Multi-label Text Categorization with Joint Learning Predictions-as-Features Method

Li Li[1] **Baobao Chang**[1] **Shi Zhao**[2] **Lei Sha**[1] **Xu Sun**[1] **Houfeng Wang**[1]
Key Laboratory of Computational Linguistics(Peking University), Ministry of Education, China[1]
Key Laboratory on Machine Perception(Peking University), Ministry of Education, China[2]
{li.l, chbb, shalei, z.s, xusun, wanghf}@pku.edu.cn

EMNLP 2015

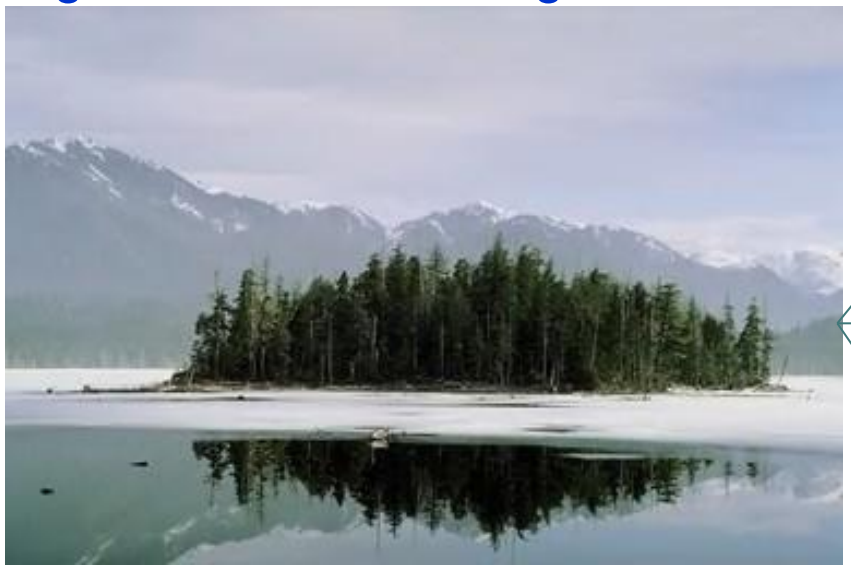
Ning Li

OUTLINE

- Introduction
- CC
- LEAD
- Joint Learning Algorithm
- Experiment
- Conclusion
- Future work

INTRODUCTION

e.g. natural scene image



Lake

Trees

Mountains

Multi-label learning

Shanghai World Expo

economics

volunteers

INTRODUCTION

- Become rather challenging due to the tremendous number of possible label sets

- Exploit correlations between different labels during multi-label learning
  - lion and grassland → Africa
  - entertainment → politics

CC

- Classifier chains for multi-label classification （2011）
  - binary relevance method → BR
  - Classifier Chains model → CC

  each binary model is extended with the 0/1 label relevances of all previous classifiers

Algorithm 1 CC's training phase for training set D and label set $\mathcal{L}$ of L labels

TRAINING($D = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$)

```
1  for j = 1, ..., L
2      do ▷ the jth binary transformation and training
3          D'_j ← {}
4          for (x, y) ∈ D
5              do x' ← [x_1, ..., x_d, y_1, ..., y_{j-1}]
6                  D'_j ← D'_j ∪ (x', y_j)
7          ▷ train h_j to predict binary relevance of y_j
8          h_j : D'_j → {0, 1}
```

CC

- Classifier chains for multi-label classification （2011）
  - Classifier Chains model

Algorithm 2 CC's prediction phase for a test instance $\mathbf{x}$

CLASSIFY($\mathbf{x}$)

1 ▷ global $\mathbf{h} = (h_1, \ldots, h_L)$
2 $\mathbf{y} \leftarrow [\hat{y}_1, \ldots, \hat{y}_L]$
3 **for** $j = 1, \ldots, L$
4 **do** $\mathbf{x}' \leftarrow [x_1, \ldots, x_d, \hat{y}_1, \ldots, \hat{y}_{j-1}]$
5 $\hat{y}_j \leftarrow h_j(\mathbf{x}')$
6 **return** $\hat{\mathbf{y}}$

CC

- Bayes-optimal probabilistic classifier chains(PCC)

$P_{\mathbf{x}}(\mathbf{y}) \equiv P(\mathbf{y}|\mathbf{x}))$ is:

$$P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{x}}(y_1) \cdot \prod_{j=2}^{L} P_{\mathbf{x}}(y_j|y_1, \ldots, y_{j-1})$$

When $h_j(\cdot)$ is a probabilistic classifier, this can be rewritten as:

$$P_{\mathbf{x}}(\mathbf{y}) = h_1(\mathbf{x}) \cdot \prod_{j=2}^{L} h_j(\mathbf{x}, y_1, \ldots, y_{j-1})$$

CC

- Why CC not PCC
  - PCC comes at an intractable computational cost in practice
  - For PCC, this implies an upper limit for L of around 10–15
  - CC needs only consider a single order of the L label variables

LEAD

- Model dependencies between former labels and the current label
  - Multi-Label Learning by Exploiting Label Dependency （KDD 2010）→LEAD
    - Learn the Bayesian network structure G
    - For each label yk, construct the new classifier by incorporating **pa**k implied in the network G into the feature set.

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{k=1}^{q} p(y_k|\mathbf{pa}_k, \boldsymbol{x}), \qquad (1)$$

LEAD

- Multi-Label Learning by Exploiting Label Dependency （KDD 2010）→LEAD

1. Construct the classifiers for all labels independently. This produces the error e_k for each label y_k (Eq. 2).
2. Learn the Bayesian network structure $\mathcal{G}$ of e_k, $1 \leq k \leq q$.
3. For each label y_k, construct the new classifier $\mathcal{C}_k$ by incorporating $\mathbf{pa}_k$ implied in the network $\mathcal{G}$ into the feature set.
4. For testing data, recursively predict y_k with the classifier $\mathcal{C}_k$ and the feature set $\boldsymbol{x} \bigcup \widehat{\mathbf{pa}}_k$ according to the ordering of the labels implied in $\mathcal{G}$.

LEAD

- Step2: Learn the Bayesian network structure

  - BDAGL (Bayesian DAG learning) package
    computing the marginal posterior probability of every edge in a Bayesian network
    O(q*2^q) q is the number of labels
    (q<20)

  - Banjo (Bayesian ANalysis with Java Objects) package
    (q>20)
    maximum a posterior (MAP) structure learning using simulated annealing and hill climbing for searching

JOINT LEARNING ALGORITHM

Draw- back :

Can't model dependencies between the current label and the latter labels

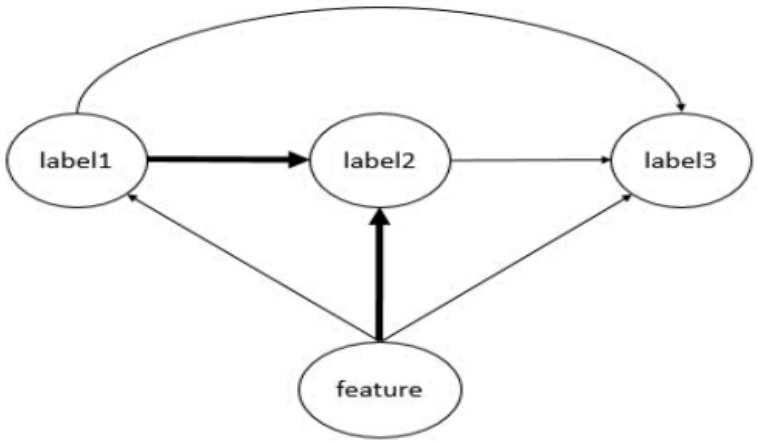


Figure 1: When training the classifier for the second label, the feature (the bold lines) consists of only the origin feature and the prediction for the first label. In this time, it is impossible to model the dependencies between the second label and the third label.

Joint Learning Algorithm

- Preliminaries
  - X denote the document feature space
  - Y = {0, 1}m denote label space with m labels
  - function **h** : X → Y

    h(x) = [h1(x), h2(x), ..., hm(x)]
  - **pa**j denotes the set of parents of the j-th classifiers

$$\mathbf{h}_j : \boldsymbol{x}, h_{k \in \boldsymbol{pa}_j}(\boldsymbol{x}) \rightarrow y_j \qquad (1)$$

Joint Learning Algorithm

- Architecture and Loss
  - pj denotes the probability the document has the j-th label
  - Wj denotes the weight vector of the j-th model
  - [x, pk∈paj] denotes the feature vector x extended with predictions [pk∈paj]

$$
\begin{aligned}
p_j &= \mathbf{h}_j(\boldsymbol{x}, p_{k \in \boldsymbol{pa}_j}) \\
&= \frac{exp([\boldsymbol{x}, p_{k \in \boldsymbol{pa}_j}]^T \boldsymbol{W}_j)}{1 + exp([\boldsymbol{x}, p_{k \in \boldsymbol{pa}_j}]^T \boldsymbol{W}_j)} \qquad (2)
\end{aligned}
$$

Joint Learning Algorithm

- log likelihood losses

$$\mathcal{L}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \sum_{j=1}^{m} \ell(p_j, y_j)$$

$$= -\sum_{j=1}^{m} (y_j log(p_j) + (1 - y_j) log(1 - p_j)) \quad (3)$$

→

$$\boldsymbol{h}^* = \underset{\boldsymbol{h}}{\operatorname{argmin}}\, \mathcal{L}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) \quad (4)$$

Joint Learning Algorithm

- minimizing the global loss function
  - the k-th classifier are updated according to
    - the loss of the k-th classifier
    - the losses of the latter classifiers.
  - take predictions by the former classifiers to extend the latter classifiers' features （CC and LEAD）

Joint Learning Algorithm

- classification models
  - logistic regression
  - L2 SVM

- minimize the global loss function
  - the Back propagation Through Structure (BTS)

  (Goller and Kuchler, 1996)

EXPERIMENTS

- Datasets

  n : the size of the entire data set,

  d : the number of the bag-of-words features,

  m : the number of labels.

dataset	n	d	m
slashdot	3782	1079	22
medical	978	1449	45
enron	1702	1001	53
tmc2007	28596	500	22

Table 2: Multi-label data sets and associated statistics.

EXPERIMENTS

- Evaluation Metrics
  - Percentage of the wrong labels to the total labels

$$Hammingloss = \frac{1}{m}|\mathbf{h}(\boldsymbol{x})\Delta\boldsymbol{y}| \qquad (5)$$

  - h(x) to match the true set of labels S *exactly*

$$0/1loss = I(\mathbf{h}(\boldsymbol{x}) \neq \boldsymbol{y}) \qquad (6)$$

  - F score is a harmonic mean between precision and recall

$$Fscore = \frac{1}{m}\sum_{i=j}^{m}\frac{2 * p_j * r_j}{p_j + r_j} \qquad (7)$$

Experiments

- Method Setup (logistic regression)
  - Baseline
    - BR
    - LEAD
    - CC
  - Our methods
    - JCC
    - JLEAD

EXPERIMENTS

Dataset	BR	CC	LEAD	JCC	JLEAD
hamming loss (lower is better)					
slashdot	0.046 ± 0.002	0.043 ± 0.001	$0.045 \pm 0.001\circ$	0.043 ± 0.001	0.043 ± 0.001
medical	0.013 ± 0.001	$0.013 \pm 0.001\bullet$	$0.012 \pm 0.000\circ$	0.011 ± 0.000	0.010 ± 0.001
enron	0.052 ± 0.001	$0.053 \pm 0.002\bullet$	$0.052 \pm 0.001\circ$	0.049 ± 0.001	0.049 ± 0.001
tmc2007	0.063 ± 0.002	0.058 ± 0.001	0.058 ± 0.001	0.057 ± 0.001	0.057 ± 0.001
0/1 loss (lower is better)					
slashdot	0.645 ± 0.013	$0.637 \pm 0.015\bullet$	$0.631 \pm 0.017\circ$	0.610 ± 0.014	0.614 ± 0.011
medical	0.398 ± 0.034	$0.377 \pm 0.032\bullet$	$0.379 \pm 0.033\circ$	0.353 ± 0.030	0.345 ± 0.030
enron	0.856 ± 0.016	0.848 ± 0.017	0.853 ± 0.017	0.848 ± 0.018	0.850 ± 0.017
tmc2007	0.698 ± 0.004	0.686 ± 0.006	0.689 ± 0.009	0.684 ± 0.006	0.681 ± 0.006
F score (higher is better)					
slashdot	0.345 ± 0.016	$0.354 \pm 0.015\bullet$	$0.364 \pm 0.015\circ$	0.385 ± 0.017	0.383 ± 0.017
medical	0.403 ± 0.012	$0.416 \pm 0.013\bullet$	$0.426 \pm 0.011\circ$	0.444 ± 0.009	0.446 ± 0.013
enron	0.222 ± 0.014	0.224 ± 0.019	0.225 ± 0.018	0.223 ± 0.017	0.222 ± 0.015
tmc2007	0.524 ± 0.007	$0.531 \pm 0.009\bullet$	$0.508 \pm 0.017\circ$	0.547 ± 0.007	0.546 ± 0.006

Table 1: Performance (mean±std.) of each approach in terms of different evaluation metrics. $\bullet/\circ$ indicates whether JCC/JLEAD is statistically superior to CC/LEAD respectively (pairwise t-test at 5% significance level).

EXPERIMENTS

Criteria	JCC against CC	JLEAD against LEAD
hamming loss	**2/2/0**	**3/1/0**
0/1 loss	**2/2/0**	**2/2/0**
F-score	**3/1/0**	**3/1/0**
Total	**7/5/0**	**8/4/0**

Table 3: The win/tie/loss results for the joint learning algorithm against the original predictions-as-features methods in terms of different evaluation metrics (pairwise t-test at 5% significance level).

EXPERIMENTS

The training time

Dataset	CC	JCC	LEAD	JLEAD
slashdot	63.85	85.63	52.17	73.85
medical	134.11	142.51	115.33	128.78
enron	234.28	257.89	196.87	218.95
tmc2007	153.70	169.52	145.80	158.56

Table 4: The average training time (in seconds) of each approach

CONCLUSION

- CC and LEAD suffer from the drawback that neglects dependencies between current label and the latter labels.
- Joint learning algorithm that allows the feedbacks to be propagated from the latter classifiers to the current classifier.
- Our experiments illustrate the models trained by our algorithm outperform the original models.

Future work

- CC:
  - One of the first classifiers predict poorly → effect of error propagation along the chain
  - Solutions: with several random label order (ECC)
- LEAD
  - Explore better way to encode the conditional dependencies of the labels with the feature set as the common parents

Thank You

For Listening