

Improving Hierarchical Classification with Partial Labels

Nam Nguyen
Department of Computer Science
Cornell University, USA

Abstract. In this paper, we address the problem of semi-supervised hierarchical learning when some cases are fully labeled while other cases are only partially labeled, named *Hierarchical Partial Labels*. Given a label hierarchy, a fully labeled example provides a path from the root node to a leaf node while a partially labeled example only provides a path from the root node to an internal node. We introduce a discriminative learning approach, called Partial HSVM, that incorporates partially labeled information into the hierarchical maximum margin-based learning framework. The partially labeled hierarchical learning problem is formulated as a quadratic optimization that minimizes the empirical risk with L2-norm regularization. We also present an efficient algorithm for the hierarchical classification in the presence of partially labeled information. In our experiments with the WIPO-alpha patent collection, we compare our proposed algorithm with two other baseline approaches: Binary HSVM, a standard approach to hierarchical classification, which builds a binary classifier (SVM) at each node in the hierarchy, and PL-SVM, a flat multiclass classifier which can take advantages of the partial label information. Our empirical results show that Partial HSVM outperforms Binary HSVM and PL-SVM across different performance metrics. The experimental results demonstrate that our proposed algorithm, Partial HSVM, combines the strength of both methods, the Binary HSVM and PL-SVM, since it utilizes both the hierarchical information and the partially labeled examples. In addition, we observe the positive correlation between the labeling effort in obtaining partially labeled data and the improvement in performance.

1 Introduction

While fully labeled data is usually expensive, time consuming to collect, and sometimes requires human domain experts to annotate, partially labeled data is often relatively easier to obtain. Much attention in the machine learning community has been focused on integrating partially labeled data that are complementary to the fully labeled training data into existing learning framework. Partially labeled training data such as pairwise constraints [14, 15] and partial-label set [8] have been shown to improve performance of classification. In this paper, we investigate the usefulness of partially labeled information in the framework of hierarchical classification.

The problem of hierarchical classification has been extensively studied in recent years [1, 4, 2, 3, 7, 9, 11, 5]. In hierarchical classification, examples are classified into different paths of a given label hierarchy. Traditionally, a fully labeled example is labeled by a path from the root node to a leaf node in the label hierarchy. Here, a partially labeled example is labeled by a path from the root node to an internal node. In Figure 1, we demonstrate the difference between a fully labeled example and a partially labeled example. Unlike fully

labeled data that would require users to have extensive prior knowledge or experience with a data set, partially labeled examples relatively require often less effort from users. For example, in the task of hierarchical text classification, it is relatively easier for users to determine if a document belongs a science category versus politics or sport categories than to classify into different subcategories of science such as chemistry, physics, etc. Specifically, in web taxonomies such as ODP and Yahoo! which contain hundreds of thousands of categories, much of the web content does not have the full path label [6]. We will show that partially labeled data in the framework of hierarchical classification helps producing better classifiers without too much labeling annotation from users.

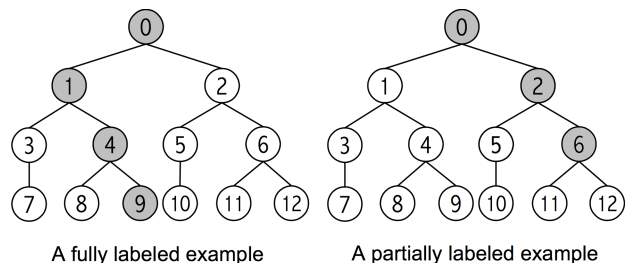


Figure 1. Examples of fully labeled and partially labeled hierarchical data.

In this work, we propose a discriminative learning approach which incorporates partially labeled information into the hierarchical margin-based learning framework, called Partial HSVM. In our experiment with the WIPO-alpha patent collection [13], we show that Partial HSVM outperforms both Binary HSVM, a standard method for hierarchical classification, which builds a binary SVM classifier at each node of the hierarchy; and PL-SVM, a flat multiclass classifier which can take advantages of the partial label information. In addition, we also show the trade-off between the labeling effort of obtaining partially labeled data and the improvement in performance of our proposed method. The paper is structured as follow: in section 2, we describe in detail the novel partially-labeled hierarchical classification algorithm; in section 3 we review related work on supervised learning with partially labeled data and on hierarchical classification; the experimental results and conclusion are given in section 4 and 5, respectively.

2 Partially Labeled Hierarchical Classification

In this section, we address the problem of hierarchical learning when there are additional partially labeled data augmented with fully labeled data. In the hierarchical classification problem, without loss of generality we assume that the label hierarchy is a equal-depth tree,

i.e. every leaf node has the same depth. Here, we introduce some notations that are used throughout the paper. Given the label hierarchy $\mathcal{H}$, the label of an example is denoted as $\vec{y}$ which indicates the unique path from the root node to the y node. In Figure 1, the label of the fully labeled example is denoted as $\vec{9}$ representing the path $[0 \rightarrow 1 \rightarrow 4 \rightarrow 9]$. Similarly, $\vec{6}$, $[0 \rightarrow 2 \rightarrow 6]$, is the label of the partially labeled example. We denote $\mathcal{H}^{\text{leaf}}$ and $\mathcal{H}^{\text{internal}}$ as the set of label paths from the root node to a leaf node and from the root node to an internal node, respectively. In addition, we first define

$$\text{ED}(\vec{y}) = \left\{ \vec{y}' \mid \text{Depth}(\vec{y}') = \text{Depth}(\vec{y}) \right\},$$

to be the set of label paths in the label hierarchy that have the same depth as $\vec{y}$. For example, in Figure 1 $\text{ED}(\vec{6}) = \{\vec{3}, \vec{4}, \vec{5}, \vec{6}\}$. Second, we define $\text{SubPath}(\vec{y}) = \left\{ \vec{y}' \mid \vec{y}' \subseteq \vec{y} \right\}$, to be the set of sub-paths of $\vec{y}$. For example, in Figure 1 $\text{SubPath}(\vec{9}) = \{\vec{1}, \vec{4}, \vec{9}\}$. Third, we define

$$\begin{aligned} \text{LeafDescendant}(\vec{y}) &= \left\{ \vec{y}' \mid \vec{y}' \text{ is a leaf node} \right. \\ &\quad \left. \text{and a descendant of } \vec{y} \right\}, \\ \neg \text{LeafDescendant}(\vec{y}) &= \left\{ \vec{y}' \mid \vec{y}' \text{ is a leaf node} \right. \\ &\quad \left. \text{and not a descendant of } \vec{y} \right\}, \end{aligned}$$

to be the set of leaf nodes that are descendants of $\vec{y}$ and that are not descendants of $\vec{y}$, respectively. For example, in Figure 1 $\text{LeafDescendant}(\vec{6}) = \{\vec{11}, \vec{12}\}$ and $\neg \text{LeafDescendant}(\vec{6}) = \{\vec{7}, \vec{8}, \vec{9}, \vec{10}\}$.

In the hierarchical supervised setting, a learning algorithm takes a set of fully labeled training examples, $\mathbf{L} = \{(x_i, \vec{y}_i)\}_{i=1}^n$ as input, where $x_i \in \mathcal{X}$ and the corresponding label $\vec{y}_i \in \mathcal{H}^{\text{leaf}}$ is a path from the root node to the leaf node y_i in the label hierarchy $\mathcal{H}$. In addition to the fully labeled examples we also have a set of partially labeled training data, $\mathbf{PL} = \{(x_j, \vec{y}_j)\}_{j=n+1}^{n+m}$, where $x_j \in \mathcal{X}$ and the corresponding label $\vec{y}_j \in \mathcal{H}^{\text{internal}}$ from the root node to an internal node y_j . The goal of hierarchical classification is to form a hypothesis $h: \mathcal{X} \mapsto \mathcal{H}^{\text{leaf}}$ which maps an input $x \in \mathcal{X}$ to a label path $\vec{y} \in \mathcal{H}^{\text{leaf}}$. Consider a mapping $\Phi: \mathcal{X} \times \mathcal{H} \mapsto \mathcal{F}$ which projects each example-label pair $(x, \vec{y}) \in \mathcal{X} \times \mathcal{H}$ to $\Phi(x, \vec{y})$ in a new space $\mathcal{F}$ which is defined as

$$\Phi(x, \vec{y}) = \begin{bmatrix} x \cdot \mathcal{I}(0 \in \vec{y}) \\ \vdots \\ x \cdot \mathcal{I}(i \in \vec{y}) \\ \vdots \\ x \cdot \mathcal{I}(|\mathcal{H}| \in \vec{y}) \end{bmatrix},$$

where $\mathcal{I}(\cdot)$ is the indicator function to decide whether a node belongs to a path and $|\mathcal{H}|$ is the number of nodes in the label hierarchy.

Many machine learning algorithms are formulated to minimize the regularized empirical risk, $\min_w R_{\text{reg}}(w) := \lambda \Omega(w) + L(w)$, where $\Omega(w)$ is a convex and monotonically increasing function which serves as a regularizer with a regularization constant $\lambda > 0$; and $L(w)$ is the empirical risk which measures the inconsistency between the correct labels and the predicted labels arising from using the weight parameter w .

We can obtain the partially labeled hierarchical margin-based algorithm, Partial HSVM, by considering the situation where we use the L2-norm regularization,

$$\Omega(w) = \frac{1}{2} \|w\|^2,$$

and the empirical risk is composed of two components, $L(w) = L_1(w) + L_2(w)$. The first component $L_1(w)$ measures the risk of predicting a different label from the correct one,

$$L_1(w) = \frac{1}{n+m} \sum_{i=1}^{n+m} \sum_{\vec{y}'_i \in \text{SubPath}(\vec{y}_i)} l_1(x_i, \vec{y}'_i, w),$$

where $l_1(x_i, \vec{y}'_i, w) =$

$$\max_{\vec{y} \in \text{ED}(\vec{y}'_i)} \left(\Delta(\vec{y}'_i, \vec{y}) - w^T [\Phi(x_i, \vec{y}'_i) - \Phi(x_i, \vec{y})] \right),$$

and $\Delta(\vec{y}'_i, \vec{y})$ is the cost of predicting the label path $\vec{y}$ instead of the correct label path $\vec{y}'_i$. In this learning framework, $\Delta(\cdot, \cdot)$ is defined as the number of levels needed to move up the label hierarchy until a common node of the two paths is reached, e.g. $\Delta(\vec{9}, \vec{8}) = 1$ and $\Delta(\vec{9}, \vec{7}) = 2$, referred to the label hierarchy in Figure 1. For each example $(x_i, \vec{y}_i) \in \mathbf{L} \cup \mathbf{PL}$, $L_1(w)$ not only penalizes for predicting a different label from $\vec{y}_i$ but also for predicting a different label from any sub-path of the correct label $\vec{y}_i \in \text{SubPath}(\vec{y}_i)$. This helps to utilize both the information in the fully labeled and partially labeled examples.

In addition, for each partially labeled example $(x_j, \vec{y}_j) \in \mathbf{PL}$ the second component $L_2(w)$ measures the cost of predicting a label that is not a leaf descendant of $\vec{y}_j$,

$$L_2(w) = \frac{1}{m} \sum_{j=1}^m \max_{\vec{y}'_j \in \mathcal{H}^{\text{leaf}}} \left(\Delta(\vec{y}'_j, \vec{y}_j) - w^T [\Phi(x_j, \vec{y}'_j) - \Phi(x_j, \vec{y}_j)] \right), \quad (1)$$

where $\vec{y}'_j = \arg \max_{\vec{y}'_j \in \text{LeafDescendant}(\vec{y}_j)} w^T \Phi(x_j, \vec{y}'_j)$.

In particular, for each partially labeled example $(x_j, \vec{y}_j)$, $L_2(w)$ ensures that the score associated with the most suitable label path, $w^T \Phi(x_j, \vec{y}'_j)$, among all leaf descendants of $\vec{y}_j$ to be greater than the scores associated with any other leaf nodes, $w^T \Phi(x_j, \vec{y})$, by a cost of $\Delta(\vec{y}'_j, \vec{y})$. For example, in Figure 1 a cost will incur if the predicted label for the partially label example is not $\vec{11}$ or $\vec{12}$.

Specifically, the partially labeled hierarchical classification (Partial HSVM) learns a weight vector w and slack variables ξ and ξ^* via the following quadratic optimization problem:

OPTIMIZATION PROBLEM: PARTIAL HSVM

$$\min_{w, \xi \geq 0, \xi^* \geq 0} : \frac{\lambda}{2} \|w\|^2 + \frac{1}{n+m} \sum_{i=1}^{n+m} \sum_{\vec{y}'_i \in \text{SubPath}(\vec{y}_i)} \xi_{i, \vec{y}'_i} + \frac{1}{m} \sum_{j=1}^m \xi_j^* \quad (2)$$

subject to:

$$\forall (x_i, \vec{y}_i) \in \mathbf{L} \cup \mathbf{PL}, \forall \vec{y}'_i \in \text{SubPath}(\vec{y}_i) :$$

$$\max_{\vec{y} \in \text{ED}(\vec{y}'_i)} \left(\Delta(\vec{y}'_i, \vec{y}) - w^T [\Phi(x_i, \vec{y}'_i) - \Phi(x_i, \vec{y})] \right) \leq \xi_{i, \vec{y}'_i}, \quad (3)$$

$$\forall (x_j, \vec{y}_j) \in \mathbf{PL} :$$

$$\max_{\vec{y}'_j \in \mathcal{H}^{\text{leaf}}} \left(\Delta(\vec{y}'_j, \vec{y}_j) - w^T [\Phi(x_j, \vec{y}'_j) - \Phi(x_j, \vec{y}_j)] \right) \leq \xi_j^*, \quad (4)$$

where $\vec{y}'_j = \arg \max_{\vec{y}'_j \in \text{LeafDescendant}(\vec{y}_j)} w^T \Phi(x_j, \vec{y}'_j)$.

After we have learned w, ξ and ξ^* , the classification of a test example x is done by

$$h(x) = \operatorname{argmax}_{\vec{y} \in \mathcal{H}^{\text{leaf}}} w^T \Phi(x, \vec{y}).$$

In this partially labeled hierarchical learning setting, the average depth of the partially labeled paths, $\frac{1}{m} \sum_{j=n+1}^{n+m} \text{Depth}(\vec{y}_j)$, provides a rough indication of the amount of labeled information given to the learning algorithm. In the limit, if $\text{Depth}(\vec{y}_j) = \text{Depth}(\mathcal{H})$ then we have the hierarchical supervised learning framework where each training example is given the fully-labeled path. Moreover, if $\text{Depth}(\vec{y}_j) = 0$ then we obtain the hierarchical semi-supervised learning framework where there is additional unlabeled data augmented with the fully labeled data. We will show later in the experiment section, how the hierarchical classification performance changes as the average depth of the partially labeled paths varies.

Algorithm 1 : Partially Labeled Hierarchical SVM (Partial HSVM)

Input: $\mathbf{L}$ - the fully labeled data
 $\mathbf{PL}$ - the partially labeled data
 λ - a parameter of the QP
 ϵ - a tolerance for stopping condition

Initialize w_1 such that $\|w_1\| \leq 1/\sqrt{\lambda}$

Set $t = 1$

repeat

for each $(x_i, \vec{y}_i) \in \mathbf{L} \cup \mathbf{PL}$

Set $\eta_t = \frac{1}{\lambda t}$

Set $w_{t+1/2} = (1 - \eta_t \lambda) w_t$

// Find violations for constraints in (3)

for each $\vec{y}_i \in \text{SubPath}(\vec{y}_i)$

if $\{ \max_{\vec{y} \in \text{ED}(\vec{y}_i)} (\Delta(\vec{y}_i, \vec{y}) - w_t^T [\Phi(x_i, \vec{y}_i) - \Phi(x_i, \vec{y})]) > 0 \}$

Set $w_{t+1/2} = w_{t+1/2} + \frac{\eta_t}{n+m} [\Phi(x_i, \vec{y}_i) - \Phi(x_i, \vec{y})]$

end if

end for

// Find violations for constraints in (4)

for each $(x_i, \vec{y}_i) \in \mathbf{PL}$

Set $\vec{y}_i = \operatorname{argmax}_{\vec{y}_i \in \text{LeafDescendant}(\vec{y}_i)} w_t^T \Phi(x_i, \vec{y}_i)$

if $\{ \max_{\vec{y} \in \mathcal{H}^{\text{leaf}}} (\Delta(\vec{y}_i, \vec{y}) - w_t^T [\Phi(x_i, \vec{y}_i) - \Phi(x_i, \vec{y})]) > 0 \}$

Set $w_{t+1/2} = w_{t+1/2} + \frac{\eta_t}{m} [\Phi(x_i, \vec{y}_i) - \Phi(x_i, \vec{y})]$

end if

end for

// Project w to the ball of radius $1/\sqrt{\lambda}$

Set $w_{t+1} = \min \left\{ 1, \frac{1/\sqrt{\lambda}}{\|w_{t+1/2}\|} \right\} w_{t+1/2}$

Set $t = t + 1$

end for

until $(R_{\text{reg}}(w_{t-1}) - R_{\text{reg}}(w_t)) < \epsilon$

Output: w_t

Due to the constraints in (4), the proposed optimization is a non-convex quadratic programming(QP). In order to solve the QP, we

employ the stochastic gradient descent approach which has shown to be very efficient and does not require transforming to the dual formulation [12, 10]. Similar to [10], we restrict the search space to the sphere of radius $1/\sqrt{\lambda}$. The algorithm alternates between gradient descent steps and projection steps until reduction of the regularized risk objective function is less than a pre-specified tolerance, ϵ . In each iteration, the algorithm first finds the violated constraints and updates the weight vector w accordingly. Then the weight vector w is projected to the sphere of radius $1/\sqrt{\lambda}$. The details of the Partial HSVM is given in Algorithm 1.

We observed that if $w_1 = 0$ then w_t can be written as

$$w_t = \sum_{(x, \vec{y})} \varphi_{x, \vec{y}} \Phi(x, \vec{y}).$$

Hence, we can incorporate the usage of kernel when computing inner product operations, i.e.:

$$\begin{aligned} \langle w, \Phi(x', \vec{y}') \rangle &= \sum_{(x, \vec{y})} \varphi_{x, \vec{y}} \mathbf{K}(x, \vec{y}, x', \vec{y}') \\ \|w\|^2 &= \sum_{(x, \vec{y})} \sum_{(x', \vec{y}')} \varphi_{x, \vec{y}} \varphi_{x', \vec{y}'} \mathbf{K}(x, \vec{y}, x', \vec{y}') \end{aligned}$$

3 Related Work

In supervised learning, partially labeled data in the form of pairwise constraints have been shown to improve performance of classifiers. In [15, 14], the authors proposed a discriminative learning framework which can simultaneously learn the fully labeled data and pairwise constraints. Recently, another source of partially labeled data called partial-label set was investigated by [8]. Partial-label set contains several possible labels for each training example, only one of which is the correct label. In [8], the authors also proposed a margin-based approach that utilizes both the partial-label and fully labeled examples to learn a more accurate classifier, PL-SVM. In the partially labeled hierarchical learning framework, we can treat each partially labeled path as a set of possible leaf nodes that are descendants of the labeled internal node. Hence, in our experiment we compare performance of the proposed algorithm, Partial HSVM, with PL-SVM.

While a large body of work exists on hierarchical classification [1, 4, 2, 3, 7, 9, 11, 5, 6], here we simply highlight some of the more relevant work. In [6], the authors evaluated performance of the Binary HSVM in web-page classification over the hierarchy of the Yahoo! categories. The authors also pointed out the difficulties in applying text categorization algorithms on Web taxonomies and the need for improvement in hierarchical categorization. Since the Binary HSVM is able to learn from partially labeled hierarchical data, we compare performance of this algorithm with the proposed algorithm, Partial HSVM, in our experiments. In addition, [1] also proposed a margin-based learning algorithm for hierarchical categorization, called HSVM. In both HSVM and Partial HSVM, the optimizations penalize for predicting a different leaf category from the correct one. However, HSVM is not able to learn from the additional partially labeled data. Our proposed algorithm, Partial HSVM, can be viewed as an extension from HSVM with additional constraints for the partially labeled examples.

4 Experiments

In this section, we perform empirical comparison of our proposed algorithm Partial HSVM against Binary HSVM. In addition, we also

show the trade-off between the labeling effort of obtaining partially labeled data and the improvement in performance of Partial HSVM. By the time of the conference, a software package implementing the Partial HSVM algorithm will be available for download.

We evaluate our proposed algorithm Partial HSVM on the WIPO-alpha patent collection [13]. The patent documents in the collection are classified according to a standard hierarchy known as International Patent Classification (IPC). In this data set, each document is belonged to exactly one leaf node. IPC categories are organized in a four-level hierarchy, i.e. sections, classes, subclasses and groups. There are eight sections in the hierarchy namely A, B, C, D, E, F, G, and H. In our experiment, we treat each section as a separate data set. Hence, the label hierarchy of each section has depth of 3. A summary of the data set is given in Table 1. For each section, we randomly divide the data set into 3 folds: one used as the fully labeled training examples, one used as the partially labeled training examples, and the last one used as the test examples. The input data to the learning algorithms is represented as unit-norm tfidf bag-of-words format. The process is repeated 10 times and the averaged performance is reported. For all experiments, we set the regularization constant λ by two-fold cross validation and use the linear kernel.

Table 1. A summary of the WIPO-alpha data set.

Section	No. Nodes ($ \mathcal{H} $)	No. Documents
A	385	9654
B	611	12186
C	427	14446
D	91	1429
E	143	2666
F	332	5259
G	237	8879
H	225	10564

To evaluate performance of Partial HSVM, we employ three different performance measures: macro F1, micro F1, and hierarchy loss. The F1 measure is the harmonic mean of precision and recall where $\text{Precision} = \frac{\text{True Positives}}{\text{Predicted Positives}}$ and $\text{Recall} = \frac{\text{True Positives}}{\text{Actual Positives}}$. As standardly reported, macro F1 weights the F1 for each class equally and micro F1 weights each example equally. Since categories that are closer the leaves have less examples, we observed that macro F1 tends to emphasize these categories while micro F1 tends to emphasize categories with more examples (i.e. closer to the root node). Moreover, hierarchy loss measures the averaged loss of the predicted label path and the correct label path which takes the interdependency between categories into account,

$$\text{Hierarchy Loss} = \frac{1}{T} \sum_{i=1}^T \Delta(\vec{y}_i, h(x_i)),$$

where T is the number of test examples.

In our experiments, we compare the classification performance of Partial HSVM against the two baseline algorithms: Binary HSVM and PL-SVM. Similar to Partial HSVM, Binary HSVM and PL-SVM are able to learn from the additional partially labeled data. For the Binary HSVM, at each category of the hierarchy a binary SVM classifier is trained to decide whether an example belongs to this category or not. The training data for each category is all examples that belongs to the parent category. At testing time, when an example is classified as a member of a category, classification continues into the sub-hierarchy beneath the category until no more positive predictions are made. For the PL-SVM, a single flat multiclass classifier is trained on both the fully labeled examples and the partially labeled examples. In Figure 2, we plot performance of Partial HSVM

in comparison to Binary HSVM and PL-SVM across three different performance metrics (one for each row) and across three different partial depths (one for each column). The partial depth indicates the label path depths of the partially labeled examples. Note when the label path depths equal to 3, we have fully labeled examples. We observe that across different performance metrics and different partial depths, Partial HSVM outperforms Binary HSVM and PL-SVM. In addition, we also report the over average performance of Partial HSVM, Binary HSVM, and PL-SVM across different data sections in Table 2. The differences in the macro F1, micro F1, and hierarchy loss of two methods are statistically significant at the $\alpha = 0.05$ level according to a two-sided t-test. Hence, Partial HSVM is able to better learn from additional partially labeled examples in comparison with Binary HSVM and PL-SVM.

We also investigate the trade-off between the labeling effort of obtaining partially labeled data, i.e. different levels of partial depths, and the improvement in performance of our proposed method, Partial HSVM. In Figure 3, we plot the performance of Partial HSVM at different levels of partial depths. The differences in performance of all three levels of partial depths are also statistically significant at the $\alpha = 0.05$ level according to a two-sided t-test. We observe a common trend across different data sections and different performance metrics that as the partial depth increases, i.e. the more effort is put into obtaining partially labeled data, the performance of Partial HSVM also increases. Therefore, in the data labeling process we should account for the trade-off between the labeling effort and the improvement in performance.

5 Conclusion

In this paper, we address the problem of hierarchical learning when some cases are fully labeled while other cases are only partially labeled. Given a label hierarchy, a fully labeled example provides a path from the root node to a leaf node while a partially labeled example only provides a path from the root node to an internal node. We formulate the partially-labeled hierarchical learning problem as a quadratic optimization minimizing the L2-norm regularized empirical risk and we also present an efficient algorithm for hierarchical classification in the presence of partially labeled information. Experiments with the WIPO-alpha patent collection (a text data set) show that partially-labeled hierarchical information along with fully labeled data improves the performance of classification. In addition, we also show that Partial HSVM is able to outperform the two state-of-art baseline methods: Binary HSVM and PL-SVM. Finally, we demonstrate the trade-off between the labeling effort of obtaining partially labeled data and the improvement in performance of our proposed method, Partial HSVM.

REFERENCES

- [1] Lijuan Cai and Thomas Hofmann, ‘Hierarchical document categorization with support vector machines’, in *ACM Conference on Information and Knowledge Management*, (2004).
- [2] Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni, ‘Incremental algorithms for hierarchical classification’, *Journal of Machine Learning Research*, **7**, 31–54, (2006).
- [3] Ofer Dekel, Joseph Keshet, and Yoram Singer, ‘Large margin hierarchical classification’, in *Proceedings of the 21st International Conference on Machine Learning*, pp. 209–216, (2004).
- [4] Susan Dumais and Hao Chen, ‘Hierarchical classification of web content’, in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 256–263, (2000).

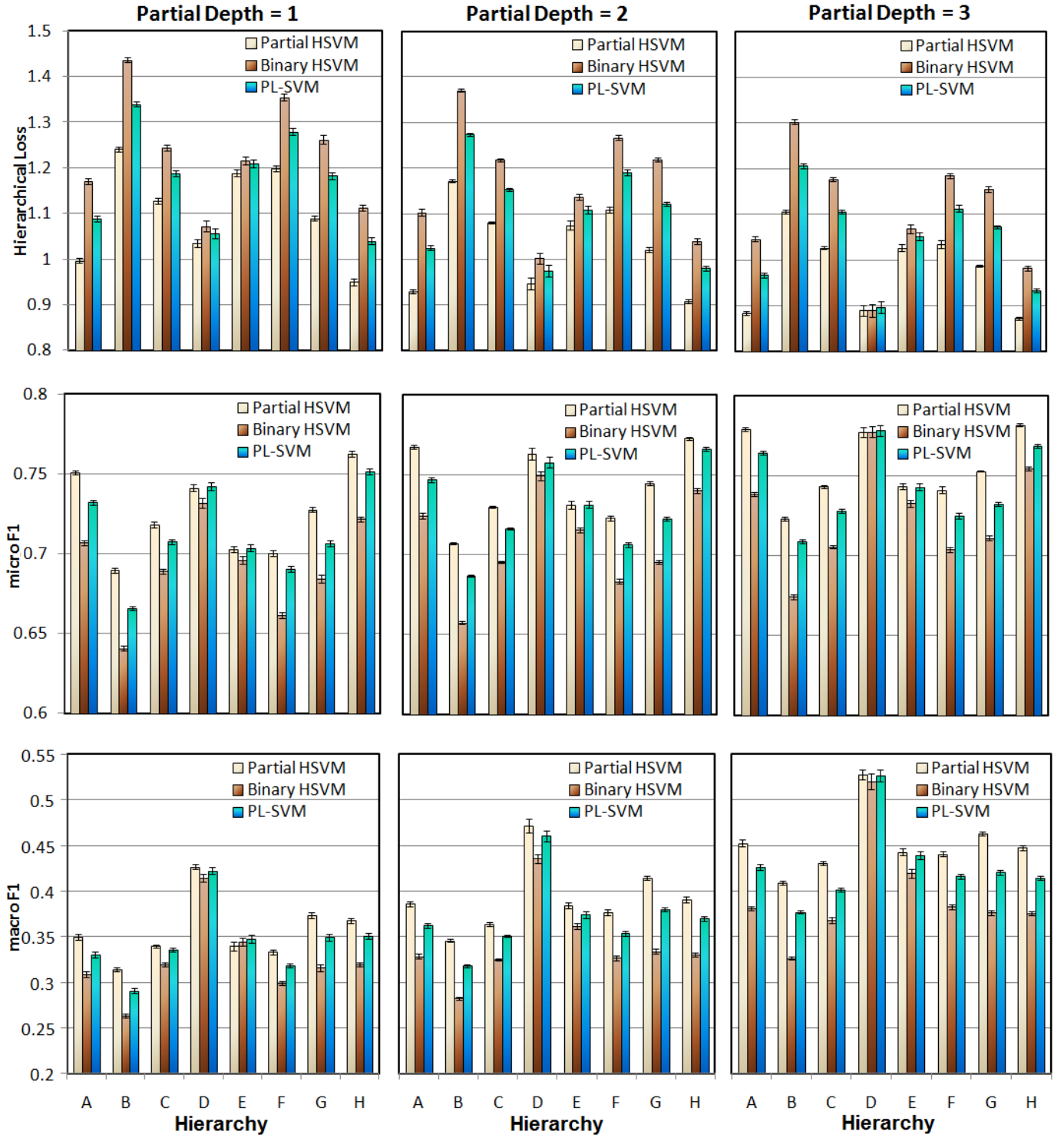


Figure 2. Performance of Partial HSVM versus Binary HSVM and PL-SVM across three different evaluation metrics: hierarchical loss, macroF1 and microF1.

- [5] Michael I Jordan and Robert A Jacobs, 'Hierarchical mixtures of experts and the EM algorithm', *Neural Computation*, 6(2), 181–214, (1998).
- [6] Tie-Yan Liu, Yiming Yang, Hao Wan, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma, 'Support vector machines classification with a very large-scale taxonomy', *SIGKDD Explorations Newsletter*, 7(1), 36–43, (2005).
- [7] Andrew McCallum, Ronald Rosenfeld, Tom Mitchell, and Andrew Y. Ng, 'Improving text classification by shrinkage in a hierarchy of classes', in *International Conference on Machine Learning*, pp. 359–367, (1998).
- [8] Nam Nguyen and Rich Caruana, 'Classification with partial labels', in *SIGKDD Conference on Knowledge Discovery and Data Mining*, (2008).

Table 2. Overall average performance of Partial HSVM, Binary HSVM and PL-SVM.

Performance Metrics	Partial Depth	Binary HSVM	PL-SVM	Partial HSVM (% Improvement over Binary HSVM/PL-SVM)
macro F1	1	0.323	0.345	0.356 (10.0/3.06)
	2	0.341	0.371	0.392 (15.0/5.58)
	3	0.394	0.428	0.452 (14.7/5.60)
micro F1	1	0.692	0.713	0.724 (4.72/1.60)
	2	0.708	0.730	0.743 (4.91/1.77)
	3	0.725	0.744	0.755 (4.22/1.49)
Hierarchy Loss	1	1.234	1.173	1.103 (10.6/5.97)
	2	1.169	1.104	1.030 (11.9/6.67)
	3	1.101	1.045	0.978 (11.1/6.37)

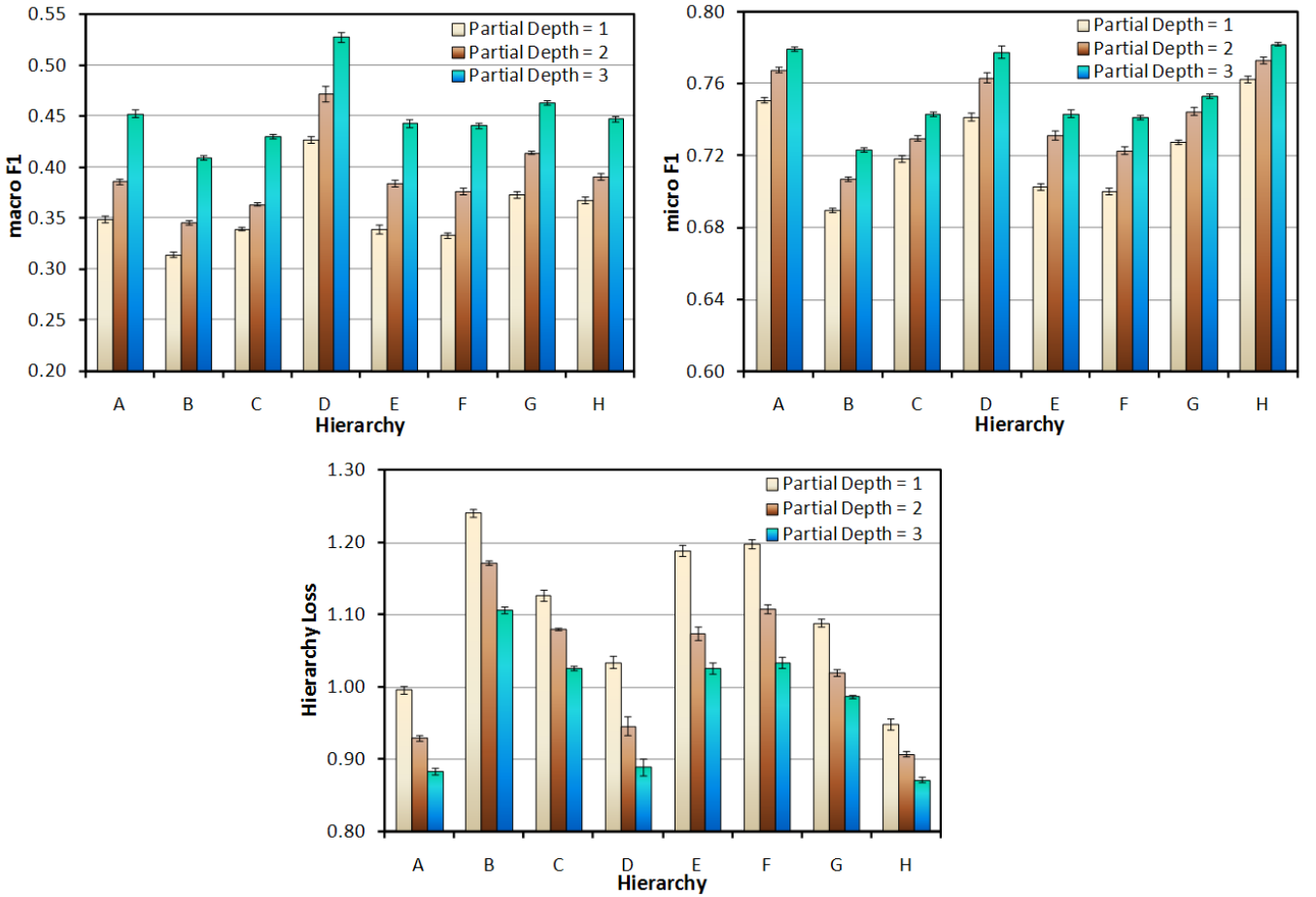


Figure 3. Performance of Partial HSVM at different partial depths of the partially labeled training data across three evaluation metrics: hierarchical loss, macroF1 and microF1.

- [9] Miguel E. Ruiz and Padmini Srinivasan, ‘Hierarchical text categorization using neural networks’, *Information Retrieval*, **5**(1), 87–118, (2002).
- [10] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro, ‘Pegasos: Primal estimated sub-gradient solver for svm’, in *Proceedings of the 24th International Conference on Machine Learning*, pp. 807–814, New York, NY, USA, (2007). ACM.
- [11] Aixin Sun and Ee-Peng Lim, ‘Hierarchical text classification and evaluation’, in *International Conference in Data Mining*, pp. 521–528, (2001).
- [12] S. V. N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin P. Murphy, ‘Accelerated training of conditional random fields with stochastic gradient methods’, in *ICML ’06: Proceedings of the 23rd international conference on Machine learning*, pp. 969–976, (2006).
- [13] World Intellectual Property Organization. Wipo-alpha dataset. URL: <http://www.wipo.int/ibis/datasets>, 2003.
- [14] Rong Yan, Jian Zhang, Jie Yang, and Alexander G. Hauptmann, ‘A discriminative learning framework with pairwise constraints for video object classification’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(4), 578–593, (2006).
- [15] Jian Zhang and Rong Yan, ‘On the value of pairwise constraints in classification and consistency’, in *Proceedings of the 24th International Conference on Machine Learning*, pp. 1111–1118, (2007).