Hierarchical Multi-Class Text Categorization with Global Margin Maximization

Xipeng Qiu
School of Computer Science
Fudan University
xpqiu@fudan.edu.cn

Wenjun Gao
School of Computer Science
Fudan University
wjgao616@gmail.com

Xuanjing Huang
School of Computer Science
Fudan University
xjhuang@fudan.edu.cn

Abstract

Text categorization is a crucial and well-proven method for organizing the collection of large scale documents. In this paper, we propose a hierarchical multi-class text categorization method with global margin maximization. We not only maximize the margins among leaf categories, but also maximize the margins among their ancestors. Experiments show that the performance of our algorithm is competitive with the recently proposed hierarchical multi-class classification algorithms.

1 Introduction

In the past serval years, hierarchical text categorization 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).

Hierarchical categorization methods can be divided in two types: local and global approaches (Wang et al., 1999; Sun and Lim, 2001). A local approach usually 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. The global approach builds only one classifier to discriminate all categories in a hierarchy. Due that the global hierarchical categorization can avoid the drawbacks about those high-level irrecoverable error, it is more popular in the machine learning domain.

The essential idea behind global approach is that the close classes(nodes) 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). A key problem for global hierarchical categorization is how to combine these underlying factors.

In this paper, we propose an method for hierarchical multi-class text categorization with global margin maximization. We emphasize that it is important to separate all the nodes of the correct path in the class hierarchy from their sibling node, then we incorporate such information into the formulation of hierarchical support vector machine.

The rest of the paper is organized as follows. Section 2 describes the basic model of multi-class hierarchical categorization with maximizing margin. Then we propose our improved versions in section 3. Section 4 gives the experimental analysis. Section 5 concludes the paper.

2 Hierarchical Multi-Class Text Categorization

Multiclass SVM can be generalized to the problem of hierarchical categorization (Cai and Hofmann, 2007), which has more than two categories in most of the case. Denote Y_i as the multilabels of $\mathbf{x}_i$ and $\bar{Y}_i$ the multilabels set not in Y_i. The separation margin of $\mathbf{w}$, with respect to $\mathbf{x}_i$, can be approximated as:

$$\gamma_i(\mathbf{w}) = \min_{\mathbf{y}\in Y_i, \bar{\mathbf{y}}\in \bar{Y}_i} \langle \Phi(\mathbf{x}_i, \mathbf{y}) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}), \mathbf{w} \rangle \tag{1}$$

The loss function can be accommodated to multi-class SVM to scale the penalties for margin violations proportional to the loss. This is motivated by the fact that margin violations involving an incorrect class with high loss should be penalized more severely. So the cost-sensitive hierarchical multiclass formulation takes takes the following form:

$$\min_{\mathbf{w},\xi} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \xi_i \tag{2}$$

$$\text{s.t.} \langle \mathbf{w}, \delta\Phi_i(\mathbf{y}, \bar{\mathbf{y}}) \rangle \geq 1 - \frac{\xi_i}{l(\mathbf{y}, \bar{\mathbf{y}})}, \ (\forall i, \mathbf{y}\in Y_i, \bar{\mathbf{y}}\in \bar{Y}_i)$$

$$\xi_i \geq 0 (\forall i)$$

Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 165–168,
Suntec, Singapore, 4 August 2009. ©2009 ACL and AFNLP

where $\delta\Phi_i(\mathbf{y}, \bar{\mathbf{y}}) = \Phi(\mathbf{x}_i, \mathbf{y}) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}})$, $l(\mathbf{y}, \bar{\mathbf{y}}) > 0$ and $\Phi(\mathbf{x}, \mathbf{y})$ is the joint feature of input $\mathbf{x}$ and output $\mathbf{y}$, which can be represented as:

$$\Phi(\mathbf{x}, \mathbf{y}) = \Lambda(\mathbf{y}) \otimes \phi(\mathbf{x}) \tag{3}$$

where $\otimes$ is the tensor product. $\Lambda(\mathbf{y})$ is the feature representation of $\mathbf{y}$.

Thus, we can classify a document $\mathbf{x}$ to label $y^\star$:

$$y^\star = \arg\max_y F(\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y})) \tag{4}$$

where $F(\cdot)$ is a map function.

There are different kinds of loss functions $l(\mathbf{y}, \bar{\mathbf{y}})$.

One is the**zero-one loss**, $l_{0/1}(\mathbf{y}, \mathbf{u}) = [\mathbf{y} \neq \mathbf{u}]$.

Another is specially designed for the hierarchy is **tree loss**(Dekel et al., 2004). Tree loss is defined as the length of the path between two multilabels with positive microlabels,

$$l_{tr} = |path(i : \mathbf{y}_i = 1, j : \mathbf{u}_j = 1)| \tag{5}$$

(Rousu et al., 2006) proposed a simplified version of l_H, namely $l_{\tilde{H}}$:

$$l_{\hat{H}} = \sum_j c_j [y_j \neq u_j \& y_{pa}(j) = u_{pa(j)}], \tag{6}$$

that penalizes a mistake in a child only if the label of the parent was correct. There are some different choices for setting c_j. One naive idea is to use a uniform weighting ($c_j = 1$). Another possible choice is to divide the loss among the sibling:

$$c_{root} = 1, c_j = c_{Parent(j)}/(|Sib(j)| + 1) \tag{7}$$

Another possible choice is to scale the loss by the proportion of the hierarchy that is in the subtree $T(j)$ rooted by j:

$$c_j = |T(j)|/|T(root)| \tag{8}$$

Using these scaling weights, the derived losses are referred as $l_{\hat{uni}}$,$l_{\hat{sib}}$ and $l_{\hat{sub}}$ respectively.

3 Hierarchical Multi-Class Text Categorization with Global Margin Maximization

In previous literature (Cai and Hofmann, 2004; Tsochantaridis et al., 2005), they focused on separating the correct path from those incorrect path. Inspired by the example in Figure 1, we emphasize it is also important to separate the ancestor node in the correct path from their sibling node.

The vector $\mathbf{w}$ can be decomposed in to the set of $\mathbf{w}_i$ for each node (category) in the hierarchy. In Figure 1, the example hierarchy has 7 nodes and 4 of them are leaf nodes. The category is encode as an integer, $1, \ldots, 7$. Suppose that the training pattern $\mathbf{x}$ belongs to category 4. Both $\mathbf{w}$ in the Figure 1a and Figure 1b can successfully classify $\mathbf{x}$ into category 4, since $F(\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}_4)) = \sum_{1,2,4} \langle \mathbf{w}_i, \mathbf{x} \rangle$ is the maximal among all the possible discriminate functions. So both learned parameter $\mathbf{w}$ is acceptable in current hierarchical support vector machine.

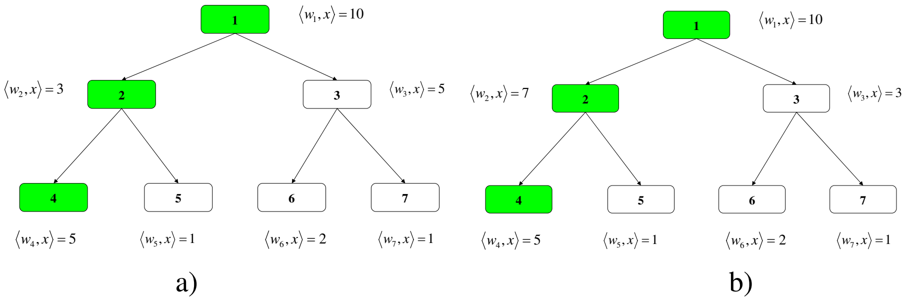
Here we claim the $\mathbf{w}$ in Figure 1b is better than the $\mathbf{w}$ in Figure 1a. Since we notice in Figure 1a, the discriminate function $\langle \mathbf{w}_2, \mathbf{x} \rangle$ is smaller than the discriminate function $\langle \mathbf{w}_3, \mathbf{x} \rangle$. The discriminate function $\langle \mathbf{w}_i, \mathbf{x} \rangle$ measures the similarity of $\mathbf{x}$ to category i. The larger the discriminate function is, the more similar $\mathbf{x}$ is to category i. Since category 2 is in the path from the root to the correct category and category 3 is not, intuitively, $\mathbf{x}$ should be closer to category 2 than category 3. But the discriminate function in Figure 1a is contradictive to this assumption. But such information is reflected correctly in Figure 1b. So we conclude $\mathbf{w}$ in Fig. 1b is superior to $\mathbf{w}$ in 1a.

Here we propose a novel formulation to incorporate such information. Denote A_i as the multilabel in Y_i that corresponds to the nonleaf categories and $Sib(z)$ denotes the sibling nodes of z, that is the set of nodes that have the same parent with z, except z itself. Implementing the above idea, we can get the following formulation:

$$\begin{aligned}
\min_{\mathbf{w},\xi,\zeta} \frac{1}{2}\|\mathbf{w}\|^2 \quad &+ \quad C_1 \sum_i \xi_i + C_2 \sum_i \zeta_i \qquad (9) \\
\text{s.t.} \langle \mathbf{w}, \delta\Phi_i(\mathbf{y}, \bar{\mathbf{y}}) \rangle \quad &\geq \quad 1 - \frac{\xi_i}{l(\mathbf{y}, \bar{\mathbf{y}})}, (\forall i, \begin{smallmatrix} \mathbf{y} \in Y_i \\ \bar{\mathbf{y}} \in \bar{Y_i} \end{smallmatrix}) \\
\langle \mathbf{w}, \delta\Phi_i(\mathbf{z}, \bar{\mathbf{z}}) \rangle \quad &\geq \quad 1 - \frac{\zeta_i}{l(\mathbf{z}, \bar{\mathbf{z}})}, (\forall i, \begin{smallmatrix} \mathbf{z} \in A(i) \\ \bar{\mathbf{z}} \in Sib(\mathbf{z}) \end{smallmatrix}) \\
\xi_i \quad &\geq \quad 0 (\forall i) \\
\zeta_i \quad &\geq \quad 0 (\forall i)
\end{aligned}$$

It arrives at the following Lagrangian:

$$\begin{aligned}
&L(\mathbf{w}, \xi_1, ..., \xi_n, \zeta_1, ..., \zeta_n) \\
&= \frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_i \xi_i + C_2 \sum_i \zeta_i \\
&- \sum_i \sum_{\substack{\mathbf{y} \in Y_i \\ \bar{\mathbf{y}} \in \bar{Y_i}}} \alpha_{i\mathbf{y}\bar{\mathbf{y}}} (\langle \mathbf{w}, \delta\Phi_i(\mathbf{y}, \bar{\mathbf{y}}) \rangle - 1 + \frac{\xi_i}{l(\mathbf{y}, \bar{\mathbf{y}})})
\end{aligned}$$



$\langle w_1, x\rangle = 10$
$\langle w_2, x\rangle = 3$
$\langle w_3, x\rangle = 5$
$\langle w_4, x\rangle = 5$
$\langle w_5, x\rangle = 1$
$\langle w_6, x\rangle = 2$
$\langle w_7, x\rangle = 1$
1
2
3
4
5
6
7
a)
$\langle w_1, x\rangle = 10$
$\langle w_2, x\rangle = 7$
$\langle w_3, x\rangle = 3$
$\langle w_4, x\rangle = 5$
$\langle w_5, x\rangle = 1$
$\langle w_6, x\rangle = 2$
$\langle w_7, x\rangle = 1$
1
2
3
4
5
6
7
b)

Figure 1: Two different discriminant function in a hierarchy

$$-\sum_i \sum_{\substack{\mathbf{z}\in A_i\\ \bar{\mathbf{z}}\in Sib(\mathbf{z})}} \beta_{i\mathbf{z}\bar{\mathbf{z}}}(\langle \mathbf{w}, \delta\Phi_i(\mathbf{z},\bar{\mathbf{z}})\rangle - 1 + \frac{\zeta_i}{l(\mathbf{z},\bar{\mathbf{z}})})$$
$$-\sum_i c_i\xi_i - \sum_i d_i\zeta_i \qquad (10)$$

The dual QP becomes

$$\max_{\boldsymbol{\alpha}} \Theta(\boldsymbol{\alpha}) = \sum_i \sum_{\substack{\mathbf{y}\in Y_i\\ \bar{\mathbf{y}}\in \bar{Y}_i}} \alpha_{i\mathbf{y}\bar{\mathbf{y}}} + \sum_i \sum_{\substack{\mathbf{z}\in A_i\\ \bar{\mathbf{z}}\in Sib(\mathbf{z})}} \beta_{i\mathbf{z}\bar{\mathbf{z}}}$$
$$-\frac{1}{2}\sum_{i,j} \sum_{\substack{\mathbf{y}\in Y_i\\ \bar{\mathbf{y}}\in \bar{Y}_i}} \sum_{\substack{\mathbf{r}\in Y_j\\ \bar{\mathbf{r}}\in \bar{Y}_j}} \theta^1_{i,j,\mathbf{y},\bar{\mathbf{y}},\mathbf{r},\bar{\mathbf{r}}} \qquad (11)$$
$$-\frac{1}{2}\sum_{i,j} \sum_{\substack{\mathbf{z}\in A_i\\ \bar{\mathbf{z}}\in Sib(\mathbf{z})}} \sum_{\substack{\mathbf{k}\in A_j\\ \bar{\mathbf{k}}\in Sib(\mathbf{k})}} \theta^2_{i,j,\mathbf{z},\bar{\mathbf{z}},\mathbf{k},\bar{\mathbf{k}}},$$

$$\text{s.t.}\alpha_{i\mathbf{y}\bar{\mathbf{y}}} \geq 0, \qquad (12)$$
$$\beta_{j\mathbf{z}\bar{\mathbf{z}}} \geq 0, \qquad (13)$$
$$\sum_{\substack{\mathbf{y}\in Y_i\\ \bar{\mathbf{y}}\in \bar{Y}_i}} \frac{\alpha_{i\mathbf{y}\bar{\mathbf{y}}}}{l(\mathbf{y},\bar{\mathbf{y}})} \leq C_1, \qquad (14)$$
$$\sum_{\substack{\mathbf{z}\in A_i\\ \bar{\mathbf{z}}\in Sib(\mathbf{z})}} \frac{\beta_{i\mathbf{z}\bar{\mathbf{z}}}}{l(\mathbf{z},\bar{\mathbf{z}})} \leq C_2, \qquad (15)$$

where $\theta^1_{i,j,\mathbf{y},\bar{\mathbf{y}},\mathbf{r},\bar{\mathbf{r}}} = \alpha_{i\mathbf{y}\bar{\mathbf{y}}}\alpha_{j\mathbf{r}\bar{\mathbf{r}}}\langle \delta\Phi_i(\mathbf{y},\bar{\mathbf{y}}), \delta\Phi_j(\mathbf{r},\bar{\mathbf{r}})\rangle$ and $\theta^2_{i,j,\mathbf{z},\bar{\mathbf{z}},\mathbf{k},\bar{\mathbf{k}}} = \beta_{i\mathbf{z}\bar{\mathbf{z}}}\beta_{j\mathbf{k}\bar{\mathbf{k}}}\langle \delta\Phi_i(\mathbf{z},\bar{\mathbf{z}}), \delta\Phi_j(\mathbf{k},\bar{\mathbf{k}})\rangle$.

3.1 Optimization Algorithm

The derived QP can be very large, since the number of α and β variables is up to $O(n*2^N)$, where n is number of training pattern and N is the number of nodes in the hierarchy. But two properties of the dual problem can be exploited to design a much more efficient optimization.

First, the constraints in the dual problem Eq. 11 - Eq. 15 factorize over the instance index for both α-variables and β-variables. The constraints in Eq. 14 do not couple α-variables and β-variables together. Further, dual variables $\alpha_{i\mathbf{y}\bar{\mathbf{y}}}$ and $\alpha_{j\mathbf{y}'\bar{\mathbf{y}}'}$ belonging to different training instances i and j do not join in a same constraints. This inspired an optimization procedure which iteratively performs subspace optimization over all dual variables $\alpha_{i\mathbf{y}\bar{\mathbf{y}}}$ belonging to the same training instance. This will in general reduced to a much smaller QP, since it freezes all $\alpha_{j\mathbf{y}\bar{\mathbf{y}}}$ with $j \neq i$ and β-variables at their current values. This strategy can be applied in solving β-variables.

Secondly, the number of active constraints at the solution is expected to be relatively small, since only a small fraction of categories $\bar{\mathbf{y}} \in \bar{Y}_i$ (or $\bar{\mathbf{y}} \in Sib(\mathbf{y})$ when $\mathbf{y} \in A_i$) will typically fail to achieve the required margin. The expected sparseness of the variable for the dual problem can be exploited by employing a variable selection strategy. Equivalently, this corresponds to a cutting plane algorithm for the primal QP. Intuitively, we will identify the most violated margin constraint with index $(i, \mathbf{y}, \bar{\mathbf{y}})$ and then add the corresponding variable to the optimization problem. This means that we start with extremely sparse problems and only successively increase the number of variables in the active set. This general approach to deal with large linear or quadratic optimization problems is also known as column selection. In practice, it is often not necessary to optimize until final convergence, which adds to the attractiveness of this approach.

We have used the LOQO optimization package (Vanderbei, 1999) in our experiments.

4 Experiment

We evaluate our proposed model on the section D in the WIPO-alpha collection[1], which consists of the 1372 training and 358 testing document. The

[1] World Intellectual Property Organization (WIPO)

Table 1: Prediction losses (%) obtained on WIPO. The values per column is calculated with the different loss function.

Test Train		$l_{0/1}$	$l_{\triangle}$	l_{tr}	l_{uni}	l_{sib}	l_{sub}
$l_{0/1}$	HSVM	48.6	188.8	94.4	97.2	5.4	7.5
	HSVM-S	**48.3**	**186.6**	**93.3**	**96.6**	**5.2**	**7.4**
$l_{\triangle}$	HSVM	49.7	187.7	93.9	99.4	5.0	7.1
	HSVM-S	**47.8**	**165.3**	**89.7**	90.5	**4.8**	**6.9**
	HM3	70.9	167.0	-	89.1	5.0	7.0
l_{tr}	HSVM	49.4	186.0	93.0	98.9	5.0	7.5
	HSVM-S	**48.9**	**181.4**	**90.2**	**97.8**	**4.9**	**7.1**
$l_{\hat{uni}}$	HSVM	47.2	181.0	90.5	94.4	5.0	7.0
	HSVM-S	**46.9**	179.3	**88.7**	91.9	**4.9**	**6.9**
	HM3	70.1	172.1	-	88.8	5.2	7.4
$l_{\hat{sib}}$	HSVM	49.4	184.9	92.5	98.9	4.8	7.4
	HSVM-S	**48.9**	**170.2**	**91.6**	**90.8**	**4.7**	7.4
	HM3	64.8	172.9	-	92.7	4.8	7.1
$l_{\hat{sub}}$	HSVM	50.6	189.9	95.0	101.1	5.2	7.5
	HSVM-S	**47.2**	**169.4**	**85.2**	**89.4**	**4.3**	**6.6**
	HM3	65.0	170.9	-	91.9	4.8	7.2

number of nodes in the hierarchy is 188, with maximum depth 3.

We compared the performance of our proposed method HSVM-S with two algorithms: HSVM(Cai and Hofmann, 2007) and HM3(Rousu et al., 2006).

4.1 Effect of Different Loss Function

We compare the methods based on different loss functions, $l_{0/1}$, $l_{\triangle}$, l_{tr}, $l_{\hat{u}ni}$, $l_{\hat{s}ib}$ and $l_{\hat{s}ub}$. The performances for three algorithms can be seen in Table 1. Those empty cells, denoted by "-", are not available in (Rousu et al., 2006).

As expected, $l_{0/1}$ is inferior to other hierarchical losses by getting poorest performance in all the testing losses, since it can not take into account the hierarchical information between categories. The results suggests that training with a hierarchical losses function, like $l_{\hat{s}ib}$ or $l_{\hat{u}ni}$, would lead to a better reduced $l_{0/1}$ on the test set as well as in terms of the hierarchical loss. In Table 1, we can also point out that when training with the same hierarchical loss, the performance of HSVM-S is better than HSVM under the measure of most hierarchical losses, since HSVM-S includes more hierarchical information,the relationship between the sibling categories, than HSVM which only separate the leave categories.

5 Conclusion

In this paper we present a hierarchical multi-class document categorization, which focus on maximize the margin of the classes at the different levels in the class hierarchy. In future work, we plan to extend the proposed hierarchical learning method to the case where the hierarchy is a DAG instead of tree and scale up the method further.

Acknowledgments

This work was (partially) funded by Chinese NSF 60673038, Doctoral Fund of Ministry of Education of China 200802460066, and Shanghai Science and Technology Development Funds 08511500302.

References

L. Cai and T Hofmann. 2004. Hierarchical document categorization with support vector machines. In *Proceedings of the ACM Conference on Information and Knowledge Management*.

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.

Ofer Dekel, Joseph Keshet, and Yoram Singer. 2004. Large margin hierarchical classification. In *Proceedings of the 21 st International Conference on Machine Learning*.

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

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*.

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

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. In *Journal of Machine Learning*.

R. J. Vanderbei. 1999. Loqo: An interior point code for quadratic programming. In *Optimization Methods and Software*.

K. Wang, S. Zhou, and S Liew. 1999. Building hierarchical classifiers using class proximities. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*.

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