Learning Classifiers Using Hierarchically Structured Class Taxonomies

Feihong Wu, Jun Zhang, and Vasant Honavar

Artificial Intelligence Research Laboratory,
Department of Computer Science, Iowa State University,
Ames, Iowa 50011-1040, USA
{wuflyh, jzhang, honavar}@cs.iastate.edu

Abstract. We consider classification problems in which the class labels are organized into an abstraction hierarchy in the form of a class taxonomy. We define a structured label classification problem. We explore two approaches for learning classifiers in such a setting. We also develop a class of performance measures for evaluating the resulting classifiers. We present preliminary results that demonstrate the promise of the proposed approaches.

1 Introduction

Machine learning algorithms to design of pattern classifiers have been well studied in the literature. Most such algorithms operate under the assumption that the the class labels are mutually exclusive. However, many real world problems present more complex classification scenarios. For instance, in computer vision application, natural scene containing multiple objects can be assigned to multiple categories [3]; in a digital library application, a text document can be assigned to multiple topics organized into a topic hierarchy; in bioinformatics, an ORF may have several functions [5]. In each of these cases, the class labels are naturally organized in the form of a hierarchically structured class taxonomy which defines an abstraction over class labels. Such a classification scenario presents two main challenges: (1) The large number of class label combinations make it hard to reliably learn accurate classifiers from relatively sparse data sets. (2) Standard metrics for evaluating classifiers in settings where class labels are mutually exclusive are not suitable for evaluation of classifiers in settings where the class labels are organized into a class hierarchy. Despite recent attempts to address some of these problems, [1, 2, 3, 4, 5, 6, 7], at present, a general solution is still lacking. Against this background, we explore approaches to learning classifiers in the presence of class taxonomies. The paper is organized as follows. Section 2 presents a precise formulation of the single label, multi label and the structured label classification problems; Section 3 describes two approaches to learning classifiers from data in the presence of class taxonomies; Section 4 explores performance measures for evaluating the resulting classifiers; Section 5, briefly describes results of experiments using the Reuters-21578 [8] data and genotype data [5]; Section 6 concludes with a summary and discussion.

J.-D. Zucker and L. Saitta (Eds.): SARA 2005, LNAI 3607, pp. 313–320, 2005.

2 Preliminaries

Many standard classifier learning algorithms normally make the basic assumption of single label instances. That is, each instance that is represented by an ordered set of attributes $\mathbf{A} = \{A_1, A_2, ..., A_N\}$ can belong to one and only one class from a set of classes $\mathbf{C} = \{c_1, c_2, ..., c_M\}$. Therefore, class labels in $\mathbf{C}$ are mutually exclusive.

In multi label classification settings, class labels are not mutually exclusive. Each instance can be labelled using a subset of labels $c_s \subset \mathbf{C}$, where $\mathbf{C} = \{c_1, c_2, ..., c_M\}$ is a finite set of possible classes. If instances can be labelled with arbitrary subsets of $\mathbf{C}$, the total number of possible multi label combinations is 2^M.

An even more complex classification scenario is one in which instances to be classified are assigned labels from a hierarchically structured class taxonomy. Here, we define class taxonomy first and then formalize the resulting structured label classification problem.

Definition 1 (Class Taxonomy). *A Class Taxonomy $\mathcal{CT}$ is a tree structured regular concept hierarchy defined over a partially order set $(\boldsymbol{C}_T, \prec)$, where $\boldsymbol{C}_T$ is a finite set that enumerates all class concepts in the application domain, and relation $\prec$ represents the is-a relationship that is both anti-reflective and transitive:*

- *The only one greatest element "ANY" is the root of the tree.*
- *$\forall c_i \in \boldsymbol{C}$, $c_i \prec c_i$ is false.*
- *$\forall c_i, c_j, c_k \in \boldsymbol{C}$, $c_i \prec c_j$ and $c_j \prec c_k$ imply $c_i \prec c_k$.*

A tree structured class taxonomy represents class memberships at different levels of abstraction. The root of a class taxonomy is the most general label (i.e., "ANY") that is applicable to any instance. The leaves of class taxonomy indicate the most specific labels. The tree structure imposes strict constraints on these class memberships. Therefore, when an instance is assigned a label l from a hierarchically structured class taxonomy, it is implicitly labelled with all the ancestors of the label l in the class taxonomy.

Definition 2 (Structured label). *Any structured label $\mathcal{C}_s$ is represented by a subtree of $\mathcal{CT}$. $\mathcal{C}_s$ is a partially order set $(\boldsymbol{C}_s, \prec)$ that defines the same* is-a *relationships as in $\mathcal{CT}$. $\forall c_i \in \boldsymbol{C}_s$, c_i is ANY or $c_i \prec parent(c_i)$, where $parent(c_i) \in \boldsymbol{C}_s$ is the immediate ancestor of c_i in $\mathcal{CT}$.*

A class taxonomy imposes constraints on the integrity and validity of the structured labels. The integrity constraint states that $\mathcal{C}_s$ is a subtree structure of $\mathcal{CT}$ sharing the same root: Structured label is not an arbitrary fragment of the class taxonomy. The validity constraint captures the *is-a* relationships among class labels within a class taxonomy. A structured label is invalid if it contains a label l but not the parents of l in a given class taxonomy.

3 Methods

3.1 Binarized Structured Label Learning

One simple approach is to build a classifier consisting of a set of binary classifiers (one for each class). However, the drawbacks of this approach are obvious: (1) When making predictions for unlabelled instances, the classification results may violate the integrity and validity constraints. (2) The set of binary classifiers fails to exploit potentially useful constraints provided by the class taxonomy during learning.

To overcome these disadvantages, we build a hierarchically organized collection of classifiers that mirrors the structure of the class taxonomy $\mathcal{CT}$. The resulting classifiers form a partially ordered set $(h_{\mathcal{CT}}, \prec)$, where $h_{\mathcal{CT}} = \{h_{C_1}, \cdots, h_{C_M}\}$ is the set of classifiers, and $\prec$ represents partial orders among classifiers. If C_j is a child of C_i in $\mathcal{CT}$, then the respective classifiers satisfy the partial order $h_{C_j} \prec h_{C_i}$. This partial order on classifiers guides the classification of an instance. If $h_{C_j} \prec h_{C_i}$, an instance will not be classified using h_{C_j} if it has been classified as not belonging to C_i (i.e., output of h_{C_j} is 0). We call our method of building such hierarchically structured classifiers "Binarized Structured Label Learning" (BSLL).

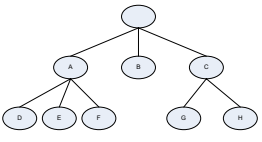

Fig. 1. Structure class taxonomy

3.2 Split-Based Structured Label Learning

A second approach to structured label learning is an adaptation of an approach to multi-label learning. We digress briefly to outline approaches to multi-label learning.

In real world applications it is very rare that each of the 2^M multi label combinations appear in the training data. The actual number of multi labels is much smaller than the possible number 2^M. Thus, we may set an upper limit on the number of possible class label combinations. If the number of labels that can occur in a multi-label is limited to 2, we will only consider the combinations of 2 class labels instead of M class labels. Another option is to consider only the multi labels that appear in the training data. In either case, we can not apply standard learning algorithms directly to the multi-label classification problem. This is because the multi label and the individual class labels are not mutually exclusive and it is not uncommon for some instances to be labelled with a single class label and others with multi labels. Because most standard learning algorithms assume mutually exclusive class labels, we will need to generate mutually exclusive classes. For example, consider $\mathbf{C} = \{A, B, C\}$

with instances set S_A, S_B, S_C respectively. Suppose the only multi label observed in the training data is $\{A, B\}$. Note that $S_A \bigcap S_B \neq \emptyset$. So the extended class label set is $\mathbf{C}' = \{\hat{A}, \hat{B}, \hat{C}, A\&B\}$,which represents instance set $S_A - S_A \bigcap S_B, S_B - S_A \bigcap S_B, S_C, S_A \bigcap S_B$.

This approach to transforming class labels to obtain mutually exclusive class labels can be applied to structured label learning problem by building split-based classifiers. We will first define a split in a class taxonomy $\mathcal{CT}$, and then for each split we show how to learn a respective classifier by learning from instances with extended label sets (as outlined above).

Definition 3 (Split). *A split is a one level subtree within a class taxonomy, which includes one parent node and all its children nodes, and the links between the parent node and children nodes.*

Obviously, the number of splits in the class taxonomy is smaller than the number of nodes. We can build a set of classifiers on the splits to solve structured label problem so to decrease the number of resulting classifiers. Within each split, the structured label problem will be reduced to a multi label problem, and we only need to consider the combinatorial extensions on class labels at that particular level. Additionally, the split-based classifiers are also partially ordered according to a given class taxonomy. Any instance to be classified will follow this topological order of the split-based classifiers: start from the classifier for the split at first position, continue to run a split-based classifier only when predicted to be "1" by the parent split-based classifier.

4 Performance Measure for Structured Label Classification

In single label classification, a loss function(like standard 0-1 loss function) $\mathbf{loss}(c_p, c_o)$ can be defined to evaluate the cost of misclassifying the instance with observed class label c_o to the predictive class label c_p. However, this approach is inadequate in a structured label problem in which there is a need to take into account the relationships between labels assigned to an instance. Here each label set corresponds to a subtree of the class taxonomy in structured label problem. We define a misclassification cost associated with the label set produced by the classifier relative to the correct label set (the correct structured label).

Definition 4 (Node Distance). *Node distance is a value $d(c_i, c_j)$ denoting the difference of labels c_i,c_j. It has the following properties:*

- $d(c_i, c_j) \geq 0$
- $d(c_i, c_j) = d(c_j, c_i)$
- $d(c_i, c_i) = 0$

Definition 5 (Dummy Label). *Dummy label θ is an"add-on" label to the class taxonomy which acts as a predicted value to the instance when a classifier can not decide the class label and does nothing. Thus this is a "label by default". It has the following properties:*

- $d(\theta, c_i) = d(\theta, c_j)$
- $d(c_i, c_j) \leq d(\theta, c_i)$

Definition 6 (Non-Redundant Operation). *A non-redundant operation (with Φ as the operator) to a label set C_i is to keep the children labels when both children labels and their parent labels are present, such that we eliminate the label redundancies within a class taxonomy.*

Definition 7 (Mapping). *A mapping f between two label sets $\mathbf{C}_1, \mathbf{C}_2$ with the same cardinality is a bijection $f : \mathbf{C}_1 \rightarrow \mathbf{C}_2$.*

We calculate the distance $d(\mathbf{C}_p, \mathbf{C}_o)$ between $\mathbf{C}_p$ and $\mathbf{C}_o$, the predicted and actual label (respectively) for each classified instance as follows:

- If the cardinalities of $\mathbf{C}_p$ and $\mathbf{C}_o$ are equal, find a mapping to minimize the sum of node distances and divide by the cardinality of the label sets to obtain the distance.
- If the cardinalities of the two label sets are not equal, add as many dummy labels θ as needed to the label set with fewer elements to make the cardinalities of the two label sets equal and then calculate the distance between the two label sets as before.

The performance of the classifier on a test set is obtained by averaging the distances between predicted and actual labels of instances in the test set T as follows: $\bar{d} = \frac{\sum_{\mathbf{T}} d(\mathbf{C}_p, \mathbf{C}_o)}{|\mathbf{T}|}$. The lower the value of this measure, the better the classifier (in terms of misclassification cost).

5 Experimental Results

Given a structured label data set, we need the pair-wise node distances between class labels to compute the misclassification cost as described above. These distances can be specified by a domain expert. Alternatively, the distances may be estimated from a training set based on cooccurence of class labels as follows: For each level in the class taxonomy, we calculate the occurrence of classes in the training set, divide it by the number of labels at that level of the class taxonomy. We calculate the distance between class labels as follows: We place the "add-on" label θ in the root node of the class taxonomy tree and set the edge distance as the level weight. For two nodes, if one is ancestor of the other, the node distance will be the sum of the edge distances along the path that connects them; if neither node is an ancestor of the other, the distance between them

is defined as the average distance of the two nodes from their nearest common ancestor. After normalization, we assign distance 1 to any two labels in the top level together with the "add-on" label θ, and the maximal node distance equals to the summation of all the level weights as 1.268 in Reuters-21578 data and 1.254 in phenotype data set.

5.1 Results on Reuters-21578 Data Set

Reuters-21578 data, originally collected by Carnegie Group for text categorization, does not have a predefined hierarchical class taxonomy. However, many documents are labelled with multiple topic classes. We extracted 670 documents. In this set, more than 72% of the documents have multiple class labels. We created a two-level class taxonomy using current categories of the documents as follows:

grain(barley,corn,wheat,oat,sorghum)
livestock(l-cattle,hog)

We used a Naive Bayes classifier as the base classifier and estimated the performance of the resulting structured label classifier using 5 fold cross validation. The results in tables 1, 2 suggest that binarized structured label learning performs as well as split-based structured label learning in this case. Both have good predictive accuracy for the classes that appear in the first level of the class taxonomy: grain, livestock. The overall performance of the two methods (as measured by the estimated misclassification cost) is slightly different, while the average recall and precision calculated over the entire class hierarchy are very close.

Table 1. Average distance: learning on Reuters-21578 data set

	binarization learning	split-based learning
d	0.217	0.251

Table 2. Recall&precision: learning on Reuters-21578 data set

	binarization learning		split-based learning	
	recall	precision	recall	precision
grain	0.993	0.964	0.993	0.968
livestock	0.766	0.893	0.752	0.917
barley	0.498	0.440	0.454	0.442
wheat	0.852	0.735	0.859	0.724
corn	0.839	0.721	0.818	0.726
oat	0.270	0.75	0.167	0.75
sorghum	0.408	0.560	0.324	0.591
l-cattle	0.146	0.417	0.167	0.339
hog	0.729	0.786	0.717	0.686

5.2 Results on Phenotype Data Set

Our second experiment used the phenotype data set introduced by Clare and King[5] whose class taxonomy is a hierarchical tree with 4 levels and 198 labels.

We choose the C4.5 decision tree as the base classifier to run the binarization learning and split-based learning in 5-fold cross validation. Split-based structured label learning shows better performance than binarized structured label learning on this data set. The misclassification cost is 0.79. The split-based structured label learning predicts 1 out of 4 class labels correctly in the 1_{st} level branches. Compared to the Reuters-21578 data set, the phenotype data set is much more sparse which might explain the fact that the results are not as good as in the case of the Reuters-21578 data set.

We also calculate accuracy, recall and precision of each class label. It turns out that the accuracy of each class label is quite high(95%). This is due to the fact that this data set is highly unbalanced and each classifier has a high true negative rate.Owing to the sparseness of the data set, many class labels do not appear in the test data set. This leads to undefined recall and precision estimates because of division by 0. Hence, only those class labels with recall and precision estimates available are listed in Figure 2. They show that split-based structured label learning performs better in terms of recall and precision, which is consistent with the relative performance of the two methods in terms of misclassification cost.

Table 3. Average distance: learning on phenotype data set

	binarization learning	split-based learning
d	1.171	0.790

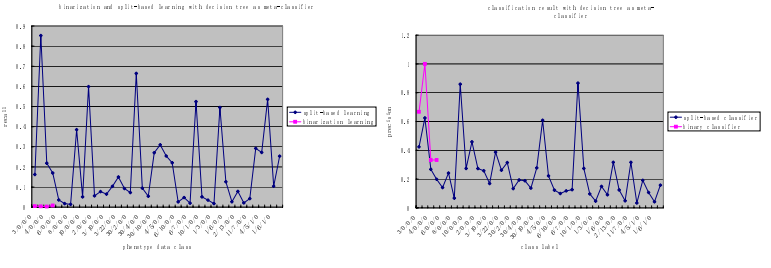

Fig. 2. Recall&precision: learning on phenotype data set

6 Summary and Discussion

In this paper, we have:

– Precisely formulated of learning from data using abstractions over class labels – the structured label learning problem – as a generalization of single label and multi label problems.

- Described two learning methods, binarized and split-based approaches to learning structured labels both of which can be adapted to work with any existing learning algorithm for single label learning task (e.g., Naive Bayes, Decision tree, Support vector machine, etc.).
- Explored a performance measure for evaluation of the resulting structured label classifiers.

Some directions for future work include:

- Development of algorithms to incorporate techniques for exploiting CT (class taxonomies) to handle partially specified class labels.
- Development of more sophisticated metrics for evaluation of structured label classifiers.

Acknowledgements. This research was supported in part by a grant from the National Institutes of Health (GM066387) to Vasant Honavar

References

1. A. McCallum "Multi label text classification with a mixture model trained by EM". AAAI'99 Workshop on Text Learning., 1999.
2. T,Joachims. "Text categorization with Support Vector Machines: Learning with many relevant features". In Machine Learning: ECML-98, Tenth European Conference on Machine Learning, pp. 137–142. 1998
3. X. Shen, M. Boutell, J. Luo, and C. Brown "Multi label Machine learning and its application to semantic scene classification" , in Proceedings of the 2004 International Symposium on Electronic Imaging (EI 2004), Jan. 18-22, 2004
4. H.-P. Kriegel, P. Kroeger, A. Pryakhin, and M. Schubert "Using Support Vector Machines for Classifying Large Sets of Multi-Represented Objects", in Proc. 4th SIAM Int. Conf. on Data Mining, pp. 102-114, 2004
5. A. Clare and R. D. King "Knowledge Discovery in Multi label Phenotype Data", 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2001), volume 2168 of Lecture Notes in Arti cial Intelligence, pages 42-53, 2001
6. H. Blockeel, M. Bruynooghe, S. Dzeroski, J. Ramon, and J. Struyf. "Hierarchical Multi-Classification", Proceedings of the First SIGKDD Workshop on Multi-Relational Data Mining (MRDM-2002), pages 21–35, July 2002
7. K. Wang, S. Zhou, S.C. Liew, "Building hierarchical classifiers using class proximity", Technical Report, National University of Singapore, 1999
8. The Reuters-21578, Distribution 1.0 test collection is available from `http://www.daviddlewis.com/resources/testcollections/reuters21578`