Maximum-margin Framework for Training Data Synchronization in Large-scale Hierarchical Classification

Rohit Babbar, Ioannis Partalas, Eric Gaussier and Massih-Reza Amini

Laboratoire d'Informatique de Grenoble
Université Joseph Fourier, Grenoble, France
{firstname.lastname}@imag.fr

Abstract. In the context of supervised learning, the training data for large-scale hierarchical classification consist of (i) a set of input-output pairs, and (ii) a hierarchy structure defining parent-child relation among class labels. It is often the case that the hierarchy structure given a-priori is not optimal for achieving high classification accuracy. This is especially true for *large* web-taxonomies such as Yahoo! directory which consist of tens of thousand of classes, and also the fact that an important goal of hierarchy design is to render better navigability and browsing. In this work, we propose a maximum-margin framework for automatically adapting the given hierarchy based on the set of input-output pairs to yield a new hierarchy. The proposed method is not only theoretically justified but also provides a more principled approach for hierarchy flattening techniques proposed earlier, which are ad-hoc and empirical in nature. The empirical results on large-scale public datasets demonstrate that classification with new hierarchy leads to better or comparable generalization performance than the hierarchy flattening techniques. Moreover, since the proposed method largely maintains the overall hierarchical structure, it leads to faster prediction and lower space complexity.

1 Introduction

Large-scale web taxonomies, e.g. the Open Directory Project (ODP), consist of millions of websites, distributed among hundreds of thousand classes which are arranged in a tree hierarchy. For example, ODP has around 5 million websites and the hierarchy contains over 1 million classes. Due to the ever-increasing scale of data from various sources on the web, there is a definite requirement to partially or fully eliminate the manual effort involved in managing such taxonomies. In this context, large-scale hierarchical classification systems aim to automatically classify documents to target classes using also the hierarchical information. The main challenge in large-scale hierarchical classification is to exploit the hierarchical structure to design a scalable classification system which has acceptable prediction accuracy as well as training and prediction speed. In order to evaluate the current state of art in this domain, open challenges such as the Pascal Large Scale Hierarchical Text Classification (LSHTC) [1] have been organized.

[1] http://lshtc.iit.demokritos.gr/

Most approaches exploit the hierarchy structure to design appropriate loss functions for classification and use it to apply the divide-and-conquer paradigm to keep the scale of the classification problem manageable. However, the taxonomy structure given a-priori as part of the training data may not be best suited to yield high classification accuracy due to the following reasons:

1. Large-scale web taxonomies are designed with an intent of better user-experience and navigability, and not for the goal of classification.
2. Taxonomy design is subject to certain degree of arbitrariness based on personal choices and preferences of the editors.
3. The large-scale nature of such taxonomies poses difficulties in manually designing good taxonomies for classification.

In the recent work by [3] on relatively smaller taxonomies, the impact of arbitrariness on loss-function design is minimized by appropriately calibrating the edge distance between the true and predicted class. In similar spirit of taxonomy adaptation, approaches based on flattening the hierarchy such as [7, 9], have been proposed in LSHTC for large-scale settings which lead to improvement in classification accuracy as compared to using the original hierarchy. The motivation for these hierarchy flattening approaches is to minimize the error propagation due to a longer cascade from the root to leaves. Hierarchy flattening approaches remove entire levels in the hierarchy by replacing all the parents in that level by its children. This is illustrated in Figure 1 where the first and the third levels of the hierarchy are removed. Such approaches based on flattening entire levels suffer from the following drawbacks:

– These are based on ad-hoc heuristics and a-priori it is not clear which levels in the hierarchy should be flattened. This is crucial for hierarchies such as Yahoo! Directory which have more than 10 levels.
– Excessive flattening leads to increase in training and prediction speed, both of these factors adversely impact applicability of the resulting hierarchical classifiers in many scenarios of practical importance.

In order to tackle the incompatibility of the given hierarchy structure among target classes and the set of input-output pairs in large-scale hierarchical classification, we propose a principled strategy for adapting the hierarchy to better suit the classification problem at hand.

1.1 Our Contributions

In this work, we present a margin-based framework for choosing the most appropriate candidate nodes for replacement by their children nodes rather than all the nodes in a level. The proposed approach for taxonomy adaptation is based on well-founded theoretical results for generalization error analysis of margin-based classifiers deployed in a tree-based top down cascade [5]. The replacement is performed only for those classes which are more likely to be confused with other classes at the same level in the hierarchy. We exploit the margin information obtained at optimality while training the one-vs-rest classifiers to determine

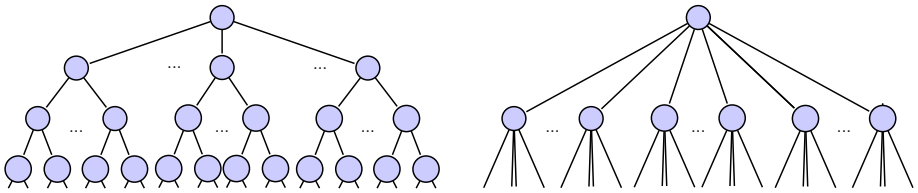

Fig. 1. Flattening the first and the third levels (right) of the original hierarchy (left).

the extent of confusion for each candidate class. This approach can be seen as synchronization of the two components of training data, taxonomy information on one hand and the set of input-output pairs on the other hand.

The proposed method is based on a more principled approach for node replacement as compared to ad-hoc methods based on flattening entire layers. As a result, our method is easily applicable to taxonomy structures in which the cascade length is arbitrarily long. Another advantage of our approach is that by choosing the most relevant candidates for replacement, it limits the extent of flattening and maintains the over-all hierarchical structure. On the other hand, entire level flattening methods lead to excessive flattening and hence increase the training and prediction time by multiple folds.

1.2 Other Related Work

Some of the earlier works on exploiting hierarchy among target classes for the purpose of text classification have been studied in [2, 4] wherein the number of target classes were limited to a few hundreds. However, the work by [6] is among the pioneering in hierarchical classification towards addressing Web-scale directories such as Yahoo! directory consisting of over 100,000 target classes. The authors analyze the performance with respect to accuracy and training time complexity for flat and hierarchical classification. More recently, other techniques for large-scale hierarchical text classification have been proposed. Prevention of error propagation by applying *Refined Experts* trained on a validation set was proposed in [1]. In this approach, bottom-up information propagation is performed by utilizing the output of the lower level classifiers in order to improve classification at top level. Deep Classification [10] proposes to first identify a much smaller subset of target classes. Prediction of a test instance is then performed by re-training Naive Bayes classifier on the subset of target classes identified from the first step.

2 Problem Setup

In single-label multi-class hierarchical classification, the training data can be represented by a set of input-output pairs $S = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ and hierarchical structure among target classes $\mathcal{G}$. In the context of text classification, $\mathbf{x}^{(i)} \in \mathcal{X}$ denotes the vector representation of document i in the input space $\mathcal{X} \subseteq \mathbb{R}^d$.

Assuming that there are K classes denoted by the set $\mathcal{Y} = \{1 \dots K\}$, the label $y^{(i)} \in \mathcal{Y}$ represents the class associated with the instance $\mathbf{x}^{(i)}$. The hierarchy in the form of rooted tree is given by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} \supseteq \mathcal{Y}$ denotes the set of nodes of $\mathcal{G}$, and $\mathcal{E}$ denotes the set of edges with parent-to-child orientation. Let $v_0 \in \mathcal{V}$ denote the root node of the hierarchy tree. In this setup, the leaves of the tree which form the set of target classes, which is given by $\mathcal{Y} = \{u \in \mathcal{V} : \nexists v \in \mathcal{V}, (u, v) \in \mathcal{E}\}$. Since the class hierarchy represents a transition from general to specialization of a concept as one traverses from root towards leaves, the documents which belong a particular leaf node also belong to all the nodes on the path from the root to that lead node.

In the above setup, given a new test instance $\mathbf{x}$, the goal is to predict the class $\widehat{y}$. This is done by making a sequence of predictions iteratively in a top-down fashion starting from the root until a leaf node is reached. At each non-leaf node $v \in \mathcal{V}$, a score $f_c(\mathbf{x}) \in \mathbb{R}$ is computed for each child c and the child $\widehat{c}$ with the maximum score is predicted i.e. $\widehat{c} = \underset{c:(v,c)\in\mathcal{E}}{\operatorname{argmax}} f_c(\mathbf{x})$.

In addition to being *highly accurate* for prediction, we also focus on *prediction speed*, which are two seemingly contradicting design requirements for a machine learning algorithm.

The motivation of approaches based on layer flattening such as [7, 9] illustrated in Figure 1 is that by reducing the length of the cascade, the extent of propagation error can be reduced. However, these approaches lead to multiple folds increase in training time as shown in [9]. Prediction speed also suffers by employing excessive flattening as studied in the work by [6] showing that the space complexity of a flat classifier is much higher than a hierarchical model. Moreover, for predicting an unseen test instance in a K class problem, one needs to evaluate $O(K)$ classifiers in flat classification as against $O(\log K)$ classifiers in a top-down manner. In order to achieve a better trade-off among various metrics of interest in large-scale hierarchical classification, we next propose a technique for hierarchy adaptation which not only provides comparable or better performance to level flattening techniques but also maintains the overall hierarchical structure to enjoy faster training and prediction speed.

3 Taxonomy Adaptation in Large-scale Hierarchical Classification

In this section, we propose a principled approach to adapt the taxonomy given a-priori as part of the training data by using the input-output pairs to output a taxonomy which leads to better accuracy. For our analysis, we focus on $L2$-regularized $L2$-loss linear Support Vector Machine (SVM), wherein the decision function $f_c(\mathbf{x})$ is modeled as a linear classifier such that $f_c(\mathbf{x}) = \mathbf{w}_c^T \mathbf{x}$. To learn an SVM-based discriminative classifier for node v, we solve the following optimization problem for each child c of v

$$f_c^* = \min_{\mathbf{w}_c} \left[\frac{1}{2} {\mathbf{w}_c}^T \mathbf{w}_c + C \sum_{\{i:y^{(i)} \in L_v\}} (\max(0, 1 - \operatorname{sgn}(y^{(i)} \circ L_c) {\mathbf{w}_c}^T \mathbf{x}^{(i)}))^2 \right] \quad (1)$$

where L_v denotes the set of leaves in the subtree rooted at node v and C denotes the parameter for mis-classification penalty. We focus on one-vs-rest technique to tackle the multi-class nature of the classification problem of identifying the most relevant child c of parent node v and hence y_i and L_c are related such that

$$\text{sgn}(y^{(i)} \circ L_c) = \begin{cases} +1 & \text{if } y^{(i)} \in L_c \\ -1 & \text{otherwise} \end{cases}$$

3.1 Margin-based approach to Taxonomy Adaptation

We derive our intuition from the recently proposed result[2] on the generalization error of maximum-margin classifiers deployed in the tree structure which can be stated as follows:

Theorem 1. *[5] Let m random input-output pairs are correctly classified using $\mathcal{G}$ containing $|\mathcal{V}|$ decision nodes with margins $\{\gamma_j, \forall j \in \mathcal{G}\}$, then the generalization error with probability greater than $1-\delta$ is less than*

$$\frac{130R^2}{m}(D' log(4em) log(4m) + |\mathcal{V}| log(2m) - log(\frac{2}{\delta}))$$

where $D' = \sum_j^{|\mathcal{V}|} \frac{1}{\gamma_j^2}$ and R is the radius of the ball containing the distribution's support.

Though the above result is stated for the separable case, it indicates that in order to achieve better generalizability, one needs to decrease the quantity D'. Clearly, this quantity can be reduced if one removes those nodes from the tree which correspond to lower margin. The decision nodes with lower margin correspond to those classification problems which are relatively harder as compared to those nodes at which higher margin can be achieved. This is illustrated in Figure 2, in which not all nodes at a layer are replaced by their children but only those for which the margin is among the lowest. This strategy essentially lead to reducing the effective VC dimension or the Rademacher complexity of the overall hierarchical classifier and leading to a reduction in the generalization error in accordance with the Theorem 1.

Since we deal with the non-separable case, we need to remove those nodes for which the inverse of the margin and empirical error are jointly maximum. This quantity is captured for each decision node $c \in \{\mathcal{V}/v_0\}$ by the optimal value obtained from the objective function value f_c^* as given in Equation 1. For each parent node $v \in \mathcal{V}$, we consider the respective value of f_c^* $\forall c \in \mathcal{V}, (v,c) \in \mathcal{E}$ for each child c of v. The values f_c^* are sorted in decreasing order, which represents the preferential ordering of the nodes to be considered for flattening. Top r-ranked nodes are flattened for which f_c^* is among the highest, where r is chosen based on the distribution of these values. Once the difference between the f_c^* values of the current and next candidate child node is more than the previous

[2] The theorem can be found in the supplementary material of [5].

difference, the flattening procedure for the current parent node is stopped. This process is repeated for all the parent nodes in the hierarchy irrespective of the length of the cascade. The algorithmic depiction of the procedure is shown in Algorithm 1. Note that the value of parameter C does not significantly affect the ordering of the nodes pruned by the proposed algorithm.

Since the algorithm maintains the overall hierarchy structure, it benefits from the properties of low space complexity and faster prediction, as we demonstrate in the next section. It was also observed that the resulting hierarchy after applying the transformation as given by the algorithm leads to more balanced classification problems at various levels.

Algorithm 1 The proposed comparative evaluation procedure.

Require: a hierarchy $\mathcal{G}$, input-output set S
  Train $L2$-regularized,$L2$-loss SVM in a top-down manner
  $gap \leftarrow 0$
  for $v \in \mathcal{V}$ **do**
    Sort of the child nodes in decreasing order of f_c^*
    Flatten 1st and 2nd ranked child nodes, say c_1 and c_2
    $gap = f_{c_1}^* - f_{c_2}^*$
    $c_{prev} \leftarrow c_2$ ▷ Set the previous flattened node to c_2
    for $c \in \mathcal{V} - \{c_1, c_2\}, (v, c) \in \mathcal{E}$ **do**
      if $f_{c_{prev}}^* - f_c^* < gap$ **then**
        Flatten c
        $gap \leftarrow f_{c_{prev}}^* - f_c^*$
        $c_{prev} \leftarrow c$ ▷ Set the previous flattened node to c
      else
        break
      end if
    end for
  end for
  return $\mathcal{G}$

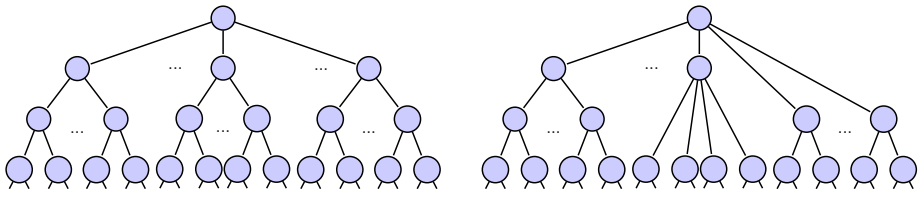

Fig. 2. Partial flattening (right) of the original hierarchy (left) for the proposed method.

4 Experiments and Results

We use the publicly available DMOZ data set from the LSHTC 2010 (**DMOZ-1**) and a subset of DMOZ from the LSHTC 2011 (**DMOZ-2**). The datasets, after having been pre-processed by stemming and stopword removal, appear in the LibSVM format. The data was processed to convert the term-frequency (tf) representation to tf-idf format. Table 1 presents the numeric values corresponding to the important properties of the dataset.

Properties	**DMOZ-1**	**DMOZ-2**
Tr. Set Size	93,805	36,834
Feature Set Size	347,255	155,641
Target Classes	12,294	3,672
Test Set Size	34,880	36,834

Table 1. Dataset Properties

We compare three strategies to evaluate the impact of techniques on classification accuracy : (i) Fully Hierarchical (FH) technique which uses the original hierarchy, (ii) Top Level Flattening (TLF) by removing the first layer, (iii) Multiple Level Flattening by removing first and third levels (MLF) as proposed in [7, 9] and (iv) the proposed Margin-based strategy for Taxonomy Adaptation (MTA). We do not compare with our previous work on adaptive classifier selection in large-scale hierarchical classification [8], since the accuracy results using that approach were marginally better than FH method. We use Liblinear to train the models for L2-regularized L2-loss support vector classification. In order to maintain consistency, the value of the penalty parameter C was fixed to 1, for all the four methods.

As shown in Figure 3, the proposed method MTA achieves comparable or better accuracy as compared to the entire layer flattening techniques, MLF and TLF. The s-test [11] ($p < 0.001$) showed statistical differences of MTA over FH and TLF. Table 2 presents the comparison on **DMOZ-1** dataset for training time (including re-training), model sizes and prediction speed. Clearly, since the proposed MTA method preserves the overall hierarchical structure of the taxonomy it achieves better values for these metrics of practical significance.

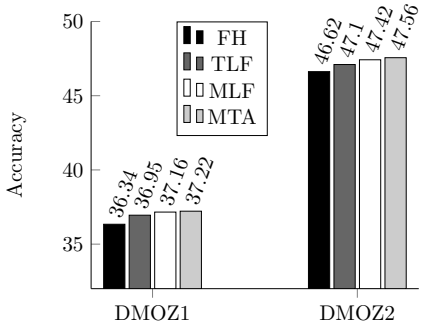

Fig. 3. Accuracy of algorithms for each dataset.

Properties	**MLF**	**MTA**
Tr. Time (in hours)	6.3	3.2
Tr. Model Size (in GB)	7.8	4.2
Pred. Time (in mins)	55	17

Table 2. Comparison of MLF and MTA Training Time, Model size on hard-disk and Prediction Time for the DMOZ-1 dataset

5 Conclusion

We presented a principled method for automatically adapting the given hierarchy of classes, in large scale hierarchical classification, to output a new hierarchy which leads to better generalization. The proposed approach is backed by well-founded theoretical insights and exploits the margin information to identify those decision nodes in the hierarchy which correspond to relatively harder classification problems and removing those nodes to minimize the impact of propagation of error. Not only does it lead to comparable or better accuracy, but enjoys favourable training and prediction speed. This approach can be viewed as an instance of a more general paradigm of making the two parts of the input in a supervised learning problem more compatible, towards achieving the common goal.

References

1. P. N. Bennett and N. Nguyen. Refined experts: improving classification in large taxonomies. In *In Proc. 32nd Int'l ACM SIGIR Conf. on Research and Dev. in Info. Retr.*, SIGIR 2009, pages 11–18.
2. L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *CIKM*, pages 78–87. ACM, 2004.
3. O. Dekel. Distribution-calibrated hierarchical classification. In *Advances in Neural Information Processing Systems*, pages 450–458, 2009.
4. O. Dekel, J. Keshet, and Y. Singer. Large margin hierarchical classification. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 27–34, 2004.
5. T. Gao and D. Koller. Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2072–2079, 2011.
6. T.-Y. Liu, Y. Yang, H. Wan, H.-J. Zeng, Z. Chen, and W.-Y. Ma. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explor. Newsl.*, pages 36–43, 2005.
7. H. Malik. Improving hierarchical svms by hierarchy flattening and lazy classification. In *1st Pascal Workshop on Large Scale Hierarchical Classification*, 2009.
8. I. Partalas, R. Babbar, É. Gaussier, and C. Amblard. Adaptive classifier selection in large-scale hierarchical classification. In *ICONIP (3)*, pages 612–619, 2012.
9. X. Wang and B.-L. Lu. Flatten hierarchies for large-scale hierarchical text categorization. In *Fifth IEEE International Conference on Digital Information Management*, pages 139–144, 2010.
10. G.-R. Xue, D. Xing, Q. Yang, and Y. Yu. Deep classification in large-scale text hierarchies. In *In Proc. 31st Int'l ACM SIGIR Conf. on Research and Dev. in Info. Retr.*, SIGIR '08, pages 619–626. ACM.
11. Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the* 22^{nd} *annual International ACM SIGIR conference*, pages 42–49. ACM, 1999.