Classifying Entities into an Incomplete Ontology

Bhavana Dalvi
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
bbd@cs.cmu.edu

William W. Cohen
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
wcohen@cs.cmu.edu

Jamie Callan
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
callan@cs.cmu.edu

ABSTRACT

Exponential growth of unlabeled web-scale datasets, and class hierarchies to represent them, has given rise to new challenges for hierarchical classification. It is costly and time consuming to create a complete ontology of classes to represent entities on the Web. Hence, there is a need for techniques that can do hierarchical classification of entities into incomplete ontologies. In this paper we present Hierarchical Exploratory EM algorithm (an extension of the Exploratory EM algorithm [7]) that takes a seed class hierarchy and seed class instances as input. Our method classifies relevant entities into some of the classes from the seed hierarchy and on its way adds newly discovered classes into the hierarchy. Experiments with subsets of the NELL ontology and text datasets derived from the ClueWeb09 corpus show that our Hierarchical Exploratory EM approach improves seed class F1 by up to 21% when compared to its semi-supervised counterpart.

Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering-Algorithms

Keywords

Clustering, Expectation maximization (EM), Hierarchical classification, Knowledge bases, Ontologies, Semi-supervised learning

1. INTRODUCTION

With the exponential growth in world wide web there is a need for organizing this information in a manner that is easier to understand and browse through. The Open Directory Project and the Yahoo! Directory are examples of hierarchies developed to organize pages on the Web. Wordnet, NELL and Freebase are examples of large knowledge bases that organize entities into such hierarchies.

The availability of large unlabeled datasets and hierarchical class structures present new challenges. Manually created static hierarchies are no longer sufficient to classify ever increasing and continuously changing web data, as manually creating ontologies (along with labeled examples) and keeping them up-to-date is extremely costly. Hence there is a need for techniques that can work with incomplete ontologies, use the available seed examples to populate known classes with better precision, and discover new entity classes that can be added to the ontology to extend the set of classes further. Knowledge Base Population (KBP) task [12] is targeting similar problems.

There have been numerous research studies in the area of supervised hierarchical classification. Instead of focusing on individual classes in isolation, hierarchical classification addresses joint training and inference based on the hierarchical dependencies among classes. There are variety of techniques [4, 5, 9, 14] developed to do hierarchical classification given an ontology and sufficient training data as input. There has also been some work on extending existing ontologies [11, 13]. However, they are limited to only discovering new relations or attributes of existing classes. There has also been some work in topic modeling [2, 3] on discovering ontologies in a completely unsupervised way. In this paper we focus on the problem of populating and extending existing knowledge base using few seed examples and a large unlabeled dataset with the focus being on improving precision/recall of seed classes. This problem is at the intersection of "semi-supervised hierarchical classification" and "unsupervised ontology discovery".

Contributions:

Our proposed technique, called "Hierarchical Exploratory Learning" deals with hierarchical semi-supervised classification with incomplete class hierarchies. It enriches existing knowledge base in two ways: first, by adding new instances to the existing classes; and second, by discovering new classes and adding them at appropriate places in the ontology. To achieve this we extend the Exploratory EM algorithm [7] to work with a class hierarchy. Our method improves the F1 score for seed classes when compared to a semi-supervised learning method that works with original seed hierarchy. In these experiments we assume that classes are arranged in a tree structure and classes at any level of hierarchy are mutually exclusive. We also discuss simple extensions by which our method can be applied in the presence of complex class constraints.

In Section 2, we define the problem formally and then present our proposed technique "Hierarchical Exploratory EM". Experimental results are covered in Section 3, followed by the related work (Section 4) and conclusions (Section 5).

2. HIERARCHICAL EXPLORATORY EM

Our method is derived from the Exploratory EM algorithm proposed by Dalvi et al. [7]. Exploratory learning takes the same inputs as traditional Semi-Supervised Learning(SSL) methods, i.e. a set of classes $C_1, C_2, \ldots C_k$, a few labeled datapoints X^l and a large number of unlabeled datapoints X^u. X^l contains a (usually small) set of "seed" examples of each class, the task is to learn a

AKBC'13, October 27 - 28, 2013, San Francisco, CA, USA.

model from X_l and use it to label datapoints in X_u. Every example x may be predicted to be in either a known class $C_i \in C_1 \dots C_k$, or a newly discovered class $C_i \in C_{k+1} \dots C_m$.

There are two main differences between the Exploratory EM algorithm and standard classification-EM approaches to SSL. First, in the E step, each of the unlabeled datapoint x is either assigned to one of the existing classes, or to a newly-created class. A new class is introduced when the probabilities of x belonging to existing classes are close to uniform. This suggests that x is not a good fit to any of the existing classes, and that adding x to any existing class will lower the total data likelihood substantially. Second, in the M-step of iteration t, we choose either the model proposed by Exploratory EM method which might have more classes than the previous iteration $t-1$, or the baseline model with same number of classes as iteration $t-1$. This choice is based on whether exploratory model satisfies a model selection criterion in terms of increased data likelihood and model complexity. This intuition is derived from the Structural EM approach [8].

As per the experimental results presented in [7], the Exploratory EM method is comparable or better than "Gibbs sampling with Chinese Restaurant Process(CRP) approach" and does not involve tuning the concentration parameter for CRP. However, Exploratory EM is limited to flat class hierarchies. In this paper we propose the Hierarchical Exploratory EM algorithm which can work with incomplete class hierarchies and small amount of seed labels.

2.1 Problem Definition

Given a set of class constraints Z_k, a small set of labeled data points X^l, their labels Y^l, and a large set of unlabeled data points X^u; the task is to label data points from X^u, adding m new classes if needed and extending the class constraints to Z_{k+m}. Here, each point from X^l can have multiple labels at different levels of the hierarchy satisfying constraints Z_k, and Z_{k+m} defines the class constraints on k seed and m newly added classes, $Z_k \subseteq Z_{k+m}$ and the labels Y^u of X^u satisfy Z_{k+m}. (Section 2.3 provides concrete examples of class constraints.)

2.2 Proposed Method

One simple way to use Exploratory EM algorithm for our purpose will be to run it as it is at each level of hierarchy. Henceforth, we refer to this approach as FLAT-ExploreEM and consider it as a baseline for our proposed Hierarchical Exploratory EM approach. At each level, it selects one of the existing classes or creates a new class in case the datapoint doesn't clearly belong to one of the known classes. This algorithm does not make explicit use of class constraints while making class assignments at each step. Hence the assignments done by this algorithm might not be consistent, since assignments at level 3 are not influenced by assignments at level 2.

Next, we define a generic Hierarchical exploratory algorithm that can take a set of class constraints in terms of subclass or mutual exclusion relationships. This algorithm, in each iteration assigns a bit vector to each data point (number of bits equals number of classes), each bit indicating whether the data point belongs to the class. Class constraints decide whether a bit vector is consistent or not. E.g. if class constraints include "Car is a of type Machine", then for consistency, when the bit for "Car" is set, the bit for "Machine" should also be set. Further new classes can be added during each iteration, hence the length of these bit vectors changes dynamically and the algorithm maintains class constraints containing old as well as newly added classes. The generic algorithm is described in Algorithm 1. There can be multiple ways to implement functions "ConsistentAssignment" and "UpdateConstraints".

One simple instantiation of Algorithm 1 can be done by using Divide-and-Conquer(DAC) strategy. Here we assume that classes are arranged in a tree hierarchy, and classes at any one level are mutually exclusive. To do class assignments for any non-seed point, we will first start with root. Every data point belongs to the root node. Then at level 2 use exploreEM to either assign this point to one of the existing clusters or create a new one. Recurse the same procedure for child tree of the node selected at level 2. Algorithm 2 describes the "ConsistentAssignment" and "UpdateConstraints" functions for this approach, named DAC-ExploreEM.

Algorithm 1 Generic Hierarchical Exploratory EM algorithm

1: **function Hierarchical-ExploreEM** (X^l, Y^l, X^u, Z_k): Z_{k+m}, Y^u
2: **Input**: X^l labeled data points; Y^l labels of X^l; X^u unlabeled data points; Z^0 manually input constraints on k seed classes (subclass-superclass or mutual-exclusion kind); P_{new} probability of creating a new class
3: **Output:** Z_{k+m} Extended set of class constraints on k seed and m newly added classes; Y^u labels for X^u
4: h = height of ontology that is part of Z_k
5: Initialize classifiers θ_j for each class C_j using seeds provided for C_j
6: **for** t = 1 to $maxIter$ **do**
7: **for** i=1 to $|X|$ **do**
{E step: Classify each point at each level}
8: Find $P(C_j|X_i)$ for all classes C_j
9: Z^t = **UpdateConstraints**($\{X^l \cup X^u\}$, $\{Y^l \cup Y^u\}$, Z^t)
10: Y_i^t = **ConsistentAssignment**($P(C_j|X_i)$, h, Z_C)
11: **end for**
{M step: Recompute model parameters}
12: Compute cluster centroids based on class assignments Y^t in E step
13: $Z^{t+1} = Z^t$
14: **end for**
15: **end function**

2.3 Modeling Class Constraints

There are different kinds of class constraints imposed by ontologies. Two example constraints are as follows:
(1) The "Subclass-Superclass" constraint between "Mammals" and "Animals" categories suggests that if a datapoint is classified as "Mammals", then it should also be classified as "Animals".
(2) The "Mutual Exclusion" constraint between "Animals" and "Electronic Devices" says if a datapoint is member of "Animals", then it should not be member of "Electronic Devices", and vice versa.

Note that the example ontologies we are using (refer Figure 1) has a tree structure. In practice, class constraints can be more complicated e.g. existence of overlapping classes. Note that DAC-ExploreEM (Algorithm 2) assumes tree structure and mutual exclusion of classes at any level of hierarchy. However, Algorithm 1 is generic enough to work with more complicated class constraints.

In ongoing work in this direction we are exploring the use of Markov Random Fields(MRF). Here, MRF is built for each datapoint using class probabilities modeled as node potentials and the class constraints modeled as edge potentials. Belief propagation (or loopy belief propagation) is then used to infer class assignments for each data point satisfying the given class constraints. This method can easily support any kind of class constraints, hence not restricted to the tree structured ontologies.

3. EXPERIMENTAL EVALUATION

In this section we present the experimental results of our Hierarchical Exploratory EM approach. For this task we work with subsets of NELL's ontology at two different points in NELL's development. Figure 1 shows the two ontologies we used for the experiments presented in this paper.

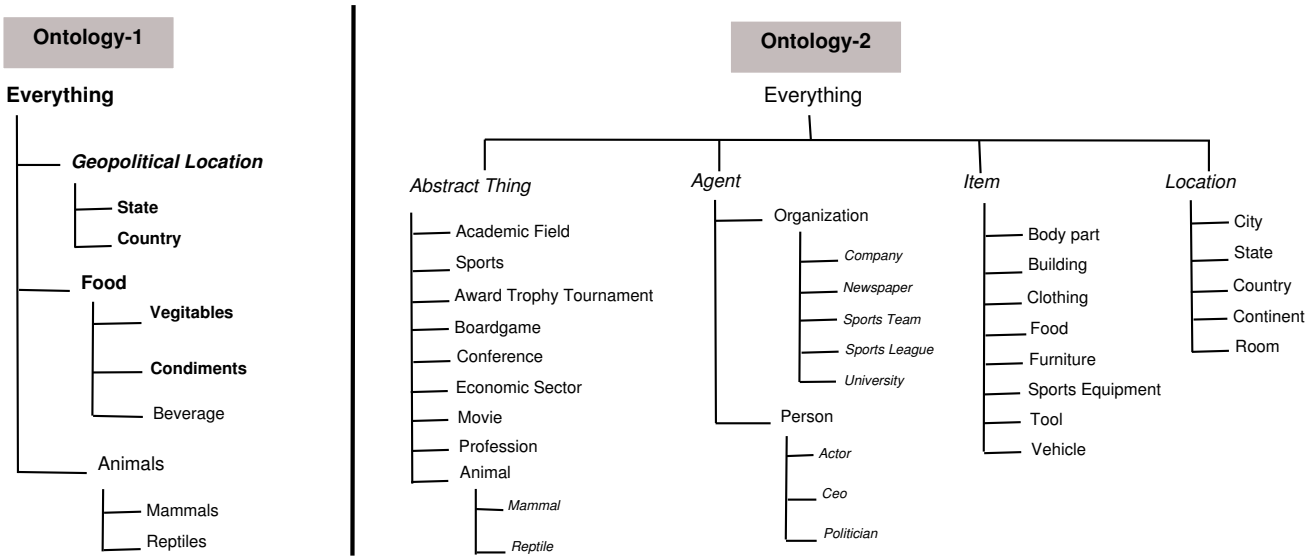

Figure 1: Ontologies used for hierarchical classification.

Datasets

For experimental evaluation we wanted to create datasets that have ground truth labels for all entities in them. For this purpose, we derived two datasets using the ontologies in Figure 1 and text pattern dataset (used by Carlson et al. [5]) extracted from the ClueWeb09 corpus. For each entity we have the text patterns as features and feature weights are TFIDF weighted occurrence counts of each pattern with the entity. To create a dataset from an ontology, we took all entities that are labeled with at least one of the classes under consideration, and retrieved their feature representation in terms of occurrences with text patterns. The first ontology (Figure 1 (left)) has 3 levels and 11 classes. The dataset created using this ontology is referred to as DS-1. The second ontology (Figure 1 (right)) has 4 levels and 39 classes. The dataset created using this ontology is referred to as DS-2. Table 1 shows the statistics for DS-1 and DS-2.

Results

Next, we compare FLAT-ExploreEM algorithm with DAC-ExploreEM. The evaluation metric used for comparison is "macro-averaged seed class F1" (computed by macro averaging F1 values of seed classes only). This metric is further averaged for 10 runs of both algorithms. Each run's input consists of different seed ontologies and randomly sampled 10% of relevant datapoints as seed examples. The same set of inputs is given to both SemisupEM and Exploratory EM algorithms for both FLAT and DAC classification.

For DS-1, we generated 10 sets of seed examples, for the same seed ontology. The chosen seed ontology is bold-faced in Figure 1 (left). Here the seed ontology always contains 7 out of 11 classes. For DS-2, seed ontology also varies across runs. In each run, we randomly chose 10 leaf nodes according to their class frequency (i.e. popular class is more likely to be chosen in each run). After sampling 10 leaf nodes (sampling without replacement), we included all their ancestors to create the seed ontology for that run. 10% of the datapoints from these classes are then randomly sampled as training data for the run. The average ontology size generated using this method was 16.7. Table 2 column 2 shows average number of nodes at each level.

In Table 2 we can see that, Exploratory EM version of each algorithm improves seed class F1 when compared to SemisupEM. The statistical significance of results is computed by doing a pairwise t-test on 10 runs of both algorithms. ▲ (and △) indicates that improvements w.r.t. SemisupEM are statistically significant with 0.05 (and 0.1) significance level. Here we used seeded K-Means algorithm for clustering and MinMax criterion [7] for new class creation [1]. The best performance in each row is bold-faced. We can clearly see that DAC approach of hierarchical classification beats flat classification in 4 out of 5 cases.

The results are encouraging in the sense, they show that hierarchical classification is more effective than flat classification as it is making use of class constraints while doing inference for each datapoint. Further exploratory approach gives comparable or better results when compared to its semi-supervised counterpart. In our MATLAB implementation, the running time of ExploratoryEM is longer compared to SemisupervisedEM, but not unreasonably so: on average for DS-2 dataset DAC-SemisupervisedEM algorithm took 372 seconds while DAC-ExploratoryEM took 715 seconds. On the same dataset FLAT-SemisupervisedEM took 80 seconds while FLAT-ExploratoryEM took 810 seconds.

4. RELATED WORK

There has been a lot of research done in the area of supervised hierarchical classification. Instead of focusing on individual classes in isolation, hierarchical classification addresses joint training and inference based on the hierarchical dependencies among the classes. Cai and Hofmann [4] proposed a SVM based hierarchical classification method that is based on discriminant functions that are structured in a way that mirrors the class hierarchy. Gopal et al. [9] propose Bayesian methods to model hierarchical dependencies among class labels using multivariate logistic regression. Zhou et al. [14] propose an SVM based classifier that operates with a tree structured class hierarchy and classifies the examples recursively from the root to the leaves. Their technique encourages the classifiers at each node of the tree to be different from the classifiers at

[1]MinMax criterion introduces a new class given a data point and posterior distribution of classes given the data point, if the maximum to minimum probability ratio is less than 2.

Dataset	#classes	#Levels	#Classes per level	# NELL entities	# contexts	# entity, context pairs	# NELL entity-label pairs
DS-1	11	3	1, 3, 7	2.5K	3.4M	8.8M	6.7K
DS-2	39	4	1, 4, 24, 10	12.9K	6.7M	25.8M	42.2K

Table 1: Dataset statistics

Dataset	#train/test data-points	Level	#seed/#Ideal classes	Macro-average Seed Class F1			
				FLAT		*DAC*	
				SemisupEM	Exploratory EM	SemisupEM	Exploratory EM
DS-1	335/2.2K	2	2/3	43.2	**78.7** ▲	69.5	77.2 △
		3	4/7	34.4	42.6 ▲	31.3	**44.4** ▲
DS-2	1.5K/11.4K	2	3.9/4	64.3	53.4 ▼	65.4	**68.9** ▲
		3	9.4/24	31.3	33.7 ▲	34.9	**41.7** ▲
		4	2.4/10	27.5	38.9 ▲	**43.2**	42.4

Table 2: Comparison of Exploratory EM w.r.t. SemisupEM in terms of macro average F1 for seed classes. ▲ (and △) indicates that Exploratory EM results are statistically significant w.r.t their respective SemisupEM baselines with 0.05 (and 0.1) significance level.

Algorithm 2 Divide-And-Conquer Hierarchical Exploratory EM

function ConsistentAssignment-DAC ($P(C_j|X_i)$, Z_k}): $assgn_x$, Z_{k+m}
Input: $P(C_j|x)$ probability distribution of classes given a datapoint x; Z_k class constraints on k seed classes.
Output: Y_x assignment of x to classes satisfying Z_k; $Z_{k'}$ extended set of class constraints on k' resultant classes.
for l = 1 to h **do**
  if X_i has seed label at level l **then**
    label(X_i, $level_l$) = seed label;
  else
    $candidateClasses$ = children(label(X_i, $level_{l-1}$))
    if $candidateClasses$ is not empty **then**
      Let P_{cand} = probability distribution over candidate classes
      if P_{cand} is nearly uniform **then**
        Create a new class C_{new} at level l
        Assign X_i to C_{new}
        Set parent(C_{new}) = class choice at level $l-1$
      else
        Assign X_i to $argmax_{C_i} P_{cand}$
      end if
    end if
  end if
end for
end function

function UpdateConstraints-DAC (X, Y, Z^{old}): Z^{new}
Input: X: Dataset; Y: Class assignments for each point in X; Z^{old}: Old constraints on the existing set of classes.
Output: Z^{new}: Updated set of class constraints,
Each newly created class is assigned a single parent at the time of creation
Add each parent, child relationship as a constraint in $Z_{k'}$
end function

its ancestors. All these methods assume that the class hierarchy is complete and there is enough training data to learn classifiers for each node in the class hierarchy. On the other hand we considered the situation where only part of the ontology is known upfront with very few seed examples for each of the seed class. Further our method can be easily extended to cases where class constraints are more complicated than the examples considered in this paper, e.g. overlapping classes and mutual exclusion constraints.

Another related research area is constructing web-scale knowledge bases by doing information extraction from various data-sources e.g. NELL [10], Freebase [1] etc. NELL internally uses Coupled semi-supervised learning [5] that takes into account subclass and mutual exclusion constraints among classes to filter extraction patterns and instances at the end of each bootstrapping iteration. The ontology (class hierarchy) is not explicitly used in the prediction process. I.e. it does flat classification with class constraints applied as post-processing in between two iterations of bootstrapping. Our approach on the other hand does collective inference i.e. hierarchical classification.

There has also been some work to extend existing ontologies. Mohamed et al. [11] propose a co-clustering based two step approach to discover new relations between two already existing classes in the knowledge base. These new relations are named using centroid features of the intermediate clusters. This method is focused on relation discovery between known classes. Reisinger and Paşca [13] addressed the same problem as ours, working with the Wordnet hierarchy. Their fixed-structure and sense selective approaches use the Wordnet hierarchy directly and annotate existing concepts with generic property fields (attributes). On the other hand, Nested Chinese Restaurant Process (nCRP) approach is hierarchical extension of LDA to infer arbitrary fixed-depth tree structures from data. nCRP generates its own annotated hierarchy whose concept nodes do not necessarily correspond to Wordnet concepts. Our method is in the middle of these two approaches, as it uses the existing class hierarchy with small amount of training data and extends it dynamically as new clusters of datapoints are discovered.

There has also been some work on completely unsupervised information extraction and ontology discovery [2, 3, 13, 6]. Though very effective, these approaches are not making use of the valuable information hidden in the existing knowledge bases. Our approach is relatively novel in the sense that it is in between semi-supervised and unsupervised learning, where some part of ontology is known, and this knowledge is used to discover the missing parts of the ontology along with populating it with new data instances.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we propose the Hierarchical Exploratory EM approach that can take an incomplete seed ontology as input, along with a few examples of each seed class, to populate new instances of seed classes and extend the ontology with newly discovered classes. Preliminary experiments show encouraging results. Hierarchical classification performs better than flat classification at deeper levels of hierarchy. The relative improvements in seed class F1 increases as we go further down the hierarchy.

Another observation is that exploratory learning gives comparable or better performance when compared to their semi-supervised counterpart for both flat and hierarchical classification. Continuing further in this direction, we are trying to incorporate arbitrary class constraints while doing hierarchical classification. For this purpose we are exploring the use of Markov Random Fields.

Acknowledgments

This work is supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory (AFRL) contract number FA8650-10-C-7058. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. This work is also partially supported by the Google Research Grant. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Google, IARPA, AFRL, or the U.S. Government.

6. REFERENCES

[1] Freebase. *http://freebase.com*.

[2] R. P. Adams, Z. Ghahramani, and M. I. Jordan. Tree-structured stick breaking for hierarchical data. *NIPS*, 2010.

[3] D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 2010.

[4] L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *CIKM*, 2004.

[5] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka, Jr., and T. M. Mitchell. Coupled semi-supervised learning for information extraction. In *WSDM*, 2010.

[6] B. Dalvi, W. Cohen, and J. Callan. Websets: Extracting sets of entities from the web using unsupervised information extraction, 2012.

[7] B. Dalvi, W. W. Cohen, and J. Callan. Exploratory learning. In *ECML*, 2013.

[8] N. Friedman. The bayesian structural em algorithm. In *UAI*, 1998.

[9] S. Gopal, B. Bai, Y. Yang, and A. Niculescu-Mizil. Bayesian models for large-scale hierarchical classification.

[10] T. Mitchell. Nell: Never-ending language learning. *http://rtw.ml.cmu.edu/rtw/*.

[11] T. P. Mohamed, E. R. Hruschka Jr, and T. M. Mitchell. Discovering relations between noun categories. In *EMNLP*, 2011.

[12] NIST. Knowledge base population (KBP) task. *Text Analysis Conference http://www.nist.gov/tac/2013/KBP/*, 2013.

[13] J. Reisinger and M. Paşca. Latent variable models of concept-attribute attachment. In *ACL-IJCNLP*, 2009.

[14] D. Zhou, L. Xiao, and M. Wu. Hierarchical classification via orthogonal transfer. In *ICML*, 2011.