Hierarchical Text Classification with Latent Concepts

Xipeng Qiu, Xuanjing Huang, Zhao Liu and **Jinlong Zhou**
School of Computer Science, Fudan University
{xpqiu,xjhuang}@fudan.edu.cn, {zliu.fd,abc9703}@gmail.com

Abstract

Recently, hierarchical text classification has become an active research topic. The essential idea is that the descendant classes can share the information of the ancestor classes in a predefined taxonomy. In this paper, we claim that each class has several latent concepts and its subclasses share information with these different concepts respectively. Then, we propose a variant Passive-Aggressive (PA) algorithm for hierarchical text classification with latent concepts. Experimental results show that the performance of our algorithm is competitive with the recently proposed hierarchical classification algorithms.

1 Introduction

Text classification is a crucial and well-proven method for organizing the collection of large scale documents. The predefined categories are formed by different criterions, e.g. "Entertainment", "Sports" and "Education" in news classification, "Junk E-mail" and "Ordinary Email" in email classification. In the literature, many algorithms (Sebastiani, 2002; Yang and Liu, 1999; Yang and Pedersen, 1997) have been proposed, such as Support Vector Machines (SVM), k-Nearest Neighbor (kNN), Naïve Bayes (NB) and so on. Empirical evaluations have shown that most of these methods are quite effective in traditional text classification applications.

In past serval years, hierarchical text classification has become an active research topic in database area (Koller and Sahami, 1997; Weigend et al., 1999) and machine learning area (Rousu et al., 2006; Cai and Hofmann, 2007). Different with traditional classification, the document collections are organized as hierarchical class structure in many application fields: web taxonomies (i.e. the Yahoo! Directory http://dir.yahoo.com/ and the Open Directory Project (ODP) http://dmoz.org/), email folders and product catalogs.

The approaches of hierarchical text classification can be divided in three ways: **flat**, **local** and **global** approaches.

The **flat** approach is traditional multi-class classification in flat fashion without hierarchical class information, which only uses the classes in leaf nodes in taxonomy(Yang and Liu, 1999; Yang and Pedersen, 1997; Qiu et al., 2011).

The **local** approach proceeds in a top-down fashion, which firstly picks the most relevant categories of the top level and then recursively making the choice among the low-level categories(Sun and Lim, 2001; Liu et al., 2005).

The **global** approach builds only one classifier to discriminate all categories in a hierarchy(Cai and Hofmann, 2004; Rousu et al., 2006; Miao and Qiu, 2009; Qiu et al., 2009). The essential idea of global approach is that the close classes have some common underlying factors. Especially, the descendant classes can share the characteristics of the ancestor classes, which is similar with multi-task learning(Caruana, 1997; Xue et al., 2007).

Because the global hierarchical categorization can avoid the drawbacks about those high-level irrecoverable error, it is more popular in the machine learning domain.

However, the taxonomy is defined artificially and is usually very difficult to organize for large scale taxonomy. The subclasses of the same parent class may be dissimilar and can be grouped in different concepts, so it bring great challenge to hierarchi-

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 598–602,
Portland, Oregon, June 19-24, 2011.

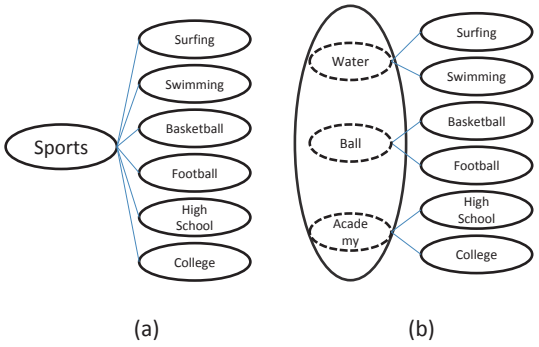

Figure 1: Example of latent nodes in taxonomy

cal classification. For example, the "Sports" node in a taxonomy have six subclasses (Fig. 1a), but these subclass can be grouped into three unobservable concepts (Fig. 1b). These concepts can show the underlying factors more clearly.

In this paper, we claim that each class may have several latent concepts and its subclasses share information with these different concepts respectively. Then we propose a variant Passive-Aggressive (PA) algorithm to maximizes the margins between latent paths.

The rest of the paper is organized as follows. Section 2 describes the basic model of hierarchical classification. Then we propose our algorithm in section 3. Section 4 gives experimental analysis. Section 5 concludes the paper.

2 Hierarchical Text Classification

In text classification, the documents are often represented with vector space model (VSM) (Salton et al., 1975). Following (Cai and Hofmann, 2007), we incorporate the hierarchical information in feature representation. The basic idea is that the notion of class attributes will allow generalization to take place across (similar) categories and not just across training examples belonging to the same category.

Assuming that the categories is $\Omega = [\omega_1, \cdots, \omega_m]$, where m is the number of the categories, which are organized in hierarchical structure, such as tree or DAG.

Give a sample $\mathbf{x}$ with its class path in the taxonomy $\mathbf{y}$, we define the feature is

$$\Phi(\mathbf{x}, \mathbf{y}) = \Lambda(\mathbf{y}) \otimes x, \tag{1}$$

where $\Lambda(\mathbf{y}) = (\lambda_1(\mathbf{y}), \cdots, \lambda_m(\mathbf{y}))^T \in \mathbb{R}^m$ and $\otimes$ is the Kronecker product.

We can define

$$\lambda_i(\mathbf{y}) = \begin{cases} t_i & \texttt{if } \omega_i \in \mathbf{y} \\ 0 & \texttt{otherwise} \end{cases}, \tag{2}$$

where $t_i >= 0$ is the attribute value for node v. In the simplest case, t_i can be set to a constant, like 1.

Thus, we can classify $\mathbf{x}$ with a score function,

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} F(\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y})), \tag{3}$$

where $\mathbf{w}$ is the parameter of $F(\cdot)$.

3 Hierarchical Text Classification with Latent Concepts

In this section, we first extent the Passive-Aggressive (PA) algorithm to the hierarchical classification (HPA), then we modify it to incorporate latent concepts (LHPA).

3.1 Hierarchical Passive-Aggressive Algorithm

The PA algorithm is an online learning algorithm, which aims to find the new weight vector $\mathbf{w}_{t+1}$ to be the solution to the following constrained optimization problem in round t.

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathbb{R}^n} \frac{1}{2}||\mathbf{w} - \mathbf{w}_t||^2 + \mathcal{C}\xi$$
$$\textbf{s.t. } \ell(\mathbf{w}; (\mathbf{x}_t, y_t)) <= \xi \textbf{ and } \xi >= 0. \tag{4}$$

where $\ell(\mathbf{w}; (\mathbf{x}_t, y_t))$ is the hinge-loss function and ξ is slack variable.

Since the hierarchical text classification is loss-sensitive based on the hierarchical structure. We need discriminate the misclassification from "nearly correct" to "clearly incorrect". Here we use **tree induced error** $\Delta(\mathbf{y}, \mathbf{y}')$, which is the shortest path connecting the nodes $\mathbf{y}_{leaf}$ and $\mathbf{y}'_{leaf}$. $\mathbf{y}_{leaf}$ represents the leaf node in path $\mathbf{y}$.

Given a example $(\mathbf{x}, \mathbf{y})$, we look for the $\mathbf{w}$ to maximize the separation margin $\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y}))$ between the score of the correct path $\mathbf{y}$ and the closest error path $\hat{\mathbf{y}}$.

$$\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y})) = \mathbf{w}^T\Phi(\mathbf{x}, \mathbf{y}) - \mathbf{w}^T\Phi(\mathbf{x}, \hat{\mathbf{y}}), \tag{5}$$

where $\hat{\mathbf{y}} = \arg\max_{\mathbf{z}\neq\mathbf{y}} \mathbf{w}^T\Phi(\mathbf{x},\mathbf{z})$ and Φ is a feature function.

Unlike the standard PA algorithm, which achieve a margin of at least 1 as often as possible, we wish the margin is related to tree induced error $\Delta(\mathbf{y},\hat{\mathbf{y}})$.

This loss is defined by the following function,

$$\ell(\mathbf{w};(\mathbf{x},\mathbf{y})) = \begin{cases} 0, & \gamma(\mathbf{w};(\mathbf{x},\mathbf{y})) > \Delta(\mathbf{y},\hat{\mathbf{y}}) \\ \Delta(\mathbf{y},\hat{\mathbf{y}}) - \gamma(\mathbf{w};(\mathbf{x},\mathbf{y})), & \text{otherwise} \end{cases} \quad (6)$$

We abbreviate $\ell(\mathbf{w};(\mathbf{x},\mathbf{y}))$ to ℓ. If $\ell = 0$ then $\mathbf{w}_t$ itself satisfies the constraint in Eq. (4) and is clearly the optimal solution. We therefore concentrate on the case where $\ell > 0$.

First, we define the Lagrangian of the optimization problem in Eq. (4) to be,

$$\mathcal{L}(\mathbf{w},\xi,\alpha,\beta) = \frac{1}{2}||\mathbf{w}-\mathbf{w}_t||^2 + \mathcal{C}\xi + \alpha(\ell-\xi) - \beta\xi \quad \textbf{s.t. } \alpha,\beta >= 0. \quad (7)$$

where α,β is a Lagrange multiplier.

We set the gradient of Eq. (7) respect to ξ to zero.

$$\alpha + \beta = \mathcal{C}. \quad (8)$$

The gradient of $\mathbf{w}$ should be zero.

$$\mathbf{w} - \mathbf{w}_t - \alpha(\Phi(\mathbf{x},\mathbf{y}) - \Phi(\mathbf{x},\hat{\mathbf{y}})) = 0 \quad (9)$$

Then we get,

$$\mathbf{w} = \mathbf{w}_t + \alpha(\Phi(\mathbf{x},\mathbf{y}) - \Phi(\mathbf{x},\hat{\mathbf{y}})). \quad (10)$$

Substitute Eq. (8) and Eq. (10) to objective function Eq. (7), we get

$$\mathcal{L}(\alpha) = -\frac{1}{2}\alpha^2||\Phi(\mathbf{x},\mathbf{y}) - \Phi(\mathbf{x},\hat{\mathbf{y}})||^2 + \alpha\mathbf{w_t}(\Phi(\mathbf{x},\mathbf{y}) - \Phi(\mathbf{x},\hat{\mathbf{y}}))) - \alpha\Delta(\mathbf{y},\hat{\mathbf{y}}) \quad (11)$$

Differentiate Eq. (11 with α, and set it to zero, we get

$$\alpha^* = \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) - \mathbf{w}_t(\Phi(\mathbf{x},\mathbf{y}) - \Phi(\mathbf{x},\hat{\mathbf{y}})))}{||\Phi(\mathbf{x},\mathbf{y}) - \Phi(\mathbf{x},\hat{\mathbf{y}})||^2} \quad (12)$$

From $\alpha + \beta = \mathcal{C}$, we know that $\alpha < \mathcal{C}$, so

$$\alpha^* = \min(\mathcal{C}, \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) - \mathbf{w}_t(\Phi(\mathbf{x},\mathbf{y}) - \Phi(\mathbf{x},\hat{\mathbf{y}})))}{||\Phi(\mathbf{x},\mathbf{y}) - \Phi(\mathbf{x},\hat{\mathbf{y}})||^2}). \quad (13)$$

3.2 Hierarchical Passive-Aggressive Algorithm with Latent Concepts

For the hierarchical taxonomy $\Omega = (\omega_1,\cdots,\omega_c)$, we define that each class ω_i has a set $H_{\omega_i} = h^1_{\omega_i},\cdots,h^m_{\omega_i}$ with m latent concepts, which are unobservable.

Given a label path $\mathbf{y}$, it has a set of several **latent paths** $H_{\mathbf{y}}$. For a latent path $\mathbf{z} \in H_{\mathbf{y}}$, a function $Proj(\mathbf{z}) \doteq \mathbf{y}$ is the projection from a latent path $\mathbf{z}$ to its corresponding path $\mathbf{y}$.

Then we can define the predict latent path $\mathbf{h}^*$ and the most correct latent path $\hat{\mathbf{h}}$:

$$\hat{\mathbf{h}} = \arg\max_{proj(\mathbf{z})\neq\mathbf{y}} w^T\Phi(\mathbf{x},\mathbf{z}), \quad (14)$$

$$\mathbf{h}^* = \arg\max_{proj(\mathbf{z})=\mathbf{y}} w^T\Phi(\mathbf{x},\mathbf{z}). \quad (15)$$

Similar to the above analysis of HPA, we re-define the margin

$$\gamma(\mathbf{w};(\mathbf{x},\mathbf{y}) = \mathbf{w}^T\Phi(\mathbf{x},\mathbf{h}^*) - w^T\Phi(\mathbf{x},\hat{\mathbf{h}}), \quad (16)$$

then we get the optimal update step

$$\alpha^*_L = \min(\mathcal{C}, \frac{\ell(\mathbf{w}_t;(\mathbf{x},\mathbf{y}))}{||\Phi(\mathbf{x},\mathbf{h}^*) - \Phi(\mathbf{x},\hat{\mathbf{h}})||^2}). \quad (17)$$

Finally, we get update strategy,

$$\mathbf{w} = \mathbf{w}_t + \alpha^*_L(\Phi(\mathbf{x},\mathbf{h}^*) - \Phi(\mathbf{x},\hat{\mathbf{h}})). \quad (18)$$

Our hierarchical passive-aggressive algorithm with latent concepts (LHPA) is shown in Algorithm 1. In this paper, we use two latent concepts for each class.

4 Experiment

4.1 Datasets

We evaluate our proposed algorithm on two datasets with hierarchical category structure.

WIPO-alpha dataset The dataset[1] consisted of the 1372 training and 358 testing document comprising the D section of the hierarchy. The number of nodes in the hierarchy was 188, with maximum depth 3. The dataset was processed into bag-of-words representation with TF·IDF

[1] World Intellectual Property Organization, http://www.wipo.int/classifications/en

input : training data set: $(\mathbf{x}_n, \mathbf{y}_n), n = 1, \cdots, N$, and parameters: $\mathcal{C}, K$
output: $\mathbf{w}$

Initialize: $\mathbf{cw} \leftarrow 0$,;
for $k = 0 \cdots K-1$ **do**
  $\mathbf{w}_0 \leftarrow 0$;
  for $t = 0 \cdots T-1$ **do**
    get $(\mathbf{x}_t, \mathbf{y}_t)$ from data set;
    predict $\hat{\mathbf{h}}, \mathbf{h}^*$;
    calculate $\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y}))$ and$\Delta(\mathbf{y}_t, \hat{\mathbf{y}}_t)$;
    if $\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y})) \leq \Delta(\mathbf{y}_t, \hat{\mathbf{y}}_t)$ **then**
      calculate α_L^* by Eq. (17);
      update $\mathbf{w}_{t+1}$ by Eq. (18). ;
    end
  end
  $\mathbf{cw} = \mathbf{cw} + \mathbf{w}_T$;
end
$\mathbf{w} = \mathbf{cw}/K$;

Algorithm 1: Hierarchical PA algorithm with latent concepts

weighting. No word stemming or stop-word removal was performed. This dataset is used in (Rousu et al., 2006).

LSHTC dataset The dataset[2] has been constructed by crawling web pages that are found in the Open Directory Project (ODP) and translating them into feature vectors (content vectors) and splitting the set of Web pages into a training, a validation and a test set, per ODP category. Here, we use the dry-run dataset(task 1).

4.2 Performance Measurement

Macro Precision, **Macro Recall** and **Macro** $F1$ are the most widely used performance measurements for text classification problems nowadays. The macro strategy computes macro precision and recall scores by averaging the precision/recall of each category, which is preferred because the categories are usually unbalanced and give more challenges to classifiers. The Macro $F1$ score is computed using the standard formula applied to the macro-level precision and recall scores.

$$MacroF1 = \frac{P \times R}{P + R}, \tag{19}$$

[2]Large Scale Hierarchical Text classification Pascal Challenge, http://lshtc.iit.demokritos.gr

Table 1: Results on WIPO-alpha Dataset."-" means that the result is not available in the author's paper.

	Accuracy	F1	Precision	Recall	TIE
PA	49.16	40.71	43.27	38.44	2.06
HPA	50.84	40.26	43.23	37.67	1.92
LHPA	51.96	41.84	45.56	38.69	1.87
HSVM	23.8	-	-	-	-
HM3	35.0	-	-	-	-

Table 2: Results on LSHTC dry-run Dataset

	Accuracy	F1	Precision	Recall	TIE
PA	47.36	44.63	52.64	38.73	3.68
HPA	46.88	43.78	51.26	38.2	3.73
LHPA	48.39	46.26	53.82	40.56	3.43

where P is the Macro Precision and R is the Macro Recall. We also use **tree induced error** (**TIE**) in the experiments.

4.3 Results

We implement three algorithms[3]: **PA**(Flat PA), **HPA**(Hierarchical PA) and **LHPA**(Hierarchical PA with latent concepts). The results are shown in Table 1 and 2. For WIPO-alpha dataset, we also compared **LHPA** with two algorithms used in (Rousu et al., 2006): **HSVM** and **HM3**.

We can see that LHPA has better performances than the other methods. From Table 2, we can see that it is not always useful to incorporate the hierarchical information. Though the subclasses can share information with their parent class, the shared information may be different for each subclass. So we should decompose the underlying factors into different latent concepts.

5 Conclusion

In this paper, we propose a variant Passive-Aggressive algorithm for hierarchical text classification with latent concepts. In the future, we will investigate our method in the larger and more noisy data.

Acknowledgments

This work was (partially) funded by NSFC (No. 61003091 and No. 61073069), 973 Program (No.

[3]Source codes are available in FudanNLP toolkit, http://code.google.com/p/fudannlp/

2010CB327906) and Shanghai Committee of Science and Technology(No. 10511500703).

References

L. Cai and T. Hofmann. 2004. Hierarchical document categorization with support vector machines. In *Proceedings of CIKM*.

L. Cai and T. Hofmann. 2007. Exploiting known taxonomies in learning overlapping concepts. In *Proceedings of International Joint Conferences on Artificial Intelligence*.

R. Caruana. 1997. Multi-task learning. *Machine Learning*, 28(1):41–75.

D. Koller and M Sahami. 1997. Hierarchically classifying documents using very few words. In *Proceedings of the International Conference on Machine Learning (ICML)*.

T.Y. Liu, Y. Yang, H. Wan, H.J. Zeng, Z. Chen, and W.Y. Ma. 2005. Support vector machines classification with a very large-scale taxonomy. *ACM SIGKDD Explorations Newsletter*, 7(1):43.

Youdong Miao and Xipeng Qiu. 2009. Hierarchical centroid-based classifier for large scale text classification. In *Large Scale Hierarchical Text classification (LSHTC) Pascal Challenge*.

Xipeng Qiu, Wenjun Gao, and Xuanjing Huang. 2009. Hierarchical multi-class text categorization with global margin maximization. In *Proceedings of the ACL-IJCNLP 2009 Conference*, pages 165–168, Suntec, Singapore, August. Association for Computational Linguistics.

Xipeng Qiu, Jinlong Zhou, and Xuanjing Huang. 2011. An effective feature selection method for text categorization. In *Proceedings of the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining*.

Juho Rousu, Craig Saunders, Sandor Szedmak, and John Shawe-Taylor. 2006. Kernel-based learning of hierarchical multilabel classification models. In *Journal of Machine Learning Research*.

G. Salton, A. Wong, and CS Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

F. Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys*, 34(1):1–47.

A. Sun and E.-P Lim. 2001. Hierarchical text classification and evaluation. In *Proceedings of the IEEE International Conference on Data Mining*.

A. Weigend, E. Wiener, and J Pedersen. 1999. Exploiting hierarchy in text categorization. In *Information Retrieval*.

Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. 2007. Multi-task learning for classification with dirichlet process priors. *The Journal of Machine Learning Research*, 8:63.

Y. Yang and X. Liu. 1999. A re-examination of text categorization methods. In *Proc. of SIGIR*. ACM Press New York, NY, USA.

Y. Yang and J.O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proc. of Int. Conf. on Mach. Learn. (ICML)*, volume 97.