Learning Hierarchically Decomposable Concepts with Active Over-Labeling

Yuji Mo* Stephen Scott† Doug Downey‡

Abstract

Many classification tasks target high-level concepts that can be decomposed into a hierarchy of finer-grained subconcepts. For example, some string entities that are Locations are also Attractions, some Attractions are Museums, and so on. Such hierarchies are common in named entity recognition (NER), document classification, and biological sequence analysis. We present a new approach for learning hierarchically decomposable concepts. The approach learns a high-level classifier (e.g., location versus non-location) by seperately learning multiple finer-grained classifiers (e.g., museum versus non-museum), and then combining the results. Soliciting labels at a finer level of granularity than that of the target concept is a new approach to active learning, which we term "active over-labeling." We show that simple methods for active over-labeling can improve performance, in terms of theoretical guarantees and empirical results. In experiments in NER and document classification tasks, we show that active over-labeling substantially improves area under the precision-recall curve, when compared with standard passive or active learning.

1 Introduction

Several popular applications of machine learning have instances that can be classified with hierarchical labeling schemes. For example, in named entity recognition (NER), the phrase "The Metropolitan Museum of Art" can be labeled as a Location, as a Building, or as a Museum, where each label forms a subcategory of the previous one. Likewise, in the Gene Ontology [**?**], a biological sequence can be labeled with multiple terms from a hierarchical labeling scheme.

While some recent work has considered classification into fine-grained (low-level) categories [**?** **?**] or hierarchies [**?**], the majority of classification tasks in practice target a small number of labels. Typically, NER, for example, targets a small set of coarse-grained labels such as Location, Organization, and Person [**?**]. We show how classifiers aimed at coarse-grained tasks can be improved by training on fine-grained labels. For example, we show an NER system can be made more precise by not treating the broad "Organization" label as a *single* concept, but instead explicitly learning a combination of fine-graned labels comprising the category (e.g., "University," "Railroad Company," and so on). As we argue theoretically in Section **??** and establish in our experiments, fine-grained labeling information can significantly improve the precision of a classifier aimed at a coarse-grained task. Further, actively soliciting informative fine-grained labels, rather than passively sampling them, provides an additional accuracy boost. We refer to this new approach, which extends the standard active learning model to one in which the learner can solicit labels at finer levels of the hierarchy than that targeted for classification, as *active over-labeling*.

We present a general schema for performing active over-labeling for any given hierarchical classification task, using any given probabilistic base learner. In our experiments, we demostrate the effectiveness of the schema across multiple classification tasks (NER and document classification) and multiple base learners (Conditional Random Fields and Logistic Regression). We show that over-labeling improves accuracy over standard passive or active learning, and that active over-labeling outperforms passive over-labeling, in terms of area under the precision-recall curve.

Finally, it is reasonable to assume that obtaining a finer-grained label will often be more costly than obtaining a coarse-grained label. For example, in the Gene Ontology, one could label a sequence as being involved in biological processes, but a finer-grained label for the same sequence could specify it as involved in growth, immune system process, localization, or one of 21 other labels. The greater specificity of the fine-grained label requires more expertise and effort to obtain. Thus, we also analyze the trade-off between the increased cost of obtaining finer-grained labels, versus the potential increased benefit in learning.

The rest of this paper is organized as follows. In Section **??**, we formally define our learning setting and give intuition as to why our approach offers advantages to learning in this context. Section **??** provides theoretical results showing that over-labeling improves computational complexity and label complexity when compared to conventional passive or

*Dept. of Computer Science and Engineering, University of Nebraska, email: ymo@cse.unl.edu

†Dept. of Computer Science and Engineering, University of Nebraska, email: sscott@cse.unl.edu

‡EECS Department, Northwestern University, email: ddowney@eecs.northwestern.edu

active learning. In Section **??** we present our algorithms in our learning model, and we then present our experimental results in Section **??**. We discuss related work in Section **??** and conclude in Section **??**.

2 Problem Definition

Our setting resembles conventional pool-based active learning. Formally, our task is to learn a target concept over an input space $\mathcal{X}$, i.e., a function $f : \mathcal{X} \to \mathcal{Y} = \{0, 1\}$. We are given a pool of unlabeled examples $U \subset \mathcal{X}$, a base probabilistic machine learner L that can be trained on labeled examples $(\mathbf{x}, y)$, and access to an oracle that can be queried at some cost for the label of any example $u \in U$. Our goal is to choose a relatively small number of examples from U to be labeled, and train L on the labeled examples to output a relatively accurate classifier.

Our setting resembles conventional active learning models in that the learner purchases labels and builds its classifier in order to make predictions of the labels of new instances. The key distinction is that in our model, the oracle can also return more refined label information for each query example. Specifically, we consider target concepts that can be decomposed hierarchically into constituent subconcepts, at varying levels of granularity. We assume the decomposition is given to the learner, and that the oracle can then return labels corresponding to any node in the hierarchy, which we refer to as a *labeling tree*. An example labeling tree based on Reuters Corpus Volume I (RCV1) [**?**] is shown in Figure **??**.

Formally, nodes of the tree each represent concepts (i.e., subsets of the instance space $\mathcal{X}$). The root of the labeling tree corresponds to the target concept to be learned, and lower levels of the tree are sub-concepts of their ancestor concepts. The oracle in our setting returns a *vector* of labels, corresponding to a path starting at the root of the tree. For example, based on the label tree in Figure **??**, an instance could be labeled as ⟨Location,Building,Museum⟩, ⟨Location,Attraction,Museum⟩, ⟨Location,Lake,X⟩, or $\langle X, X, X\rangle$, where an 'X' indicates that no value at that level applies. Thus, the latter labeling example indicates an instance that is not a location.

A vector of labels is denoted $\langle \ell_1, \ldots, \ell_k\rangle$, where a label ℓ_i is the instance's label at the ith level of the tree, or 'X' if that is undefined in the tree. If $\ell_i = X$, then $\ell_j = X$ for all $j > i$ (i.e., if a level-i label does not apply, then no other label farther from the root may either). Further, if i is the largest value such that $\ell_i \neq X$, then the values $\ell_i, \ldots, \ell_1$ must form a path from a leaf to the tree's root.

Each instance in U is initially labeled with the vector ⟨?,?,. . .,?⟩, where '?' denotes a label value that is yet unspecified but can be purchased. For a specific instance, a value for ℓ_i may be purchased at cost $c_i \geq c_j > 0$ for all $i > j$. We assume that a purchase of ℓ_i automatically yields the values of ℓ_1 through ℓ_{i-1}. E.g., a purchase of ℓ_3 of an instance could yield ⟨Location,Building,Museum⟩, ⟨Location,Attraction,Museum⟩, ⟨Location,Lake,X⟩, or $\langle X, X, X\rangle$.[1] Further, an ℓ_2 purchase could yield ⟨Location,Building,?⟩, ⟨Location,Attraction,?⟩, ⟨Location,Lake,X⟩, or $\langle X, X, X\rangle$ (once an 'X' or a leaf is encountered, one can fill in the rest of the vector with 'X').

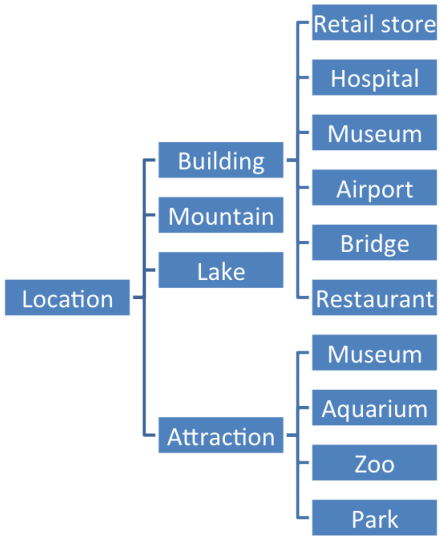

Figure 1: An example labeling tree based on Reuters Corpus Volume I (RCV1) [**?**].

2.1 Intuition Over-labeling relies on learning classifiers for the fine-grained (non-root) concepts and combining the results, rather than simply directly learning the coarse-grained (root) concept. To see the potential advantage of over-labeling, consider the simple example of Figure **??**. In the figure, the coarse-grained concept to be learned is circles (positives) versus diamonds (negatives). If one limits the set of possible classifiers $\mathcal{C}$ to the set of single axis-parallel boxes, then any hypothesis that has high recall will have low precision (any rectangle that contains most of the circles will also contain many diamonds). However, if it is the case that the set of positive instances can be decomposed into fine-grained classes such as the four separate types of circles in Figure **??**, then we can decompose the problem of classifying circles versus diamonds into four problems: classifying green solid circles versus everything else, classifying blue open circles versus everything else, and so on. We could thus train four fine-grained classifiers, one per circle type. With these inferred fine-grained classifiers, one can predict on a new instance by predicting its membership in each of the

[1] We assume that all labels in the same level are distinct, e.g., 'Museum' under 'Attraction' is distinguishable from the one under 'Building'.

four fine-grained classes and then returning a root-level prediction of circle if any fine-grained classifier predicts 'yes'.

In general, over-labeling takes advantage of a natural decomposition of the target class into finer, possibly simpler sub-classes. If the sub-classes are in fact simpler to learn, then we can more easily learn the general class by first learning the sub-classes and combining the sub-class predictions via, e.g., the union operator. Since the union of hypotheses is a larger, more general hypothesis space that includes the space of original hypotheses, this lends us a potentially strong advantage in terms of representational ability.

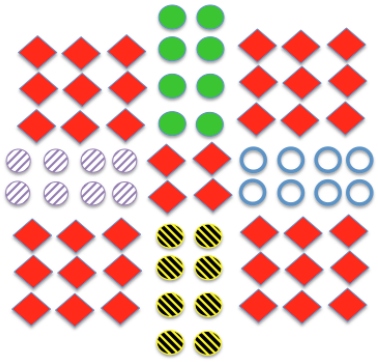

Figure 2: An example of the potential usefulness of learning multiple fine-grained concepts to support the learning of a single coarse-grained one. Negative instances are diamonds and positive ones are circles.

2.2 Learning-Theoretic Advantages

~~3 Theoretical Results~~

We begin by proving formally that active over-labeling provides advantages in terms of both computational complexity and label complexity. We present results in both the *probably approximately correct (PAC)* [**?**] and *exact* [**?**] models of learning. The results below focus on the simple, but important case of learning concepts that are unions of axis-parallel boxes over real and ordinal feature spaces.

2.2.

~~3~~.1 PAC Learning with Active Over-labeling

In PAC, a learner is given parameters $0 < \epsilon, \delta < 1/2$ and access to labeled training instances drawn iid according to arbitrary distribution $\mathcal{D}$. The learner then outputs a hypothesis in polynomial time that, with probability at least $1-\delta$, has error at most ϵ on new instances drawn according to $\mathcal{D}$.

2.2.

~~3~~.1.1 Computational Complexity

In the context of computational complexity, we consider the case of *proper learning*, in which the training instances are labeled by a concept from $\mathcal{C}$ and the hypothesis inferred by the learner is required to also be from $\mathcal{C}$. We consider the task of learning concepts that are unions of k axis-parallel boxes in $\mathbb{R}^d$. This task is not properly PAC-learnable if RP $\neq$ NP.

Observation

~~Proposition~~ 1. *Learning* k*-unions of axis-parallel boxes in* $\mathbb{R}^d$ *is not PAC-learnable unless RP* $=$ *NP.*

properly

(add specific thrm?)

Proof: From Blumer et al. [**?**], concept class $\mathcal{C}$ is properly PAC learnable iff there exists a polynomial-time algorithm to find a hypothesis from $\mathcal{C}$ consistent with a size-m labeled training sample $\mathcal{X}$ (called the *consistent hypothesis problem*). It is known to be NP-hard [**?** **?**] to find a smallest set of rectangles to cover a set of points in $\mathbb{R}^d$ even for $d = 2$. Thus, the consistent hypothesis problem for k-unions of boxes is NP-hard, implying that one cannot properly PAC learn k-unions of boxes. □

Move proof to text before observation; combine Obs 1 with 2 into single observation

In contrast, consider an over-labeling version of this learning problem, in which each of the k boxes is a separate subconcept, as in Figure **??**. Thus, examples from the ith box ($i = 1, \ldots, k$) have a fine-grained label (call it I_i) and all other examples are labeled '$-$'.

~~Proposition~~ 2. *In the over-labeling setting,* k*-unions of boxes is properly PAC-learnable.*

Proof: For each subconcept, a consistent concept (single bounding box) can be learned from the fine-grained labels in time $O(dm)$. The learner can learn each of the k subconcepts separately and output their union in time $O(kdm)$, and this will be consistent with the labeled examples. Thus, the consistent hypothesis problem can be solved in polynomial time, which Blumer et al.'s [**?**] result implies that k-unions with over-labeling is properly PAC learnable. □

~~3~~.1.2 Label Complexity

We now consider label complexity, in which one wants to minimize the number of labels purchased by a pool-based active learning algorithm. We will work in a model where we are given a size-m set of training data U, but initially the labels are missing. When seeking a PAC algorithm for learning, one can apply a standard result from Blumer et al. [**?**] that says if the algorithm efficiently finds a hypothesis consistent with U with size at least

$$m(\epsilon, \delta) = \max\left(\frac{2}{\epsilon}\log\frac{2}{\delta}, \frac{8D}{\epsilon}\log\frac{13}{\epsilon}\right) \tag{3.1}$$

(where D is the *VC dimension* of $\mathcal{C}$), then with probability $\geq 1-\delta$, the hypothesis will have error at most ϵ. If the instances of U are unlabeled, the goal in active learning is to purchase as few labels of instances of U as possible and still guarantee a consistent hypothesis to yield the PAC result.

For this example, we focus on what we term the *disjoint* k*-intervals problem*. I.e., $\mathcal{C}$ is the set of unions of $\leq k$ disjoint intervals on $\mathbb{R}$. When a coarse-grained label of instance $x \in U$ is purchased, it returns '$+$' if x lies in one of the k target intervals and '$-$' otherwise. When a fine-grained

Emphasize process: m unlabeled instances given and static; goal is to minimize number of labels purchased

Add footnote 3 from ICDM 15 to emphasize contrast of proper vs non poper

The theoretical results section needs some clarification.
Proposition 2 seems to hold only if each box is a separate subconcept, which needs to be stated in the proposition, as it is an important limitation. I suggest to formally define the disjoint k-intervals problem before proposition 3 (it is explained informally in the text, though).
In the proof of proposition 3, I am not convinced about the worst-case $\Omega(m_c)$. What is the argument against using binary search here like in the later proof of proposition 4?
In section 3.2, when writing "it is similar to an active learning model" (and in the subsequent sentences), it should be stated more precisely that "it is similar to an pool-based active learning model", as in query synthesis the actively requested instances need not to be from a pre-specified set.
I suggest to extend the discussion on proposition 5 and its the relation to results from [11].

The paper has little to no novelty to offer in terms of theoretical results. Proofs are written rather shoddily.
1) The work of Blumer et al *defines* PAC learnability in a certain way. It is not an if and only if result the way the proof of Proposition 2 makes it look like.
2) Proposition 4 is a simple extension of the toy problem used to illustrate the benefits of active learning.
3) I am not sure if the proof of Proposition 1 qualifies as a proof since it simply appeals to a result of Blumer et al which negates the possibility of finding the minimum cover using axis aligned rectangles. This negates consistent learning but not PAC learnability which only requires an approximate cover to be learnt and that too with some confidence.
4) Proposition 3 is not proved properly - the term "worst case" is used without even defining what class of learning algorithms are being used. Indeed, if the distribution over the entire instance space is uniform then constantly many queries would yield the location of a single negative example between two adjacent positive intervals after which the binary search algorithm would yield the boundaries in logarithmically many more queries.

label of x is purchased, the label is an indicator of which of the k target intervals it lies in $(I_1, \ldots, I_k)$ or '$-$' if it does not lie in any interval. We assume that there is at least one point from U in each interval I_j and that there is at least one point from U between each adjacent pair of intervals.

In the following two propositions, we bound the number of purchases needed in each labeling scheme. Since the total number of instances needed for PAC learning (per Equation **??**) differs between them (due to different VC dimension), in the next two propositions, we use m_c for the number of instances in coarse-grained learning and m_f needed for fine-grained.

Assume that, for each target interval, there is one instance of U that is pre-labeled for free. I.e., in the coarse-grained case, there are k instances labeled '$+$' (one in each target interval) and in the fine-grained case there is one instance labeled I_1, one labeled I_2, etc.

Again, maybe merge both props into single obs, with proofs beforehand

~~PROPOSITION~~ 3. *The disjoint k-intervals problem with coarse-grained labels on m_c instances requires $\Omega(m_c)$ purchases in the worst case.*

Proof: The algorithm must find the left and right boundaries of each of the k target intervals, which is tantamount to identifying the leftmost and rightmost negatively labeled points between each consecutive pair of intervals. Consider two consecutive intervals I_j and I_ℓ. In searching for the negative points from U between I_j and I_ℓ, the learner must purchase the label of some point between x_j and x_ℓ, where x_j and x_ℓ are the pre-labeled points from U from I_j and I_ℓ, respectively. In the worst case, every query will result in a response of '$+$', until only one remains to be labeled '$-$'. Summed over all pairs of intervals, this requires $\Omega(m_c)$ purchases in the worst case. □

~~PROPOSITION~~ 4. *The disjoint intervals problem with fine-grained labels on m_f instances requires $O(k \log m_f)$ queries in the worst case.*

Proof: An algorithm in the active over-labeling setting can perform a binary search between x_j and x_ℓ (labeled I_j and I_ℓ rather than simply '$+$') until a negatively labeled instance x_- is found. When that is done, the learner can simply perform two binary searches: one between x_- and the right-most point in I_j and one between x_- and the left-most point in I_ℓ. This requires at most $O(\log m_f)$ queries per pair of adjacent intervals, for a total of $O(k \log m_f)$ queries. □

To bound m_c, we use Equation **??** with $D = 2k$ and get (ignoring the typically smaller first term) a number of purchases $\Omega(m_c) = \Omega((k/\epsilon)(\log 1/\epsilon))$. To bound m_f, note that we have k independent learning problems, each a single box. Thus, we can use $D = 2$, but the parameters ϵ and δ must each be reduced by a factor of k, since the errors of these hypotheses accumulate. Further, we must apply the learning process k times, so (again ignoring the first term) $m_f = O((k^2/\epsilon)\log(k/\epsilon))$, so our worst-case upper bound of purchases is $O(k \log(k/\epsilon) + k \log\log(k/\epsilon))$. Both bounds grow linearly in k but the coarse-grained learner's bound is worse by a factor exponential in $1/\epsilon$.

~~3~~.2 Exact Learning with Active Over-labeling

We illustrate the computational complexity advantages of active over-labeling in the exact learning setting. In exact learning, the learner gets access to two oracles: a *membership query* (MQ) oracle and an *equivalence query* (EQ) oracle. An efficient learner will learn the exact identity of the target concept in time and number of queries that are polynomial in the problem size. When the learner poses an EQ, it passes to the oracle a hypothesis h that it thinks is exactly equivalent to the target concept, i.e., that will label all instances correctly. The oracle either responds that the hypothesis is exactly correct or gives to the learner a counterexample, which is an instance on which h is wrong. An MQ oracle receives from the learner an instance x and provides x's label. It is similar to an active learning model, except that in the MQ model, the instances can be arbitrary while in active learning, the instances must come from a pre-specified set.

pool-based

pool-based

Here we consider the passive (where all instances are labeled) proper learning of k-unions of axis-parallel boxes, in a bounded, discretized, d-dimensional instance space $\{0, \ldots, t-1\}^d$.

~~PROPOSITION~~ 5. *With over-labeling, discrete k-unions of boxes can be exactly learned with $O(k)$ EQs and $O(kd \log t)$ MQs and time polynomial in the number of queries.*

Proof: Using fine-grained labels for k distinct fine-grained hypotheses (each using one box), one can exactly learn each box individually with one EQ (to get a positive instance) and $O(d \log t)$ MQs (for binary search to find the box's $2d$ boundaries), for a total of $O(k)$ EQs and $O(kd \log t)$ MQs and time polynomial in the number of queries. □

This contrasts with a result from Bshouty and Burroughs [**?**] that one cannot exactly properly learn unions of $\leq k$ axis-parallel boxes (*k-unions*) when (constant) $d > 2$ unless P = NP. I.e., while one can learn k-unions with $O(d \log k)$-unions, one cannot efficiently learn k-unions with k-unions if P $\neq$ NP. Note that our positive result for over-labeling works for non-constant d, while the hardness result for direct proper learning holds for constant d.

4 Approach

We now present our method for performing the learning task outlined in Section **??**. We refer to our method as HAL, for Hierarchical Active Learner. The high-level steps of our algorithm are given in Algorithm **??**.

HAL iteratively purchases labels at each level of the labeling tree in batches, where the proportion of labels

purchased at each level is specified by vector $\mathbf{p}$. In our experiments, we explore how different settings for $\mathbf{p}$ impact performance. In the step PURCHASE($b\,p_i, i, C^*(x)$), the system purchases $b\,p_i$ dollars worth of label vectors defined up to level i of the labeling tree, where the instances to be labeled are chosen actively via uncertainty sampling relative to classifier $C^*(x)$ (discussed below). [2] Because label vectors are defined up to level i, they include labels for all levels $m \leq i$, (Section **??**). LABELMAP(E', m, j) then creates individual labeled examples for class m at level j corresponding to the given label vectors E', for training the classifiers $C_{i,j}$.

HAL then trains a probabilistic binary classifier $C_{i,j}$ for each class j at level i, using the machine learning algorithm L. $C_{i,j}(x)$ denotes $C_{i,j}$'s estimate of the probability that an arbitrary example x is positive for class j at level i (e.g., $C_{2,\text{"Lake"}}(x)$ is the probability that instance x is a Lake). The choice of L depends on the particular learning task (we use Gradient Boosted Regression Trees, Logistic Regression and Conditional Random Fields in our experiments).

A key problem for HAL is how to combine the classifiers $C_{i,j}$ into an ensemble classifier for the coarse-grained level-1 concept. In principle, the level-1 concept is a disjunction over the concepts j at any given level i. However, modeling the $C_{i,j}$ for a given i explicitly as a disjunction can be challenging, due to dependencies across the different level-i classifiers (which are trained on related sets of data $E_{i,j}$). In preliminary experiments, we explored combining the classifiers with a noisy-or model (i.e., assuming independence), a linear model [**?**], or taking a p-norm across all j. None of these approaches outperformed a simple approach of simply taking a maximum. Thus, we define:

$$C^*(x) = \max_{i,j} C_{i,j}(x) \tag{4.2}$$

as the output ensemble classifier.

Finally, when purchasing examples with active learning, HAL uses uncertainty sampling. The uncertainty at level i is measured with respect to the ensemble classifier $C^*(x)$. Specifically, we define the uncertainty $u_i(x)$ of the label for example x for level i as:

$$u(x) = 0.5 - |C^*(x) - 0.5| \ . \tag{4.3}$$

5 Experiments

We first examine the benefit of our model in a synthetic binary classification task. We then perform a similar analysis to demonstrate the advantage of our approach in two real-world data sets. We then analyze the impact of varying labeling costs on performance. Finally, we present a multi-level classifier learned with HAL, and study how its performance varies with the relative labeling cost between coarse (root) and fine (lower-level) labels.

[2] The *number* of examples purchased at each level will vary inversely with the cost of labels at that level, and we explore how varying label cost ratios impacts performance of the algorithm for a given budget level in our experiments (see Section **??**).

Algorithm 1 Method for learning the concept at the root of a given labeling tree. Purchase and LabelMap are separate subroutines (see text).

function HAL(Unlabeled examples U, labeling tree T, machine learner L, budget B, per-iteration budget b, Purchase proportions $\mathbf{p} = (p_1, \ldots, p_k)$ with $||\mathbf{p}||_1 = 1$ and $p_i \geq 0$)
  $E_{i,j} \leftarrow \emptyset$ ▷ binary-labeled train set for level i, label j
  Initialize $C_{i,j}$ for all i, j
  while $B > b$ **do**
    $B \leftarrow B - b$
    for all Level $i \in T$ **do**
      $E' \leftarrow$ PURCHASE($b\,p_i, i, C_{i,*}$)
      for all Level $m \leq i$ **do**
        for all Class j in Level m **do**
          $E_{m,j} \cup =$ LABELMAP(E', m, j)
        end for
      end for
    end for
    for all Level $i \in T$ **do**
      for all Class j in Level i **do**
        $C_{i,j} \leftarrow$ Train L on $E_{i,j}$
      end for
    end for
  end while
  return Ensemble classifier $C^*(\{C_{i,j}\}, \mathbf{p})$
end function

5.1 Synthetic Dataset Experiments First we assess the advantage provided by using fine-grained label data in a synthetic binary classification task. In this dataset the sole feature is a single continuous value $x \in [0, 18)$. The positive instances are all points in the 9 level-3 intervals $\{[0,1), [2,3), [4,5), \ldots, [16,17)\}$. We then define the level-2 intervals by taking the union of consecutive triples of the level-3 intervals: $\{[0,1), [2,3), [4,5)\}$, $\{[6,7), [8,9), [10,11)\}$, and $\{[12,13), [14,15), [16,17)\}$. Finally, the level-1 label (positive versus negative) is the union of the three level-2 labels. The goal is to learn the level-1 label of '+' versus '−'. We use as our base learner the Gradient Boosted Regression Tree (GBRT) [**?**], which is an ensemble of regression trees. We set the maximum depth in GBRT to be 1 so that each tree maps to a continuous interval. For the fine grained learner, the classifers at level 3 are combined with the level-2 classifier and then the coarse-level

classifier using Equation **??**. Because GBRT can be a union of intervals, the classifiers at each level should be expressive enough to capture the target concept.

We chose $n = 30$ trees, learning rate $\lambda = 0.9$ and sample rate $r = 0.8$ for our GBRT learners. Figure **??** shows the learning curve of coarse and combined fine-grain method on both active learning and passive learning[3]. Each learning curve is averaged over 50 rounds. Learners started with 150 initial training examples to ensure that all learners had initial instances in all fine-grained classes. In each step, each learner can purchase a label of one instance from a pool of 1000 instances. We simulated noise by flipping the labels for 10% of the examples (choosing noisy fine-grained labels uniformly at random). The classifiers are then tested against a set of 1800 examples uniformly distriubuted in $[0, 18)$.

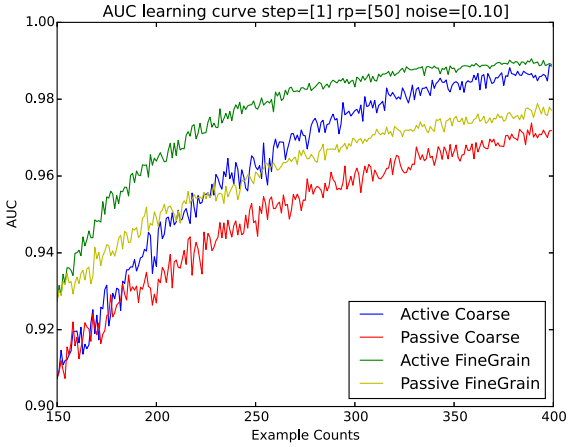

Figure 3: Learning curve on synthetic dataset.

From Figure **??**, we can see that the two fine-grained methods have better AUC (area under precision-recall curve) compared to their coarse opposites. Similarly, the two active curves are better than the two passive curves. Among the four curves, the active fine-grain method outperforms the rest. The accuracy difference between the active coarse and active fine-grained classifiers after 400 training examples is statistically significant ($p < 0.05$, Fisher Exact Test).

5.2 Document Classification Experiments We now present results from a document classification task. The RCV1 data set [**?**] contains 23149 training documents and 781265 test documents labeled with a 117-node hierarchy of Reuters Topics categories. Each document was represented as a sparse feature vector in cosine-normalized log TF-IDF [**?**] format. Our coarse-based and fine-based learners used logistic regression [**?**] as the base learner, with L2 regularization ($\beta = 0.1$).

We used ECAT as the coarse-grained class, which contains 119920 positive examples and 33 sub-classes in multiple levels underneath it. We started with a seed set of 2000 randomly-selected instances and ran 90 iterations of active learning with 200 labels acquired per iteration from the pool of the remaining 22949 instances using both coarse-based and fine-based methods. Since the articles are not uniformly distributed across the fine-grained categories, we filtered out all fine-grained classes with fewer than 10 instances. We tested the model on the entire test set, which is independent from the initial seed training set and candidate pool set. Each learning curve is the average of 10 rounds.

Similar to Section **??**, we built four learning curves from the results, which appear in Figure **??**. In this figure, we can see that each active method converges to its upper limit faster than the corresponding passive method. In addition, we found that over-labeling is effective: the fine-grained methods have higher performance limits than the coarse methods. Among the four methods, active learning with the fine-grained method has the largest AUC. The accuracy difference between the active coarse and active fine-grained classifiers is statistically significant ($p < 0.05$, Fisher Exact Test).

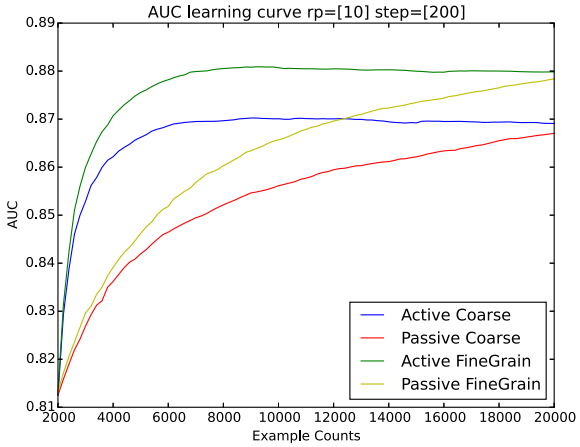

Figure 4: Learning curve on ECAT in RCV1 dataset.

5.3 Sequence Tagging Experiments Our third experiment focuses on entity recognition in sequences of text. We took OCR results from digitized editions of the *Richmond Daily Dispatch* from November 1860 through December 1865 that had been tagged with XML labels according

[3]Our passive learners are trained with the same number of training instances as our active learners, but the instances are chosen randomly rather than via uncertainty sampling.

to a two-level hierarchical labeling scheme [**?**]. The dataset we used consists of 375026 manually labelled organization names across 1384 newspaper articles. These names are further categorized into a pool of 82 fine-grained categories, like bank names, railroad names and government agency names. Thus, the coarse-grained labels were "organization" versus "not organization" and each fine-grained label is, e.g., "bank" versus "not bank".

In the *Dispatch* experiment, we used conditional random fields (CRFs) [**?**] as the base learner. We trained CRFs using standard 2–3 letter prefix, postfix, capitalization and numerical features. We evaluate the trained CRF by performing the Forward-Backward algorithm [**?**] on a new sentence s to get an estimate of the probability that each token $x \in s$ is an organization. Evaluating the set of tokens above varying thresholds on this probability yields the precision-recall curves we use for evaluation.

We compared training fine-based classifiers via active learning, fine-based classifiers via passive learning, coarse-based via active learning, and coarse-based via passive learning. First, we set aside 59120 sentences for the test set and 14380 sentences for candidate set. The experiment starts with a initial training set of 400 sentences.

Our experiments proceeded in 100 iterations, with batch sizes of 50 sentences each iteration. The passively trained coarse-based learner and the passively trained fine-based learner each purchase 50 randomly selected sentences, whereas the actively trained coarse-based learner and the actively trained fine-based learner each select 50 sentences via uncertainty sampling. When computing the uncertainty of a sentence s, we took the maximum uncertainty across all tokens in s. Using the process described earlier of utilizing CRFs for classification, we evaluated the four derived classifiers on the test set of 10% of all the sentences.

Figure **??** shows the learning curve of the four approaches. The differences between each pair of results is statistically significant; e.g., at example counts of 5000, the differences between the AUC values for all pairs of results are significant at a level of $p < 0.0001$ (Fisher Exact Test). Unsurprisingly, we see that the AUC for the passive approaches are much lower than those for the active approaches. More interestingly, the benefit of using fine-grained labels is more significant for active learners versus passive ones.

6 Discussion

Our previous results demonstrate the advantages offered by purchasing fine-grained labels in an active learning context to improve performance of the resulting classifier. So far, we have ignored differences in cost between label types. As discussed above, in practice fine-grained labels are likely to be more expensive to obtain than coarse-grained labels.

New cost-sensitive methods for active over-labeling are an item of future work. In this section, we provide an analysis indicating that active over-labeling is likely to provide value at varying ratios of cost between coarse and fine-grained labels. In particular, we examine the numbers of instances labeled at the fine and coarse levels in the *Richmond* experiments needed to achieve specific AUC scores. The ratio of such numbers will give us an estimate of what ratio of cost of fine-grained labels to coarse-grained labels justifies the use of an active over-labeling approach.

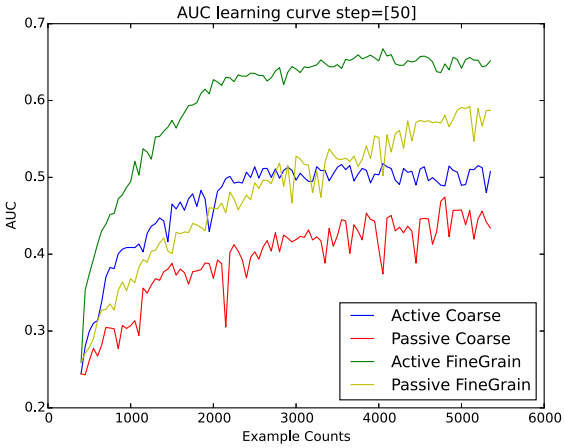

Figure 5: Learning curve of NER on Richmond dataset.

To begin, in Figure **??**, we compare the number of labeled training sentences needed to achieve particular AUC scores for our fine-based and coarse-based active learners. For example, to achieve a score of $AUC \approx 0.50$, we need approximately 1000 sentences with fine-grained labels, and about twice that in coarse-grained labels. Therefore, if our target AUC score is 0.5 and the cost to acquire a fine-grained label is more than twice that of coarse-grained, we should not use the fine-based classifier. Instead, if fine-grained labels cost less than twice that of coarse-grained, then exclusive use of the fine-based classifier is preferred.

The above cost comparison results are based on exclusive use of fine-based learners versus exclusive use of coarse-based learners. To generalize the analysis, we now present results using HAL to utilize both fine-grained and coarse-grained classifiers in concert. Depending on how many coarse- and fine-grained labels are purchased (i.e., depending on the purchase proportions vector $\mathbf{p}$) we get a family of such hybrid algorithms. As part of our results, we perform a cost analysis on this family.

We evaluate HAL allocating fractions[4] of labels $p_2 \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ of the 50 purchases to the fine-

[4]Since the *Richmond* data has two levels, our purchase proportions vector $\mathbf{p} = (1 - p_2, p_2)$.

based learning algorithm and the remainder to the coarse-based one. Figure **??** shows our results for different values of p_2. For reference, we also include curves for the actively trained coarse-based (0%) and fine-based (100%) classifiers presented in Figure **??**. We see that for the same AUC score, the larger p_2 is, the fewer training instances are required.

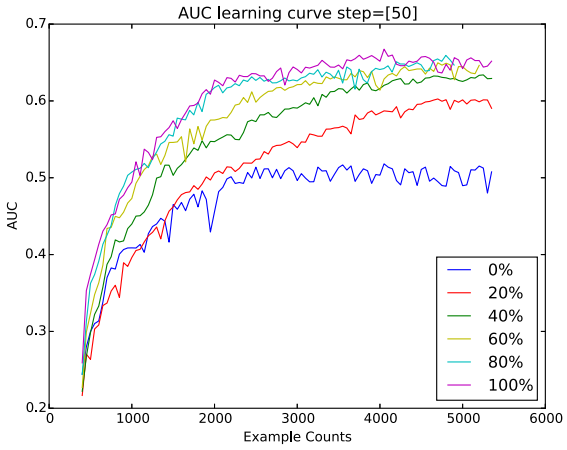

Figure 6: Learning curve using active learning with varying fine grained ratio on Richmond dataset.

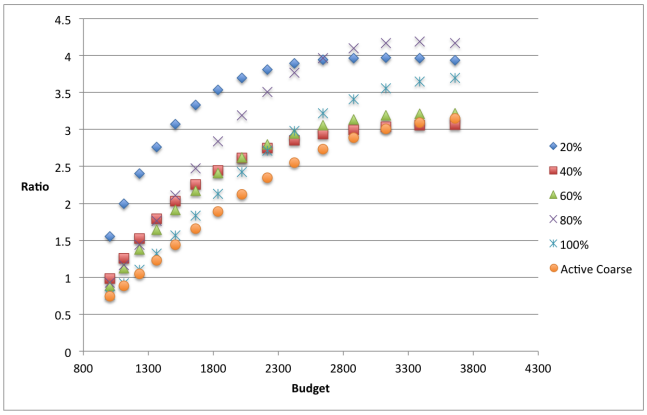

Figure 7: Cost ratios versus total budget for each algorithm.

We now use the results from Figure **??** to determine at what ratio of fine-grained to coarse-grained label cost each algorithm is preferred. For each value of p, for each value v of AUC score ranging from 0.3 to 0.5, we determined from a quadratic fit to the learning curves in Figure **??** the number of instances N_v^C required for the pure coarse-based algorithm to achieve score v and the number N_v^p needed for HAL with $p_2 = p$ to achieve score v. We then combined these values in a weighted fashion to determine the *cost ratio* for algorithm p at score v:

$$r_v^p = \frac{N^C - (1-p)\, N_v^p}{p\, N_v^p} \quad . \tag{6.4}$$

If the cost for a fine-grained label is more than r_v^p times the coarse-grained cost, then it is more cost effective to use (for score v) a pure coarse-based classifier and not utilize fine-grained labels at all. In contrast, if the fine-grained label cost is less than r_v^p times the coarse-grained cost, then for target score v, the hybrid algorithm is preferred over the pure coarse-based one.

Figure **??** shows the cost ratios r_v^p for each value of $p = p_2$, plotted as points (N^C, r_v^p). The curves show the cost break points for each hybrid algorithm when compared to the pure coarse-based learner, as a function of the total budget available (N^C). For each curve (value of p_2) and total budget for purchases, if the actual ratio of cost of fine-grained to coarse-grained labels is $>$ r_v^p, we should solely use the coarse-based learner, since the cost to acquire fine-grained labels is too high. For example, for $p_2 = 0.8$ and a budget of 3000, the break point is around 4. If fine grained-labels cost more than 4 times as coarse-grained labels, we should exclusively use the coarse-based learner. Instead, if the cost ratio is smaller than that, then the $p = 0.8$ hybrid classifier should perform better with the same budget. Unsurprisingly, we have that as the budget increases, a higher cost fine-grained label is more affordable.

interval

		diff		
		min	max	mean
bandit_e_greedy	1.0			
fixed_fine_ratio	0.0	0.000663	0.001320	0.000894
	0.1	0.008732	0.015742	0.013793
	0.2	0.000000	0.004312	0.002608
	0.3	0.000543	0.003502	0.002396
	0.4	0.000647	0.003114	0.002421
	0.5	0.000607	0.005881	0.002507
	0.6	0.000000	0.008122	0.002535
	0.7	0.000331	0.010690	0.002467
	0.8	0.000343	0.011373	0.002383
	0.9	0.000000	0.013213	0.002420
	1.0	0.000000	0.014509	0.002568

RCV1

richmond

		diff				diff_ratio				rank	
		min	max	mean	std	min	max	mean	std	min	max
bandit_e_greedy	1.0										
fixed_fine_ratio	0.0	0.000552	0.004157	0.002114	0.001312	0.000640	0.004835	0.002430	0.001519	1	5
	0.1	0.000000	0.016422	0.010424	0.006124	0.000000	0.018741	0.011931	0.006993	0	11
	0.2	0.000000	0.005630	0.002995	0.001658	0.000000	0.006425	0.003435	0.001898	0	9
	0.3	0.000000	0.013060	0.003892	0.004019	0.000000	0.015190	0.004491	0.004682	0	7
	0.4	0.000926	0.017702	0.004893	0.005702	0.001063	0.020590	0.005656	0.006640	1	4
	0.5	0.000215	0.024232	0.006955	0.007704	0.000246	0.028185	0.008035	0.008971	1	8
	0.6	0.000000	0.024978	0.006458	0.008802	0.000000	0.029053	0.007475	0.010236	0	9
	0.7	0.000000	0.028312	0.008082	0.009756	0.000000	0.032931	0.009348	0.011348	0	8
	0.8	0.000000	0.032880	0.008392	0.011638	0.000000	0.038244	0.009717	0.013535	0	9
	0.9	0.001198	0.029841	0.009283	0.010665	0.001369	0.034709	0.010735	0.012402	1	9
	1.0	0.002490	0.034787	0.011305	0.011503	0.002846	0.040462	0.013062	0.013395	6	11

		diff				diff_ratio				rank	
		min	max	mean	std	min	max	mean	std	min	max
bandit_e_greedy	1.0										
fixed_fine_ratio	0.0	0.010160	0.029180	0.020783	0.006229	0.019101	0.046240	0.034304	0.008802	2	2
	0.1	0.056396	0.166203	0.128314	0.039921	0.110621	0.263371	0.211197	0.056610	11	11
	0.2	0.002899	0.094514	0.060917	0.030579	0.005451	0.149770	0.098559	0.057535	1	1
	0.3	0.000000	0.078012	0.049456	0.030919	0.000000	0.124731	0.079901	0.048976	0	0
	0.4	0.004228	0.069065	0.039901	0.024058	0.007670	0.109442	0.064679	0.037396	3	3
	0.5	0.000714	0.053060	0.030038	0.017809	0.001296	0.084081	0.048884	0.027747	2	2
	0.6	0.000524	0.037437	0.024454	0.012501	0.000950	0.059856	0.040034	0.019514	1	1
	0.7	0.000000	0.029230	0.017760	0.008952	0.000000	0.046916	0.029307	0.014621	0	0
	0.8	0.004903	0.021512	0.013515	0.006004	0.007883	0.040440	0.022732	0.010615	2	2
	0.9	0.001047	0.024052	0.010309	0.007900	0.001900	0.045215	0.017588	0.014324	1	1
	1.0	0.000000	0.031150	0.007622	0.010009	0.000000	0.058560	0.013461	0.018903	0	0

7 Related Work

To our knowledge, our experiments are the first to demonstrate how leveraging fine-grained label information can improve the accuracy of a coarse-grained (root-level) classifier, and the first investigation into active learning in a hierarchical setting where label acquisition cost can vary.

Previous work in text classification has considered using hierarchies of labels to improve a fine-grained classifier, through techniques that back off to coarse levels of the hierarchy when fine-grained data are sparse [**?**]. By contrast, we present novel techniques that work in the opposite direction, utilizing selectively acquired fine-grained labels to improve classification over coarse categories. In named entity recognition (NER), some recent work has targeted fine-grained entity categories [**?** **?**] or hierarchies [**?**]. Our work differs from this previous work in that we focus on active learning under variable label acquisition costs. Our experiments illustrate that our active approach outperforms passive learning on the NER task, and we demonstrate how the relative cost of obtaining finer-grained labels impacts which NER approach is most appropriate to use.

Our approach builds on a variety of previous work in active learning. We focus on "pool-based" active learning, in which a learner selects instances from a pool of unlabeled data to be labeled by an oracle. When acquiring labels is costly, active learning can reduce the expense by requesting only a relatively small subset of the most informative labels [**?**]. One criterion used for selection of instances to label is to choose those that reduce uncertainty. In our case, uncertainty is measured in terms of the confidence of output values (e.g., Meriaudo [**?**]); other measures include uncertainty in the parameters of probabilistic models [**?**] or the size of a model's decision boundary [**?**].

In previous work, active learning has also been shown to reduce sampling bias by utilizing the hierarchical structure of input features [**?** **?**]. By contrast, our work focuses on active learning over hierarchically structured output labels.

Luo et al. [**?**] looked at active learning in to perform structure prediction, e.g., to predict a segmentation of an image or a parse tree of a sentence. While the predictions their algorithms made are structured in nature, it is not similar to our work, which predicts labels according to a fixed hierarchy known *a priori* and varying costs.

8 Conclusions

Hierarchical labeling schemes are increasingly common in a variety of applications. Our results demonstrate that fine-grained label data (labels specified at nodes removed from the root of a labeling tree) can be used to improve precision of a classifier for the coarse-grained (root) concept. However, it is likely that such fine-grained labels will be more expensive to obtain. We defined a new active learning approach, *active over-labeling*, to address this scenario, created a family of hybrid algorithms to actively make label purchase decisions, empirically evaluated this family of algorithms, and analyzed the relative cost points at which one algorithm is preferred over another at various budget levels.

There are many avenues of future work. More sophisticated algorithms, e.g., those that dynamically adjust the mix

of different levels of labels, could be even more effective in improving classifier performance at lower budgets. Such algorithms could benefit from more detailed cost analyses. Finally, it would be interesting to consider other hierarchically labeled data sets, e.g., that labeled by the Gene Ontology [**?**].

References

[1] "The Gene Ontology," [Online]. Available: geneontology.org

[2] M. Fleischman and E. Hovy, "Fine-grained classification of named entities," in *Proceedings of 19th Int. Conf. on Comp. Linguistics-Volume 1*, pp. 1–7, 2002.

[3] X. Ling and D. S. Weld, "Fine-grained entity recognition." in *AAAI*, 2012.

[4] M. A. Yosef, S. Bauer, J. Hoffart, M. Spaniol, and G. Weikum, "Hyena: Hierarchical type classification for entity names." in *COLING (Posters)*, 2012, pp. 1361–1370.

[5] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," *Proc. of 43rd ACL*, pp. 363–370, 2005.

[6] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research," *JMLR*, vol. 5, pp. 361–397, 2004.

[7] L. G. Valiant, "A theory of the learnable," *Commun. ACM*, vol. 27, no. 11, pp. 1134–1142, Nov. 1984.

[8] D. Angluin, "Learning regular sets from queries and counterexamples," *Inform. Comput.*, vol. 75, pp. 87–106, 1987.

[9] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Learnability and the Vapnik-Chervonenkis dimension," *J. ACM*, vol. 36, no. 4, pp. 929–965, 1989.

[10] W. J. Masek, "Some NP-complete set cover problems," unpublished manuscript, MIT Laboratory for Computer Science.

[11] N. Bshouty and L. Burroughs, "On the proper learning of axis-parallel concepts," *JMLR*, vol. 4, pp. 157–176, 2003.

[12] L. Breiman, "Stacked regressions," *Machine Learning*, vol. 24, pp. 49–64, 1996.

[13] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[14] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Proc. & Man.*, 1988.

[15] D. Cox, "The regression analysis of binary sequences," *Jour. of the Royal Stat. Soc.*, vol. 20, pp. 215–242, 1958.

[16] University of Richmond Libraries, "*Richmond Daily Dispatch, 1860–1865*," 2014.

[17] C. Sutton, "An introduction to conditional random fields for relational learning," *Graphical Models*, vol. 7, p. 93, 2006.

[18] A. Culotta, "Confidence estimation for information extraction," *HLT-NAACL* , 2004.

[19] A. McCallum, R. Rosenfeld, T. M. Mitchell, and A. Y. Ng, "Improving text classification by shrinkage in a hierarchy of classes." *ICML*, pp. 359–367, 1998.

[20] N. Rubens, D. Kaplan, and M. Sugiyama, "Active learning in recommender systems," *Recommender Systems Handbook*, pp. 1–31, 2011.

[21] A. Merialdo, "Improving Collaborative Filtering For New-Users By Smart Object Selection," *ICMF*, 2001.

[22] T. Hofmann, "Collaborative filtering via gaussian probabilistic latent semantic analysis," in *SIGIR '03*, p. 259, 2003.

[23] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," *ICML*, pp. 839–846.

[24] S. Dasgupta and D. Hsu, "Hierarchical sampling for active learning," *ICML '08*, pp. 208–215, 2008.

[25] C. Symons, N. Samatova, R. Krishnamurthy, B. Park, T. Umar, D. Buttler, T. Critchlow, and D. Hysom, "Multi-Criterion Active Learning in Conditional Random Fields," *ICTAI'06*, pp. 323–331, 2006.

[26] W. Luo, A. Schwing, and R. Urtasun, "Latent structured active learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.