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

Abstract

Multi-label text categorization is a type of text categorization, where each document is assigned to one or more categories. Recently, a series of methods have been developed, which train a classifier for each label, organize the classifiers in a partially ordered structure and take predictions produced by the former classifiers as the latter classifiers' features. These predictions-as-features style methods model high order label dependencies and obtain high performance. Nevertheless, the predictions-as-features methods suffer a drawback. When training a classifier for one label, the predictions-as-features methods can model dependencies between former labels and the current label, but they can't model dependencies between the current label and the latter labels. To address this problem, we propose a novel joint learning algorithm that allows the feedbacks to be propagated from the classifiers for latter labels to the classifier for the current label. We conduct experiments using real-world textual data sets, and these experiments illustrate the predictions-as-features models trained by our algorithm outperform the original models.

1 Introduction

The multi-label text categorization is a type of text categorization, where each document is assigned to one or more categories simultaneously. The multi-label setting is common and useful in the real world. For example, in the news categorization task, a newspaper article concerning global warming can be classified into two categories simultaneously, namely environment and science. For another example, in the task of classifying music lyrics into emotions, a song's lyrics can deliver happiness and excitement simultaneously. The research about the multi-label text categorization attracts increasing attention (Srivastava and Zane-Ulman, 2005; Katakis et al., 2008; Rubin et al., 2012; Nam et al., 2013; Li et al., 2014).

Recently, a series of predictions-as-features style methods have been developed, which train a classifier for each label, organize the classifiers in a partially ordered structure and take predictions produced by the former classifiers as the latter classifiers' features. These predictions-as-features style methods model high order label dependencies (Zhang and Zhang, 2010) and obtain high performance. *Classifier chain* (CC) (Read et al., 2011) and *multi-label Learning by Exploiting lAbel Dependency* (Lead) (Zhang and Zhang, 2010) are two famous predictions-as-features methods. CC organizes classifiers along a chain and LEAD organizes classifiers in a Bayesian network. Besides, there are other works on extending the predictions-as-features methods (Zaragoza et al., 2011; Gonçalves et al., 2013; Sucar et al., 2014). In this paper, we focus on the predictions-as-features style methods.

The previous works of the predictions-as-features methods focus on learning the partially ordered structure. They neglect a drawback. When training a classifier for one label, predictions-as-features methods can model dependencies between former labels and the current label, but they can't model dependencies between the current label and the latter labels. Consider the case of three labels. We organize classifiers in a partially ordered structure shown in figure 1. When training the classifier for the second label, the feature (the bold lines in figure) consists of the origin feature and the prediction for the first label. The information about the third label can't be incorporated. It means that we only model the dependencies between the first label and the sec-

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 835–839,
Lisbon, Portugal, 17-21 September 2015.

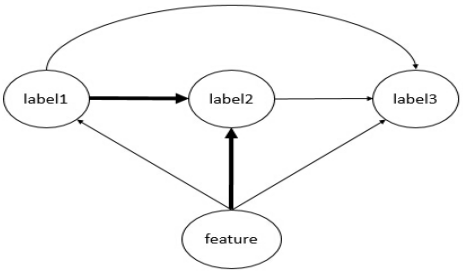

Figure 1: When training the classifier for the second label, the feature (the bold lines) consists of only the origin feature and the prediction for the first label. In this time, it is impossible to model the dependencies between the second label and the third label.

ond label and that the dependencies between the second label and the third label is missing.

To address this problem, we propose a novel joint learning algorithm that allows the feedbacks to be propagated from the classifiers for latter labels to the classifier for the current label, so that the information about the latter labels can be incorporated. It means that the proposed method can model, not only the dependencies between former labels and current label as the usual predictions-as-features methods, but also the dependencies between current label and latter labels. With not missing dependencies. Hence, the proposed method will improve the performance. Our experiments illustrate the models trained by our algorithm outperform the original models. You can find the code of this paper online [1].

The rest of this paper is organized as follows. Section 2 presents the proposed method. We conduct experiments to demonstrate the effectiveness of the proposed method in section 3. Section 4 concludes this paper.

2 Joint Learning Algorithm

2.1 Preliminaries

Let $\mathcal{X}$ denote the document feature space, and $\mathcal{Y} = \{0,1\}^m$ denote label space with m labels. A document instance $\boldsymbol{x} \in \mathcal{X}$ is associated with a label vector $\boldsymbol{y} = (y_1, y_2, ..., y_m)$, where $y_i = 1$ denotes the document has the i-th label and 0 otherwise. The goal of multi-label learning is to learn a function $\mathbf{h} : \mathcal{X} \rightarrow \mathcal{Y}$. In general, $\mathbf{h}$ consists of m functions, one for a label, i.e., $\mathbf{h}(\boldsymbol{x}) = [\mathbf{h}_1(\boldsymbol{x}), \mathbf{h}_2(\boldsymbol{x}), ..., \mathbf{h}_m(\boldsymbol{x})]$.

In the predictions-as-features methods, the classifiers are organized in a partially ordered structure and take predictions produced by the former classifiers as features. We can describe the classifier in the predictions-as-features method as follows.

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

where $\boldsymbol{pa_j}$ denotes the set of parents of the j-th classifiers in the partially ordered structure.

2.2 Architecture and Loss

In this subsection, we introduce architecture and loss function of our joint learning algorithm. As a motivating example, we employ logistic regression as the base classifier in the predictions-as-features methods. The classification function is the sigmoid function, as shown in Eq.(2).

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

where p_j denotes the probability the document has the j-th label, $\boldsymbol{W_j}$ denotes the weight vector of the j-th model and $[\boldsymbol{x}, p_{k\in \boldsymbol{pa_j}}]$ denotes the feature vector x extended with predictions $[p_{k\in \boldsymbol{pa_j}}]$ produced by the former classifiers.

The joint algorithm learns classifiers in the partially ordered structure jointly by minimizing a global loss function. We use the sum of negative log likelihood losses of all classifiers as the global loss function.

$$\begin{aligned} \mathcal{L}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) &= \sum_{j=1}^{m} \ell(p_j, y_j) \\ &= -\sum_{j=1}^{m} (y_j log(p_j) + (1 - y_j) log(1 - p_j)) \end{aligned} \tag{3}$$

The joint algorithm minimizes this global loss function, as Eq.(4) shows.

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

Minimizing this global loss function is *inequivalent* to minimizing the loss function of each base classifier separately, since minimizing the global

[1] https://github.com/rustle1314/Joint_Learning_Predictions_as_Features_for_Multi_Label_Classification

loss function results in feedbacks from latter classifiers. In the predictions-as-features methods, the weights of the k-th classifier are the factors of not only the k-th classifier but also the latter classifiers. Consequently, when minimizing the global loss function, the weights of the k-th classifier are updated according to not only the loss of the k-th classifier but also the losses of the latter classifiers. In other words, feedbacks are propagated from the latter classifiers to the k-th classifier.

The predictions-as-features models trained by our proposed joint learning algorithm can model the dependencies between former labels and current label, since they take predictions by the former classifiers to extend the latter classifiers' features, as the usual predictions-as-features methods do. Besides, they can also model the dependencies between current label and latter labels due to the feedbacks incorporated by the joint learning algorithm.

Here, we employ logistic regression as the motivating example. If we want to employ other classification models, we use other classification function and other loss function. For example, if we want to employ L2 SVM as base classifiers, we resort to the linear classification function and the L2 hinge loss function.

We employ the Back propagation Through Structure (BTS) (Goller and Kuchler, 1996) to minimize the global loss function. In BTS, parent node is computed with its child nodes at the forward pass stage; child node receives gradient as the sum of derivatives from all its parents.

3 Experiments

3.1 Datasets

We perform experiments on four real world data sets: 1) the first data set is Slashdot (Read et al., 2011). The Slashdot data set is concerned about predicting multiple labels given science and technology news titles and partial blurbs mined from Slashdot.org. 2) the second data set is Medical (Pestian et al., 2007). This data set involves the assignment of ICD-9-CM codes to radiology reports. 3) The third data set is Enron. The enron data set is a subset of the Enron Email Dataset, as labelled by the UC Berkeley Enron Email Analysis Project[2]. It is concerned about classifying emails into some categories. 4) the fourth data set is Tmc2007 (Srivastava and Zane-Ulman, 2005). It is concerned about safety report categorization, which is to label aviation safety reports with respect to what types of problems they describe.

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.

Table 2 shows these multi-label data sets and associated statistics. n denotes the size of the entire data set, d denotes the number of the bag-of-words features, m denotes the number of labels. These data sets are available online [3].

3.2 Evaluation Metrics

We use three common used evaluation metrics. The Hamming loss is defined as the percentage of the wrong labels to the total number of labels.

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

where Δ denotes the symmetric difference of two sets, equivalent to XOR operator in Boolean logic.

The multi-label 0/1 loss is the exact match measure as it requires any predicted set of labels $\mathbf{h}(\boldsymbol{x})$ to match the true set of labels S *exactly*. The 0/1 loss is defined as follows:

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

Let p_j and r_j denote the precision and recall for the j-th label. The macro-averaged F score is a harmonic mean between precision and recall, defined as follows:

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

3.3 Method Setup

In this paper, we focus on the predictions-as-features style methods, and use CC and LEAD as the baselines. Our methods are JCC and JLEAD. JCC(JLEAD) is CC(LEAD) trained by our joint algorithm and we compare JCC(JLEAD) to CC(LEAD) respectively. Put it another way, CC/LEAD provide the partial order structure of

[2] http://bailando.sims.berkeley.edu/enron_email.html

[3] http://mulan.sourceforge.net/datasets.html and http://mlkd.csd.auth.gr/multilabel.html

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$\pm$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).

classifiers, and train these classifiers one by one. JCC/LEAD train classifiers jointly in the partial order structure provided by CC/LEAD.

For LEAD and JLEAD, we use the Banjo (Bayesian ANalysis with Java Objects) (Smith et al., 2006) package as the Bayesian structure learning tool. Besides, we also perform experiments with Binary Relevance (BR), which is the baseline for the predictions-as-features methods. BR trains a classifier for a label independently, which doesn't model dependencies. The base classifier of all of them is set to logistic regression without regularization. Experiments are performed in ten-fold cross validation with pairwise t-test at 5% significance level.

3.4 Performance

We reports the detailed results in terms of different evaluation metrics on different data sets in table 1. As shown in this figures, CC and LEAD outperform BR, which shows the values of the prediction-as-features methods. JCC and JLEAD wins over CC and LEAD respectively, which shows the values of the proposed joint learning algorithm.

The improvements are much smaller on the Enron data set than other data sets. In fact, BR, the original prediction-as-features methods and our proposed methods share similar performance on the Enron data set. The reason may be that the label dependencies in the Enron dataset is weak. The label dependencies weakness can be validated by the fact that the modeling-correlation CC and LEAD can't obtain much higher performance than the not-modeling-correlation BR. Due to the weak label dependencies, the modeling-correlation-better JCC(JLEAD) can't obtain much higher performance than CC(LEAD).

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).

We summarize the detailed results into Table 3. JCC is significantly superior to CC in 7/12 cases, tie in 5/12 cases, inferior in zero case. JLEAD is significantly superior to LEAD in 8/12 cases, tie in 4/12 cases, inferior in zero case. The results indicates that our proposed joint algorithm can improve the performance of the predictions-as-features methods.

3.5 Time

The training time (mean) of each approach is showed detailed in table 4. First, we find the training time is related to the number of labels. The training time on the Tmc2007 dataset (28596 instances, 500 features and 22 labels) is less than that on the Enron dataset (1702 instances, 1001 features and 53 labels). This is very easy to understand. We train more classifiers with respect to more labels, which leads to more training time. Second, LEAD/JLEAD have slightly less training time than CC/JCC. The Bayesian network struc-

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

ture learning tool limits that a node has five parent nodes at most. Hence, the partially order structure of LEAD/JLEAD is much simpler. Third, the training time of the joint algorithm is slightly more than that of the original methods. Some time is spent on back-propagating feedbacks from latter classifiers.

4 Conclusion

The multi-label text categorization is a common and useful text categorization. Recently, a series of predictions-as-features style methods have been developed, which model high order label dependencies and obtain high performance. The predictions-as-features methods suffer from the drawback that they methods can't model dependencies between current label and the latter labels. To address this problem, we propose a novel 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.

5 Acknowledge

We sincerely thank all the anonymous reviewers for their valuable comments, which have helped to improve this paper greatly. Our work is supported by National High Technology Research and Development Program of China (863 Program) (No. 2015AA015402), National Natural Science Foundation of China(No.61370117 & No.61433015)and Major National Social Science Fund of China(No.12 & ZD227).

References

Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 347–352. IEEE.

Eduardo Corrêa Gonçalves, Alexandre Plastino, and Alex A Freitas. 2013. A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 469–476. IEEE.

Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2008. Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD*.

Li Li, Longkai Zhang, and Houfeng Wang. 2014. Muli-label text categorization with hidden components. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1816–1821, Doha, Qatar, October. Association for Computational Linguistics.

Jinseok Nam, Jungi Kim, Iryna Gurevych, and Johannes Fürnkranz. 2013. Large-scale multi-label text classification-revisiting neural networks. *arXiv preprint arXiv:1312.5419*.

John P Pestian, Christopher Brew, Paweł Matykiewicz, DJ Hovermale, Neil Johnson, K Bretonnel Cohen, and Włodzisław Duch. 2007. A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 97–104. Association for Computational Linguistics.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359.

Timothy N Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. 2012. Statistical topic models for multi-label document classification. *Machine Learning*, 88(1-2):157–208.

V Anne Smith, Jing Yu, Tom V Smulders, Alexander J Hartemink, and Erich D Jarvis. 2006. Computational inference of neural information flow networks. *PLoS computational biology*, 2(11):e161.

Ashok N Srivastava and Brett Zane-Ulman. 2005. Discovering recurring anomalies in text reports regarding complex space systems. In *Aerospace Conference, 2005 IEEE*, pages 3853–3862. IEEE.

L Enrique Sucar, Concha Bielza, Eduardo F Morales, Pablo Hernandez-Leal, Julio H Zaragoza, and Pedro Larrañaga. 2014. Multi-label classification with bayesian network-based chain classifiers. *Pattern Recognition Letters*, 41:14–22.

Julio H Zaragoza, Luis Enrique Sucar, Eduardo F Morales, Concha Bielza, and Pedro Larranaga. 2011. Bayesian chain classifiers for multidimensional classification. In *IJCAI*, volume 11, pages 2192–2197. Citeseer.

Min-Ling Zhang and Kun Zhang. 2010. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 999–1008. ACM.