Muli-label Text Categorization with Hidden Components

Li Li   Longkai Zhang   Houfeng Wang
Key Laboratory of Computational Linguistics (Peking University) Ministry of Education, China
li.l@pku.edu.cn, zhlongk@qq.com, wanghf@pku.edu.cn

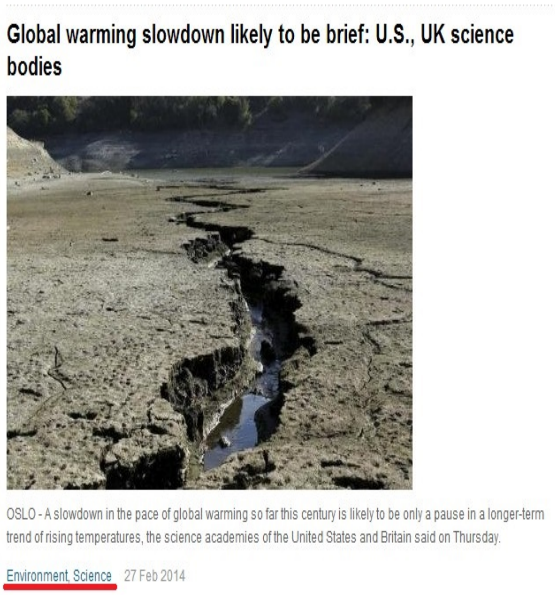
Abstract

Multi-label text categorization (MTC) is supervised learning, where a document may be assigned with multiple categories (labels) simultaneously. The labels in the MTC are correlated and the correlation results in some hidden components, which represent the ”share” variance of correlated labels. In this paper, we propose a method with hidden components for MTC. The proposed method employs PCA to capture the hidden components, and incorporates them into a joint learning framework to improve the performance. Experiments with real-world data sets and evaluation metrics validate the effectiveness of the proposed method.

1 Introduction

Many real-world text categorization applications are multi-label text categorization (Srivastava and Zane-Ulman, 2005; Katakis et al., 2008; Rubin et al., 2012; Nam et al., 2013), where a documents is usually assigned with *multiple* labels simultaneously. For example, as figure 1 shows, a newspaper article concerning global warming can be classified into two categories, *Environment*, and *Science* simultaneously. Let $\mathcal{X} = R^d$ be the documents corpus, and $\mathcal{Y} = \{0, 1\}^m$ be the label space with m labels. We denote by $\{(\boldsymbol{x_1}, \boldsymbol{y_1}), (\boldsymbol{x_2}, \boldsymbol{y_2}), ..., (\boldsymbol{x_n}, \boldsymbol{y_n})\}$ the training set of n documents. Each document is denoted by a vector $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}, ..., x_{i,d}]$ of d dimensions. The labeling of the i-th document is denoted by vector $\boldsymbol{y}_i = [y_{i,1}, y_{i,2}, ..., y_{i,m}]$, where y_{il} is 1 when the i-th document has the l-th label and 0 otherwise. The goal is to learn a function $\boldsymbol{f} : \mathcal{X} \rightarrow \mathcal{Y}$. Generally, we can assume **f** consists of m functions, one for a label.

$$\boldsymbol{f} = [f_1, f_2, ..., f_m]$$



Global warming slowdown likely to be brief: U.S., UK science bodies

OSLO - A slowdown in the pace of global warming so far this century is likely to be only a pause in a longer-term trend of rising temperatures, the science academies of the United States and Britain said on Thursday.

Environment, Science 27 Feb 2014

Figure 1: A newspaper article concerning global warming can be classified into two categories, *Environment*, and *Science*.

The labels in the MLC are correlated. For example, a ”politics” document is likely to be an ”economic” document simultaneously, but likely not to be a ”literature” document. According to the latent variable model (Tabachnick et al., 2001), the labels with correlation result in some hidden components, which represent the ”share” variance of correlated labels. Intuitively, if we can capture and utilize these hidden components in MTC, the performance will be improved. To implement this idea, we propose a multi- label text categorization method with hidden components, which employ PCA to capture the hidden components, and then incorporates these hidden components into a joint learning framework. Experiments with various data sets and evaluation metrics validate the values of our method. The research close to our work is ML-LOC (Multi-Label learning using LOcal Correlation) in (Huang and Zhou, 2012). The differ-

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1816–1821,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

ences between ours and ML-LOC is that ML-LOC employs the cluster method to gain the local correlation, but we employ the PCA to obtain the hidden code. Meanwhile, ML-LOC uses the linear programming in learning the local code, but we employ the gradient descent method since we add non-linear function to the hidden code.

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 Methodology

2.1 Capturing Hidden Component via Principle Component Analysis

The first step of the proposed method is to capture hidden components of training instances. Here we employ Principal component analysis (PCA). This is because PCA is a well-known statistical tool that converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principle components. These principle components represent the inner structure of the correlated variables.

In this paper, we directly employ PCA to convert labels of training instances into their principle components, and take these principle components as hidden components of training instances. We denote by $\boldsymbol{h}_i$ the hidden components of the i-th instance captured by PCA.

2.2 Joint Learning Framework

We expand the original feature representation of the instance $\boldsymbol{x}_i$ by its hidden component code vector $\boldsymbol{c}_i$. For simplicity, we use logistic regression as the motivating example. Let $\boldsymbol{w}_l$ denote weights in the l-th function f_l, consisting of two parts: 1) $\boldsymbol{w}_l^x$ is the part involving the instance features. 2) $\boldsymbol{w}_l^c$ is the part involving the hidden component codes. Hence f_l is:

$$f_l(\boldsymbol{x},\boldsymbol{c}) = \frac{1}{1+\exp(-\boldsymbol{x}^T\boldsymbol{w}_l^x - \boldsymbol{c}^T\boldsymbol{w}_l^c)} \quad (1)$$

where $\boldsymbol{C}$ is the code vectors set of all training instances.

The natural choice of the code vector $\boldsymbol{c}$ is $\boldsymbol{h}$. However, when testing an instance, the labeling is unknown (exactly what we try to predict), consequently we can not capture $\boldsymbol{h}$ with PCA to replace the code vector $\boldsymbol{c}$ in the prediction function Eq.(1). Therefore, we assume a linear transformation $\boldsymbol{M}$ from the training instances to their independent components, and use $\boldsymbol{Mx}$ as the approximate independent component. For numerical stability, we add a non-linear function (e.g., the tanh function) to $\boldsymbol{Mx}$. This is formulated as follows.

$$\boldsymbol{c} = tanh(\boldsymbol{Mx}) \quad (2)$$

Aiming to the discrimination fitting and the independent components encoding, we optimize the following optimization problem.

$$\min_{\boldsymbol{W},C} \sum_{i=1}^{n}\sum_{l=1}^{m} \ell(\boldsymbol{x}_i,\boldsymbol{c}_i,y_{il},f_l) + \lambda_1\Omega(\boldsymbol{f}) + \lambda_2 Z(\boldsymbol{C}) \quad (3)$$

The first term of Eq.(3) is the loss function. ℓ is the loss function defined on the training data, and $\boldsymbol{W}$ denotes all weights in the our model, i.e., $\boldsymbol{w}_1,...,\boldsymbol{w}_l,...,\boldsymbol{w}_m$. Since we utilize the logistic regression in our model, the loss function is defined as follows.

$$\begin{aligned}&\ell(\boldsymbol{x},\boldsymbol{c},y,f)\\ =\ & -yln f(\boldsymbol{x},\boldsymbol{c}) - (1-y)ln(1-f(\boldsymbol{x},\boldsymbol{c}))\end{aligned} \quad (4)$$

The second term of Eq.(3) Ω is to punish the model complexity, which we use the ℓ_2 regularization term.

$$\Omega(\boldsymbol{f}) = \sum_{l=1}^{m} ||\boldsymbol{w}_l||^2. \quad (5)$$

The third term of Eq.(3) Z is to enforce the code vector close to the independent component vector. To obtain the goal, we use the least square error between the code vector and the independent component vector as the third regularized term.

$$Z(C) = \sum_{i=1}^{n} ||\boldsymbol{c}_i - \boldsymbol{h}_i||^2. \quad (6)$$

By substituting the Eq.(5) and Eq.(6) into Eq.(3) and changing $\boldsymbol{c}$ to $tanh(\boldsymbol{Mx})$ (Eq.(2)), we obtain the following optimization problem.

$$\min_{\boldsymbol{W},\boldsymbol{M}} \sum_{i=1}^{n}\sum_{l=1}^{m} \ell(\boldsymbol{x}_i, tanh(\boldsymbol{Mx}_i), y_{il}, \boldsymbol{f}) + \lambda_1 \sum_{l=1}^{m} ||\boldsymbol{w}_l||^2 + \lambda_2 \sum_{i=1}^{n} ||\boldsymbol{Mx}_i - \boldsymbol{h_i}||^2 \quad (7)$$

2.3 Alternative Optimization method

We solve the optimization problem in Eq.(7) by the alternative optimization method, which optimize one group of the two parameters with the other fixed. When the $\boldsymbol{M}$ fixed, the third term of Eq.(7) is a constant and thus can be ignored, then Eq.(7) can be rewritten as follows.

$$\min_{\boldsymbol{W}} \sum_{i=1}^{n} \sum_{l=1}^{m} \ell(\boldsymbol{x}_i, tanh(\boldsymbol{M}\boldsymbol{x}_i), y_{il}, f_l) + \lambda_1 \sum_{l=1}^{m} ||\boldsymbol{w}_l||^2 \quad (8)$$

By decomposing Eq.(8) based on the label, the equation Eq.(8) can be simplified to:

$$\min_{\boldsymbol{w}_l} \sum_{i=1}^{n} \ell(\boldsymbol{x}_i, tanh(\boldsymbol{M}\boldsymbol{x}_i), y_{il}, f_l) + \lambda_1 ||\boldsymbol{w}_l||^2 \quad (9)$$

Eq.(9) is the standard logistic regression, which has many efficient optimization algorithms.

When $\boldsymbol{W}$ fixed, the second term is constant and can be omitted, then Ep.(7) can rewritten to Eq.(10). We can apply the gradient descent method to optimize this problem.

$$\min_{\boldsymbol{M}} \sum_{i=1}^{n} \sum_{l=1}^{m} \ell(\boldsymbol{x}_i, tanh(\boldsymbol{M}\boldsymbol{x}_i), y_{il}, f_l) + \lambda_2 \sum_{i=1}^{n} ||\boldsymbol{M}\boldsymbol{x}_i - \boldsymbol{h_i}||^2 \quad (10)$$

3 Experiments

3.1 Evaluation Metrics

Compared with the single-label classification, the multi-label setting introduces the additional degrees of freedom, so that various multi-label evaluation metrics are requisite. We use three different multi-label evaluation metrics, include the hamming loss evaluation metric.

The hamming loss is defined as the percentage of the wrong labels to the total number of labels.

$$Hammingloss = \frac{1}{m}|\mathbf{h}(\boldsymbol{x})\Delta\boldsymbol{y}| \quad (11)$$

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

The multi-label 0/1 loss, also known as subset accuracy, 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:

$$0/1loss = I(\mathbf{h}(\boldsymbol{x}) \neq \boldsymbol{y}) \quad (12)$$

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

$$F = \frac{1}{m} \sum_{i=j}^{m} \frac{2 * a_j * r_j}{a_j + r_j} \quad (13)$$

3.2 Datasets

We perform experiments on three MTC data sets: 1) the first data set is slashdot (Read et al., 2011). The slashdot data set is concerned about science and technology news categorization, which predicts multiply labels given article 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 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. The characteristics of them are shown in Table 1, where n denotes the size of the data set, d denotes the dimension of the document instance, and m denotes the number of labels.

dataset	n	d	m	Lcard
slashdot	3782	1079	22	1.18
medical	978	1449	45	1.245
tmc2007	28596	500	22	2.16

Table 1: Multi-label data sets and associated statistics

The measure label cardinality $Lcard$, which is one of the standard measures of ”multi-label-ness”, defined as follows, introduced in (Tsoumakas and Katakis, 2007).

$$Lcard(D) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} y_j^i}{n}$$

where D denotes the data set, l_j^i denotes the j-th label of the i-th instance in the data set.

3.3 Compared to Baselines

To examine the values of the joint learning framework, we compare our method to two baselines. The baseline 1 eliminates the PCA, which just adds an extra set of non-linear features. To implement this baseline, we only need to set $\lambda_2 = 0$. The baseline 2 eliminates the joint learning framework. This baseline captures the hidden component codes with PCA, trains a linear regression model to fit the hidden component codes, and utilizes the outputs of the linear regression model as features.

For the proposed method, we set $\lambda_1 = 0.001$ and $\lambda_2 = 0.1$. For the baseline 2, we employ logistic regression with 0.001 $\ell 2$ regularization as the base classifier. Evaluations are done in tenfold cross validation. Note that all of them produce real-valued predictions. A threshold t needs to be used to determine the final multi-label set $\boldsymbol{y}$ such that $l_j \in \boldsymbol{y}$ where $p_j \geq t$. We select threshold t, which makes the $Lcard$ measure of predictions for the training set is closest to the $Lcard$ measure of the training set (Read et al., 2011). The threshold t is determined as follows, where D_t is the training set and a multi-label model H_t predicts for the training set under threshold t.

$$t = \operatorname*{argmin}_{t \in [0,1]} |Lcard(D_t) - Lcard(H_t(D_t))| \quad (14)$$

Table 2 reports our method wins over the baselines in terms of different evaluation metrics, which shows the values of PCA and our joint learning framework. The hidden component code only fits the hidden component in the baseline method. The hidden component code obtains balance of fitting hidden component and fitting the training data in the joint learning framework.

3.4 Compared to Other Methods

We compare the proposed method to BR, CC (Read et al., 2011), RAKEL (Tsoumakas and Vlahavas, 2007) and ML-KNN (Zhang and Zhou, 2007). entropy. ML-kNN is an adaption of kNN algorithm for multilabel classification. methods. Binary Revelance (BR) is a simple but effective method that trains binary classifiers for each label independently. BR has a low time complexity but makes an arbitrary assumption that the labels are independent from each other. CC organizes the classifiers along a chain and take predictions produced by the former classifiers as features of the latter classifiers. ML-kNN uses kNN algorithms independently for each label with considering prior probabilities. The Label Powerset (LP) method models independencies among labels by treating each label combination as a new class. LP consumes too much time, since there are 2^m label combinations with m labels. RAndom K labEL (RAKEL) is an ensemble method of LP. RAKEL learns several LP models with random subsets of size k from all labels, and then uses a vote process to determine the final predictions.

For our proposed method, we employ the setup in subsection 3.3. We utilize logistic regression with 0.001 $\ell 2$ regularization as the base classifier for BR, CC and RAKEL. For RAKEL, the number of ensemble is set to the number of label and the size of the label subset is set to 3. For MLKNN, the number of neighbors used in the k-nearest neighbor algorithm is set to 10 and the smooth parameter is set to 1. Evaluations are done in tenfold cross validation. We employ the threshold-selection strategy introduced in subsection 3.3

Table 2 also reports the detailed results in terms of different evaluation metrics. The mean metric value and the standard deviation of each method are listed for each data set. We see our proposed method shows majorities of wining over the other state-of-the-art methods nearly at all data sets under hamming loss, 0/1 loss and macro f score. Especially, under the macro f score, the advantages of our proposed method over the other methods are very clear.

4 CONCLUSION

Many real-world text categorization applications are multi-label text categorization (MTC), where a documents is usually assigned with *multiple* labels simultaneously. The key challenge of MTC is the label correlations among labels. In this paper, we propose a MTC method via hidden components to capture the label correlations. The proposed method obtains hidden components via PCA and incorporates them into a joint learning framework. Experiments with various data sets and evaluation metrics validate the effectiveness of the proposed method.

Acknowledge

We thank anonymous reviewers for their helpful comments and suggestions. This research was partly supported by National High Tech-

	hamming↓. Lower is better.		
Dataset	slashdot	medical	tmc2007
Proposed	0.044 ± 0.004	0.010 ± 0.002	0.056 ± 0.002
Baseline1	0.046 ± 0.003●	0.010 ± 0.002	0.056 ± 0.001
Baseline2	0.047 ± 0.003●	0.011 ± 0.001	0.059 ± 0.001●
BR	0.058 ± 0.003●	0.010 ± 0.001	0.060 ± 0.001●
CC	0.049 ± 0.003●	0.010 ± 0.001	0.058 ± 0.001●
RAKEL	0.039 ± 0.002○	0.011 ± 0.002	0.057 ± 0.001
MLKNN	0.067 ± 0.003●	0.016 ± 0.003●	0.070 ± 0.002●
	0/1 loss↓. Lower is better.		
Dataset	slashdot	medical	tmc2007
Proposed	0.600 ± 0.042	0.316 ± 0.071	0.672 ± 0.010
Baseline1	0.615 ± 0.034●	0.324 ± 0.058●	0.672 ± 0.008
Baseline2	0.669 ± 0.039●	0.354 ± 0.062●	0.698 ± 0.007●
BR	0.803 ± 0.018●	0.337 ± 0.063●	0.701 ± 0.008●
CC	0.657 ± 0.025●	0.337 ± 0.064●	0.687 ± 0.010●
RAKEL	0.686 ± 0.024●	0.363 ± 0.064●	0.682 ± 0.009●
MLKNN	0.776 ± 0.020●	0.491 ± 0.083●	0.746 ± 0.003●
	F score↑. Larger is better.		
Dataset	slashdot	medical	tmc2007
Proposed	0.429 ± 0.026	0.575 ± 0.067	0.587 ± 0.010
Baseline1	0.413 ± 0.032●	0.547 ± 0.056●	0.577 ± 0.011
Baseline2	0.398 ± 0.032●	0.561 ± 0.052●	0.506 ± 0.011●
BR	0.204 ± 0.011●	0.501 ± 0.058●	0.453 ± 0.011●
CC	0.303 ± 0.022●	0.510 ± 0.052●	0.505 ± 0.011●
RAKEL	0.349 ± 0.023●	0.589 ± 0.063○	0.555 ± 0.011●
MLKNN	0.297 ± 0.031●	0.410 ± 0.064●	0.431 ± 0.014●

Table 2: Performance (mean±std.) of our method and baseline in terms of different evaluation metrics. ●/○ indicates whether the proposed method is statistically superior/inferior to baseline (pairwise t-test at 5% significance level).

nology Research and Development Program of China (863 Program) (No.2012AA011101), National Natural Science Foundation of China (No.91024009), Major National Social Science Fund of China (No. 12&ZD227). The contact author of this paper, according to the meaning given to this role by Key Laboratory of Computational Linguistics, Ministry of Education, School of Electronics Engineering and Computer Science, Peking University, is Houfeng Wang

References

Sheng-Jun Huang and Zhi-Hua Zhou. 2012. Multi-label learning by exploiting label correlations locally. In *AAAI*.

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

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.

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.

Barbara G Tabachnick, Linda S Fidell, et al. 2001. Using multivariate statistics.

Grigorios Tsoumakas and Ioannis Katakis. 2007. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13.

Grigorios Tsoumakas and Ioannis Vlahavas. 2007. Random k-labelsets: An ensemble method for multilabel classification. *Machine Learning: ECML 2007*, pages 406–417.

Min-Ling Zhang and Zhi-Hua Zhou. 2007. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048.