ sky
- ☐ cars
- ☑ water
- ☑ reflection
- ...

Multi-label Classification: example

Reuters News Article Categorization

Breakingviews

Twitter may score big with football digital rights

By Jennifer Saba | April 5, 2016

Tags: INTERNET | NFL | SPORTS | SPORTS BUSINESS | TWITTER

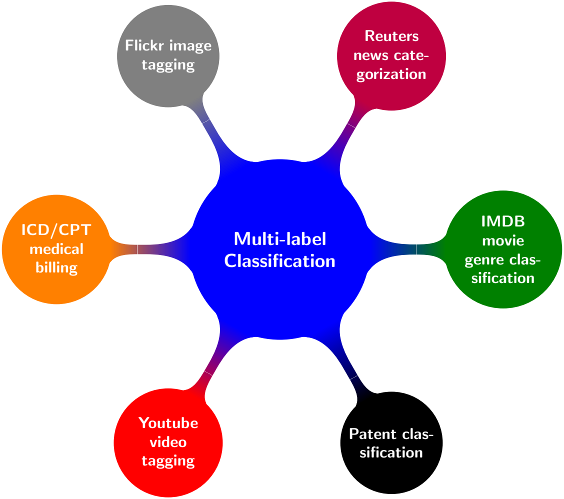
The author is a Reuters Breakingviews columnist. The opinions expressed are her own.

Twitter may finally be gaining some ground. Chief Executive Jack Dorsey's social-media company has won the rights to stream National Football League games on 10 Thursday nights for roughly $10 million, according to technology site Re/code. That's about the price of a one-minute Super Bowl commercial. After fumbling with stalled growth in the number of users, Twitter may have found a cheap way to stay on the field with rivals like Facebook.

- [x] Internet
- [ ] crime
- [x] NFL
- [ ] government
- [ ] Asia
- [x] sports
- [ ] politics
- [x] sports business
- [x] Twitter

...

Multi-label Classification: many applications



Multi-label Classification: mathematical formulation

$$\mathbf{x} \xrightarrow{h} \mathbf{Y} = [Y_1, Y_2, ..., Y_L] = \overbrace{[1, 0, 0, 1, 0, ..., 1]}^{\text{length } L}$$

L: # candidate labels

$\mathbf{x}$: instance features

h: multi-label classifier (to be built)

$\mathbf{Y}$: label subset, written as binary vector of length L

$$Y_\ell = \begin{cases} 1 & \text{label } \ell \text{ applies} \\ 0 & \text{label } \ell \text{ does not apply} \end{cases}, \ \ell = 1, 2, ..., L$$

Binary vs Multi-class vs Multi-label

- binary classification: 1 out of 2
- multi-class classification: 1 out of many
- **multi-label classification**: many out of many

Binary Relevance (BR) Method

- train one binary classifier $p(Y_\ell|\mathbf{x})$ for each label ℓ
- predict each label independently: predict label ℓ if $p(Y_\ell = 1|\mathbf{x}) > 0.5$
- prediction confidence $p(\mathbf{Y}|\mathbf{x}) = p(Y_1|\mathbf{x}) \times p(Y_2|\mathbf{x}) \times \cdots \times p(Y_L|\mathbf{x})$

Pros and Cons

- ☺ faster than many other methods
- ☺ easy to implement
- ☹ make mistakes due to ignoring label dependencies
- ☹ does not provide calibrated confidence

New Multi-label Method: BR-rerank

- capture label dependencies
- maintain the simplicity of BR
- rerank BR's predictions to improve its accuracy
- post-calibrate BR's confidence scores

BR's drawback: ignoring label dependencies

☹ Make invalid predictions that violate label constraints:

$\texttt{cat} \Longrightarrow \texttt{animal}$

- ☑ `cat`
- ☐ `animal`
- ☐ `person`
- ☐ `building`
- ☐ `car`

BR's drawback: ignoring label dependencies

☹ May not handle difficult labels well.

- `clouds,lake,sunset,sky,water`: easy to predict directly
- `reflection`: hard to predict directly

- ☐ `airport`
- ☐ `animal`
- ☑ `clouds`
- ☐ `book`
- ☑ `lake`
- ☑ `sunset`
- ☑ `sky`
- ☐ `cars`
- ☑ `water`
- ☐ `reflection`

BR's drawback: ignoring label dependencies

Better solution:

☺ Let easy labels help difficult labels

- `clouds,lake,sunset,sky,water`: easy to predict directly
- `reflection`: often co-occurs with `water` and `lake`

- ☐ `airport`
- ☐ `animal`
- ☑ `clouds`
- ☐ `book`
- ☑ `lake`
- ☑ `sunset`
- ☑ `sky`
- ☐ `cars`
- ☑ `water`
- ☑ `reflection`

BR-rerank: rerank BR's predictions

- ground truth: {`person`, `baseball bat`, `baseball glove`}
- BR predicts: {`person`, `baseball bat`}

BR-rerank: two stage prediction

input

BR-rerank: two stage prediction

input

BR

BR-rerank: two stage prediction

input **marginals**

BR

$p(\text{person}|\mathbf{x}) = 0.99$
$p(\text{baseball bat}|\mathbf{x}) = 0.99$
$p(\text{baseball glove}|\mathbf{x}) = 0.37$
$p(\text{handbag}|\mathbf{x}) = 0.03$
$p(\text{sports ball}|\mathbf{x}) = 0.03$
$p(\text{bench}|\mathbf{x}) = 0.01$
$p(\text{bear}|\mathbf{x}) = 0.00$
$p(\text{bottle}|\mathbf{x}) = 0.00$
...

BR-rerank: two stage prediction

input

BR

marginals

$p(\text{person}|\mathbf{x}) = 0.99$
$p(\text{baseball bat}|\mathbf{x}) = 0.99$
$p(\text{baseball glove}|\mathbf{x}) = 0.37$
$p(\text{handbag}|\mathbf{x}) = 0.03$
$p(\text{sports ball}|\mathbf{x}) = 0.03$
$p(\text{bench}|\mathbf{x}) = 0.01$
$p(\text{bear}|\mathbf{x}) = 0.00$
$p(\text{bottle}|\mathbf{x}) = 0.00$
$\cdots$

set prediction candidates

- person, baseball bat
- person, baseball bat, baseball glove
- person, baseball bat, handbag
- person, baseball bat, sports ball
- person, baseball bat, baseball glove, handbag

BR-rerank: two stage prediction

input

BR

marginals

$p(\text{person}|\mathbf{x}) = 0.99$

$p(\text{baseball bat}|\mathbf{x}) = 0.99$

$p(\text{baseball glove}|\mathbf{x}) = 0.37$

$p(\text{handbag}|\mathbf{x}) = 0.03$

$p(\text{sports ball}|\mathbf{x}) = 0.03$

$p(\text{bench}|\mathbf{x}) = 0.01$

$p(\text{bear}|\mathbf{x}) = 0.00$

$p(\text{bottle}|\mathbf{x}) = 0.00$

$\cdots$

set prediction candidates	**set prediction features**
person baseball bat	BR score=0.58 $[1, 1, 0, 0, 0, \cdots]$ card=2 prior=3×10^{-3}
person baseball bat baseball glove	BR score=0.35 $[1, 1, 1, 0, 0, \cdots]$ card=3 prior=5×10^{-3}
person baseball bat handbag	BR score=0.02 $[1, 1, 0, 1, 0, \cdots]$ card=3 prior=2×10^{-5}
person baseball bat sports ball	BR score=0.02 $[1, 1, 0, 0, 1, \cdots]$ card=3 prior=1×10^{-3}
person baseball bat baseball glove handbag	BR score=0.01 $[1, 1, 1, 1, 0, \cdots]$ card=4 prior=1×10^{-5}

BR-rerank: two stage prediction

input

BR

marginals

$p(\text{person}|\mathbf{x}) = 0.99$
$p(\text{baseball bat}|\mathbf{x}) = 0.99$
$p(\text{baseball glove}|\mathbf{x}) = 0.37$
$p(\text{handbag}|\mathbf{x}) = 0.03$
$p(\text{sports ball}|\mathbf{x}) = 0.03$
$p(\text{bench}|\mathbf{x}) = 0.01$
$p(\text{bear}|\mathbf{x}) = 0.00$
$p(\text{bottle}|\mathbf{x}) = 0.00$
$\cdots$

set prediction candidates	set prediction features		reranker score
person baseball bat	BR score=0.58 $[1, 1, 0, 0, 0, \cdots]$ card=2 prior=3×10^{-3}	reranker	0.16
person baseball bat baseball glove	BR score=0.35 $[1, 1, 1, 0, 0, \cdots]$ card=3 prior=5×10^{-3}	reranker	0.17
person baseball bat handbag	BR score=0.02 $[1, 1, 0, 1, 0, \cdots]$ card=3 prior=2×10^{-5}	reranker	0.04
person baseball bat sports ball	BR score=0.02 $[1, 1, 0, 0, 1, \cdots]$ card=3 prior=1×10^{-3}	reranker	0.08
person baseball bat baseball glove handbag	BR score=0.01 $[1, 1, 1, 1, 0, \cdots]$ card=4 prior=1×10^{-5}	reranker	0.03

BR-rerank: two stage prediction

input

BR

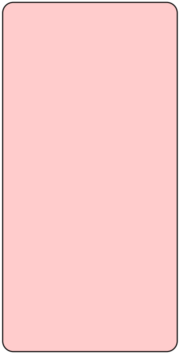
marginals

$p(\text{person}|\mathbf{x}) = 0.99$
$p(\text{baseball bat}|\mathbf{x}) = 0.99$
$p(\text{baseball glove}|\mathbf{x}) = 0.37$
$p(\text{handbag}|\mathbf{x}) = 0.03$
$p(\text{sports ball}|\mathbf{x}) = 0.03$
$p(\text{bench}|\mathbf{x}) = 0.01$
$p(\text{bear}|\mathbf{x}) = 0.00$
$p(\text{bottle}|\mathbf{x}) = 0.00$
$\cdots$

set prediction candidates	set prediction features		reranker score	
person baseball bat	BR score=0.58 $[1, 1, 0, 0, 0, \cdots]$ card=2 prior=3×10^{-3}	reranker	0.16	
person baseball bat baseball glove	BR score=0.35 $[1, 1, 1, 0, 0, \cdots]$ card=3 prior=5×10^{-3}	reranker	0.17	✓
person baseball bat handbag	BR score=0.02 $[1, 1, 0, 1, 0, \cdots]$ card=3 prior=2×10^{-5}	reranker	0.04	
person baseball bat sports ball	BR score=0.02 $[1, 1, 0, 0, 1, \cdots]$ card=3 prior=1×10^{-3}	reranker	0.08	
person baseball bat baseball glove handbag	BR score=0.01 $[1, 1, 1, 1, 0, \cdots]$ card=4 prior=1×10^{-5}	reranker	0.03	

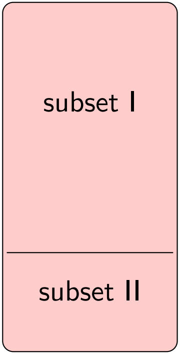
BR-rerank: two stage training

training data

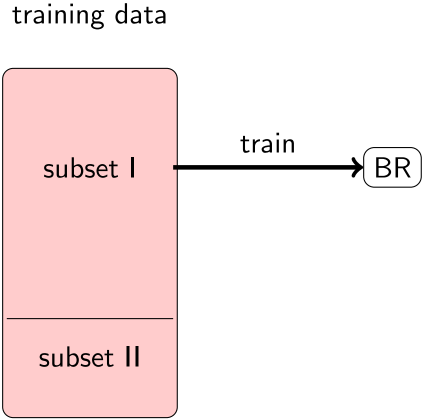


BR-rerank: two stage training

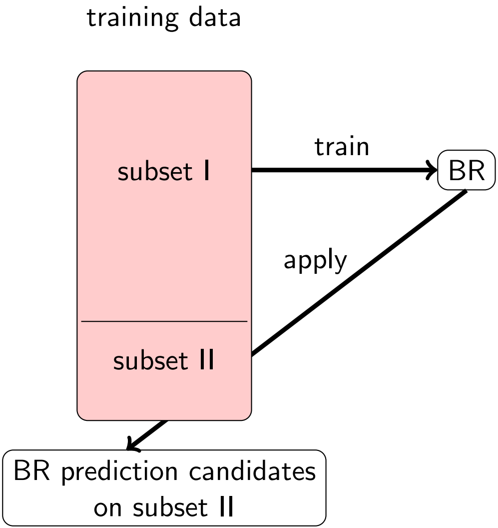
training data



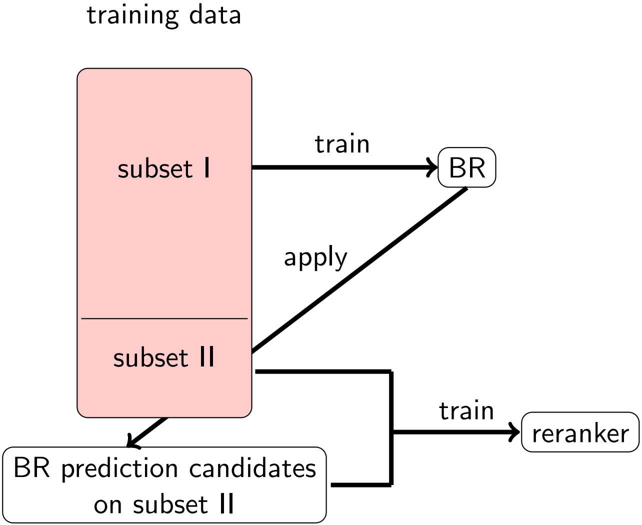
BR-rerank: two stage training



BR-rerank: two stage training



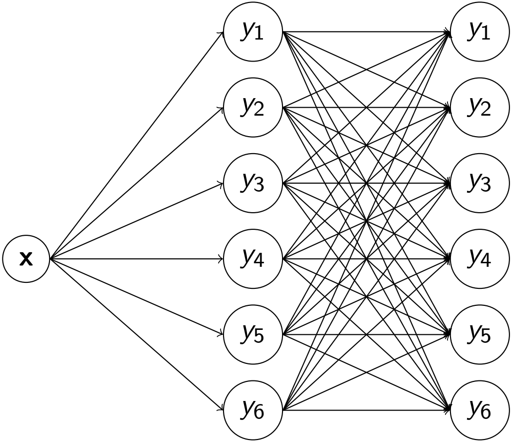
BR-rerank: two stage training



BR-rerank: two stage training

input	marginals	set prediction candidates	set prediction features		set prediction correctness (regression target)
		person, baseball bat	BR score=0.58 $[1, 1, 0, 0, 0, \cdots]$ card=2 prior=3×10^{-3}	reranker	0
		person, baseball bat, baseball glove	BR score=0.35 $[1, 1, 1, 0, 0, \cdots]$ card=3 prior=5×10^{-3}	reranker	1
 ground truth = {person, baseball bat, baseball glove} → BR →	$p(\text{person}\|\mathbf{x}) = 0.99$ $p(\text{baseball bat}\|\mathbf{x}) = 0.99$ $p(\text{baseball glove}\|\mathbf{x}) = 0.37$ $p(\text{handbag}\|\mathbf{x}) = 0.03$ $p(\text{sports ball}\|\mathbf{x}) = 0.03$ $p(\text{bench}\|\mathbf{x}) = 0.01$ $p(\text{bear}\|\mathbf{x}) = 0.00$ $p(\text{bottle}\|\mathbf{x}) = 0.00$ $\cdots$	person, baseball bat, handbag	BR score=0.02 $[1, 1, 0, 1, 0, \cdots]$ card=3 prior=2×10^{-5}	reranker	0
		person, baseball bat, sports ball	BR score=0.02 $[1, 1, 0, 0, 1, \cdots]$ card=3 prior=1×10^{-3}	reranker	0
		person, baseball bat, baseball glove, handbag	BR score=0.01 $[1, 1, 1, 1, 0, \cdots]$ card=4 prior=1×10^{-5}	reranker	0

How is BR-rerank Different from other Stacking Methods?



Stage 2 predictions:

- Other stacking methods: decide each label separately
- BR-rerank: finds the label set with the highest score

BR-rerank: classification accuracy

Table: set accuracy on test data

Dataset	BR	*BR-rerank*	2BR	DBR	CBM	CRF	SPEN	PDS	DVN	PC	PCC	Rakel	MLKNN
BIBTEX	16.6	21.5	16.1	20.2	22.9	**23.3**	14.8	16.1	16.2	20.3	21.4	18.3	8.4
OHSUMED	36.6	**42.0**	37.5	37.6	40.5	40.4	29.1	34.8	18.6	29.5	38.0	39.3	25.4
RCV1	44.5	53.2	42.3	45.8	**55.3**	53.8	27.5	40.8	13.7	39.7	48.7	46.0	46.2
TMC	30.4	**33.3**	32.1	31.7	30.8	28.2	26.7	23.4	20.3	23.0	31.3	27.6	18.9
WISE	52.9	60.5	51.8	55.8	**61.0**	46.4	-	52.4	28.3	-	55.9	3.5	2.4
MSCOCO	34.7	**35.9**	33.7	32.0	31.1	35.1	34.1	25.0	29.9	31.1	32.1	32.6	29.1
ranking	6.3	1.8	6.7	5.7	3.3	3.8	10.0	9.8	11.2	10.0	4.5	6.8	11.0

- BR-rerank performs much better than BR
- BR-rerank has the highest average ranking

BR-rerank: training time

Table: Training time of different methods, measured in seconds.

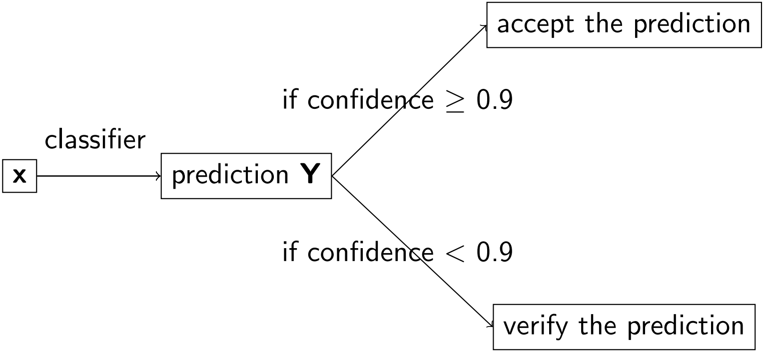
Dataset	BIBT	OHSUM	RCV1	TMC	WISE	MSCO
BR	4	3	7	8	80	1380
BR-rerank	9	6	10	11	88	1393
CBM	64	210	70	224	1320	8520
CRF	353	268	1223	771	16363	14760

- Reranking step does not add much overhead.
- BR-rerank is much faster than competitors CRF and CBM.

BR-rerank: calibrated confidence

Major benefit of BR-rerank: calibrated confidence, essential in real applications, but often overlooked in academia.

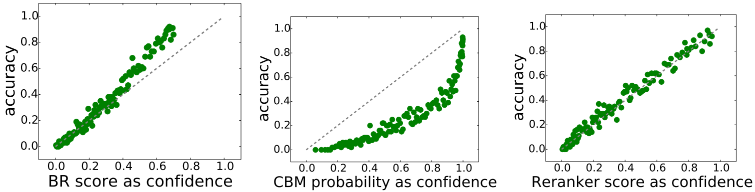
Example: using calibrated confidence to filter predictions to maintain target accuracy 0.9



BR-rerank: calibrated confidence

- ☺ calibrated confidence: score aligns with accuracy
  e.g., among all predictions with prediction score=0.7, 70% are actually correct (accuracy=70%)
- ☹ uncalibrated confidence: score does not align with accuracy
  e.g., among all predictions with prediction score=0.7, 50% are actually correct (over-confident)
  90% are actually correct (under-confident)

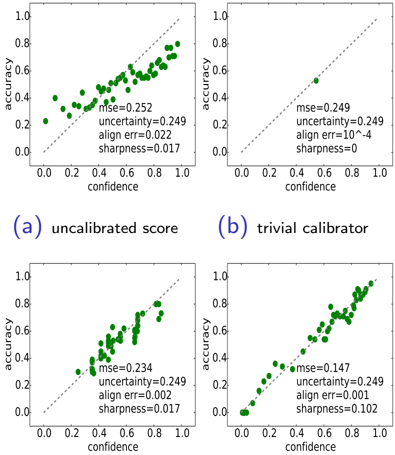
BR-rerank: calibrated confidence



- 3 models tested on MSCOCO test set
- each dot represents 100 predictions with similar confidence
- x-value=confidence, y-value=accuracy
- BR-rerank and CBM have similar overall classification accuracy. But CBM probabilities are over-confident, BR-rerank scores are well calibrated

Reranker vs other post-calibrators

- Prediction: BR predictions on WISE test set
- Calibrators: none vs trivial vs isotonic regression vs reranker



(a) uncalibrated score (b) trivial calibrator

(c) isotonic regression (d) reranker calibrator

Evaluation metrics for calibration

- $c(\mathbf{Y}) \in [0,1]$ confidence score
- $v(\mathbf{Y}) \in \{0,1\}$ 0/1 correctness
- $e(\mathrm{c}) = p[v(\mathbf{Y}) = 1 | c(\mathbf{Y}) = \mathrm{c}]$ is the average set accuracy among all predictions whose confidence is c.
- Alignment error: $\mathbb{E}[e(c(\mathbf{Y})) - c(\mathbf{Y})]^2$; the discrepancy between the claimed confidence and the actual accuracy. The smaller the better.
- Sharpness:$\mathrm{Var}[e(c(\mathbf{Y}))]$; how widely spread the confidence scores are. The bigger the better.
- The mean squared error (MSE, also called Brier Score): $\mathbb{E}[(v(\mathbf{Y}) - c(\mathbf{Y}))^2]$; the difference between the confidence and the actual 0/1 correctness.

$$\underbrace{\mathbb{E}[(v(\mathbf{Y}) - c(\mathbf{Y}))^2]}_{MSE} = \underbrace{\mathbb{E}[(e(c(\mathbf{Y})) - c(\mathbf{Y}))^2]}_{alignment\ \ error} - \underbrace{\mathrm{Var}[e(c(\mathbf{Y}))]}_{sharpness} + \underbrace{\mathrm{Var}[v(\mathbf{Y})]}_{uncertainty}$$

BR-rerank: calibrated confidence

How does reranker achieve calibration?

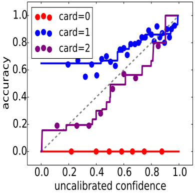
- 0/1 correctness as target and MSE as objective
- output average of targets (=accuracy=calibrated confidence)
- split data: reranker evaluates BR predictions objectively
- use more informative features to increase sharpness

Table: BR prediction calibration performance in terms of MSE (the smaller the better) and sharpness (the bigger the better).

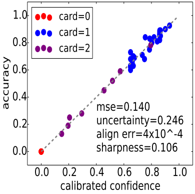
Dataset	uncentainty	uncalib		isotonic		card isotonic		tree		reranker	
		MSE	sharp	MSE	sharp	MSE	sharp	MSE	sharp	MSE	sharp
BIBTEX	0.133	0.193	0.007	0.140	0.002	0.109	0.038	0.086	0.065	**0.068**	**0.072**
OHSUMED	0.232	0.226	0.015	0.221	0.013	**0.182**	**0.051**	0.211	0.039	0.189	0.047
RCV1	0.247	0.175	0.077	0.175	0.075	0.159	0.093	0.134	**0.129**	**0.123**	0.126
TMC	0.212	0.192	0.019	0.192	0.020	0.192	0.022	0.194	0.029	**0.180**	**0.032**
WISE	0.249	0.252	0.017	0.234	0.017	0.151	0.098	0.166	0.093	**0.147**	**0.102**
MSCOCO	0.227	0.158	0.075	0.151	0.075	0.150	0.076	0.163	0.070	**0.143**	**0.083**

Supplementary

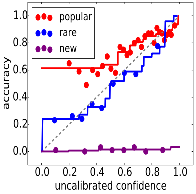
Why does considering more features help calibration?



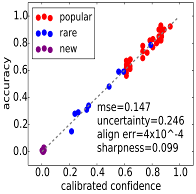
(a)



(b)



(c)



(d)

CBM, BR-rerank vs Deep Learning

- CNN: a feature extractor, can be used in CBM and BR-rerank as base learners for image data
- RNN: original designed for sequence prediction, not for sets, requires a label order in training. We have done work in adapting RNN to set prediction by make RNN training invariant to label orders. Joint work with Kechen Qin. Published at NAACL 2019.
- CBM and BR-rerank: designed for set prediction from the beginning, and do not require label orders.

BR-rerank vs Deep Value Network

- DVN: trains a neural network to evaluate prediction candidates and then uses back-propagation to find the prediction that leads to the maximum score. Only use the binary encoding of the label set. Its gradient based inference makes it very difficult to directly incorporate higher level features extracted from the label set, such as cardinality and prior set probability.
- BR-rerank: could use any feature: binary encoding, BR score, prior, cardinality

BR-rerank vs GAN

- GAN: Also has two models, one for generating samples and one for judging these samples. Unsupervised training, used for generating new samples. Two models trained simultaneously,
- BR-rerank: supervised, used for classification. Two models trained in separate stages.

BR-rerank vs CRF

- CRF: needs to pre-allocate parameters; only model pair-wise interactions; higher-order interaction requires too many parameters. Normalization is intractable; support inference eliminates unseen combinations. There is another CRF that only works for given exclusive or hierarchical label relations.
- BR-rerank: no need to pre-allocate parameters; GB automatically models interactions on the fly using binary labels as features; models higher interactions;

Time Complexity

- reduction methods; depends on the base learner
- dense CBM: $K \times L\times$ binary classifier complexity
- sparse CBM: skip certain label classifiers in each component; sub-linear in K
- BR-rerank: stage 1 BR training dominates

Table: Training time of different methods, measured in seconds. All algorithms run multi-threaded on a server with 56 cores.

Dataset	BIBT	OHSUM	RCV1	TMC	WISE	MSCO
BR	4	3	7	8	80	1380
BR-rerank	9	6	10	11	88	1393
CBM	64	210	70	224	1320	8520
CRF	353	268	1223	771	16363	14760

Time Complexity

Table: The training time and prediction time of different methods on five datasets. All numbers are in seconds.

	dataset	SCENE		RCV1		TMC2007		MEDIAMILL		NUS-WIDE	
Method	Learner	Train	Predict	Train	Predict	Train	Predict	Train	Predict	Train	Predict
BinRel	LR	2	<1	19	<1	26	<1	136	<1	128	1
PowSet	LR	35	<1	3147	<1	38037	1	85794	1	521760	34
CC	LR	3	<1	509	<1	332	<1	1949	1	2520	2
PCC	LR	3	<1	509	3	332	1	1949	4	2520	27
ECC-label	LR	22	<1	4915	27	3404	15	19642	38	25791	246
ECC-subset	LR	22	<1	4915	26	3404	18	19642	39	25791	287
CDN	LR	4	45	18417	213433	54253	596228	3126	6572	17941	41789
pairCRF	linear	11	<1	2136	<1	215	<1	2990	<1	48404	7
(dense) CBM	LR	70	<1	4412	4	1495	1	17608	13	35363	48
(sparse) CBM	LR	24	<1	182	<1	393	<1	8862	5	15561	14

Generating the K-best prediction candidates from BR

Algorithm 1 Generating the K-best prediction candidates from BR

Input: instance $\mathbf{x}$ and a BR classifier
Compute individual label probabilities based on BR: $p_l = p(y_l = 1|\mathbf{x}), l = 1, 2, ..., L$
Initialize an empty priority queue Q^k, and empty list C and an empty label set $\mathbf{y}_{best}$
for $\ell = 1, 2, ..., L$ **do**
  if $p_l > 0.5$ **then**
    add l to $\mathbf{y}_{best}$
  end if
end for
$Q^k.enqueue(\mathbf{y}_{best})$
while $|C| < K$ **do**
  $\mathbf{y} = Q^k.dequeue()$
  add $\mathbf{y}$ to C
  for $\ell = 1, 2, ..., L$ **do**
    Generate $\mathbf{y}'$ by flipping the ℓ-th bit of $\mathbf{y}$
    if $\mathbf{y}'$ has not been added to Q before **then**
      $Q^k.enqueue(\mathbf{y}')$
    end if
  end for
end while
Output: C