Learning Hierarchically Decomposable Concepts with Active Over-Labeling

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, etc. 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 vs. non-location) by seperately learning multiple finer-grained classifiers (e.g., museum vs. 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." 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. Finally, because finer-grained labels may be more expensive to obtain, we also present a cost-sensitive active learner that uses a multi-armed bandit approach to dynamically choose the label granularity to target, and show that the bandit-based learner is robust to differences in label cost and labeling budget.

1 Introduction

Several 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 (GO) [1], a biological sequence can be labeled with multiple terms from a hierarchical scheme.

While some recent work has considered classification into fine-grained (low-level) categories [2, 3] or hierarchies [4], the majority of classification tasks in practice target a small number of labels. NER, for example, typically targets a small set of coarse-grained labels such as Location, Organization, and Person [5].

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 2.2 and establish in our experiments, fine-grained labeling information can significantly improve the accuracy of a classifier aimed at a coarse-grained task. Further, actively soliciting informative fine-grained labels, rather than passively sampling them, provides an additional performance 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 performance 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, we expect that obtaining a finer-grained label will often be more costly than obtaining a coarse-grained label. For example, in GO, 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, localization, or one of 22 other labels, which requires more expertise and effort to obtain. Thus, we also present a method that uses a multi-armed bandit approach to dynamically balance cost and estimated benefit when choosing the level of granularity to query. We then show that the bandit approach is robust to changes in label cost.

The rest of this paper is organized as follows. In Section 2, we define our learning setting and give intuition as to why our approach offers advantages in this context. In Section 3 we describe our algorithms in our new model, and we then present experimental results in Section 4. We then cover related work in Section 5 and conclude in Section 6.

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 $f : \mathcal{X} \rightarrow \mathcal{Y} = \{0, 1\}$. 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) [6] is shown in Figure 1.

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 1, 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.

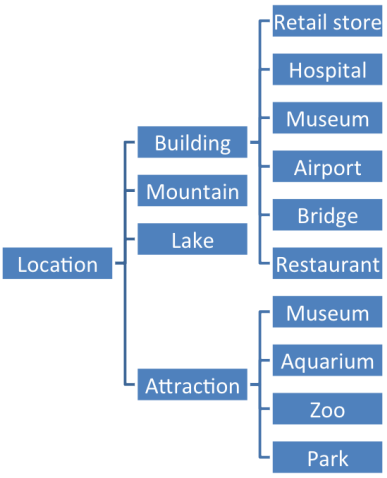


Figure 1: An example labeling tree based on Reuters Corpus Volume I (RCV1) [6].

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').

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 2. 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 2, 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 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.

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

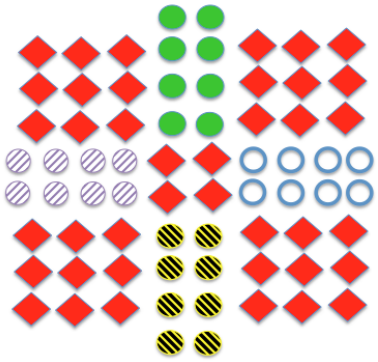


Figure 2: An example of the potential usefulness of learning multiple fine-grained concepts to support the learning of a single coarse-grained one. Negative instances are diamonds and positive ones are circles.

2.2 Learning-Theoretic Advantages To formalize the intuitive advantage described in Section 2.1, we present some simple theoretical results that immediately follow from the literature. This section's purpose is not to advance the state of the art in learning theory, but to highlight the advantages that over-labeling can provide. We present observations in both the **probably approximately correct (PAC)** [7] and **exact** [8] models of learning. The results below focus on the case of learning concepts that are unions of axis-parallel boxes over real and ordinal feature spaces.

2.2.1 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 $\mathcal{D}$.

2.2.1.1 Computational Complexity In the context of computational complexity, we consider the case of **proper learning**[2], 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 learning concepts that are unions of k axis-parallel boxes in $\mathbb{R}^d$. This task is not properly PAC-learnable (i.e., learning $\mathcal{C}$ using $\mathcal{C}$) if RP $\neq$ NP.

OBSERVATION 1. *The class of k-unions of axis-parallel boxes in $\mathbb{R}^d$ is not properly PAC-learnable unless RP = NP.*

Proof Sketch: From Theorem 3.1.1 of Blumer et al. [9], concept class $\mathcal{C}$ is properly PAC learnable iff there exists a randomized polynomial-time algorithm to find a hypothesis from $\mathcal{C}$ consistent with a sufficiently large labeled training sample $\mathcal{X}$ (called the **consistent hypothesis problem**). It is known to be NP-hard [9, 10] 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. □

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 2. 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 '$-$'.

OBSERVATION 2. *In the over-labeling setting, k-unions of boxes from $\mathbb{R}^d$ is properly PAC-learnable.*

Proof Sketch: Let m be the number of labeled training instances labeled by the target concept of some k-union of boxes. For each sub-concept, a consistent sub-hypothesis (single bounding box) can be learned from the fine-grained labels in time $O(dm)$. The learner can learn each of the k sub-concepts separately and output their union in time $O(kdm)$, and this union will be consistent with all the labeled examples. Thus, the consistent hypothesis problem can be solved in time polynomial in d, k, and m. Blumer et al. [9] show that, if m is sufficiently large, then a consistent hypothesis h will meet the PAC criteria. Specifically, Equation 2.1 below gives sufficient conditions on m for error bound ϵ and bound δ on probability of failure. Because the over-labeling approach is finding a separate consistent hypothesis for each of the k fine-grained labels, we apply Equation 2.1, but reduce ϵ and δ each by a factor of k to account for this. This yields a polynomial bound on m. □

2.2.1.2 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 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. [9] that says if the algorithm efficiently finds a hypothesis from $\mathcal{C}$ that is consistent with U, which is drawn iid from fixed distribution $\mathcal{D}$ and is of size at least

$$m(\epsilon, \delta) = \max\left(\frac{2}{\epsilon}\log\frac{2}{\delta}, \frac{8D}{\epsilon}\log\frac{13}{\epsilon}\right) \tag{2.1}$$

(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 as possible labels of instances of U and still guarantee a hypothesis consistent with all of U (including the yet unlabeled ones), which would yield a 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

[2]Note that the negative results described below for both exact and PAC learning are only for proper learning. One can get positive results for these cases by allowing a logarithmic increase in the number of boxes used, by applying the set cover approximation algorithm.

instance $x \in U$ is purchased, it returns '+' if x lies in one of the k target intervals and '$-$' otherwise. When a fine-grained 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 observations, we bound the number of purchases needed in each labeling scheme to find a consistent hypothesis. Since the total number of instances needed for PAC learning (per Equation 2.1) differs between them (due to different VC dimensions), in the next two observations, 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.

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

Proof Sketch: 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. □

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

Proof Sketch: 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 2.1 with VC dimension [9] $D = 2k$ 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 VC dimension [9] $D = 2$, 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 $1/\epsilon$.

Simply put, the advantage that the fine-grained approach has comes from the fact that, for positively-labeled instance $x \in U$, the fine-grained label indicates the interval that x lies in, while in the coarse-grained approach, the label is simply '+'. The distinct fine-grained label given by each interval allows for a binary search for interval boundaries, hence the logarithmic dependence on m_f. In contrast, the homogeneous '+' label across all intervals for the coarse-grained labels can force a number of purchases linear in m_c.

2.2.2 Exact Learning with Active Over-Labeling We now 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 $h \in \mathcal{C}$ 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. This is similar to a pool-based active learning model, except that in the MQ model, the instances can be arbitrary, while in pool-based active learning, the instances must come from a pre-specified set.

We consider proper learning of k-unions of disjoint axis-parallel boxes, in a bounded, discretized, d-dimensional instance space $\{0, \ldots, t-1\}^d$.

OBSERVATION 5. *With over-labeling, disjoint 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 Sketch: Using fine-grained labels for k distinct fine-grained hypotheses (each using one box), one can exactly learn each box j individually with one EQ (to get an instance in box j) and $O(d \log t)$ MQs (for binary search to find the box j's $2d$ boundaries), for a total of $O(k)$ EQs and $O(kd \log t)$ MQs and polynomial time. □

This contrasts with a result from Bshouty and Burroughs [11] that one cannot exactly properly learn k-unions of axis-parallel boxes 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 even for constant d.

3 Approach

We now present our method for performing the learning task outlined in Section 2. We refer to our method as HAL, for Hierarchical Active Learner. The high-level steps of our algorithm are given in Algorithm 1.

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 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 $C^*(x)$[3] (discussed below). Because label vectors are defined up to level i, they include labels for all levels $m \leq i$ (Section 2). 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 [12], or taking a p-norm across all j. None of these approaches outperformed a simple approach of simply taking a maximum. Thus, we define:

$$C^*(x) = \max_{i,j} C_{i,j}(x) \tag{3.2}$$

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:

$$u(x) = 0.5 - |C^*(x) - 0.5| \ . \tag{3.3}$$

[3]The **number** of examples purchased at each level will vary inversely with the cost of labels at that level, and later we explore how varying label cost ratios impacts performance of the algorithm for a given budget level.

Algorithm 1 Method for learning the concept at the root of labeling tree T. See text for Purchase and LabelMap.

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

3.1 Dynamically Adapting Purchase Proportions

HAL takes as input a vector of purchase proportions, specifying how much of the budget should be spent acquiring labels at each given level of granularity in the hierarchy. The cost of labeling an example can vary across levels of the hierarchy, and as we show in Section 4.5, the relative benefit of labels from a given level in the hierarchy often changes as learning progresses. Thus, we desire a cost-sensitive approach that dynamically adapts the purchase proportions.

We formulate the task of choosing which level of granularity to purchase next as a multi-armed bandit problem, and solve it using the ϵ-greedy bandit algorithm [14] (BANDIT). For clarity, we focus on dynamically choosing between two strategies corresponding to purchase proportions, $\mathbf{p}$ and $\mathbf{p}'$, but the generalization to more strategies is straightforward.

BANDIT iteratively chooses strategies, and uses a running average of the reward observed for each strategy to guide its choices. For active learning, a natural way to define reward in round j is in terms of observed model change:

$$g_j = \frac{1}{\|X\|} \sum_{x_i \in X} \log\left(|f_{j-1}(x_i) - f_j(x_i)|\right) \ , \tag{3.4}$$

where X is the set of withheld *unlabeled* examples, and $f_j(x_i)$ is HAL's output for the input x_i after the jth iteration.

However, our problem setting has special characteristics that make the gain equation above unsuitable. In particular, this gain score is not stationary as is assumed in the generic

BANDIT algorithm. Instead, as the number of purchased examples increases and HAL gradually fits the data, the model becomes less likely to change. This means the running average of the gain will slowly decrease. As a result, BANDIT will disproportionately favor arms it has not played recently (whose average gains have not recently been updated). In our preliminary experiments, we found that these characteristics led the generic BANDIT approach to thrash between arms in cases where sticking with one arm for longer was a stronger strategy.

To adapt BANDIT to our setting, instead of modeling each purchase strategy as an arm, we instead use two arms: (1) maintain the same strategy as before, and (2) switch strategies. The reward from the first arm is always zero. The reward from the second arm depends on the difference in gain before and after switching, defined as:

$$r_j = \begin{cases} -\frac{g_j}{|g_j|}, & \text{if } \mathbf{p} \to \mathbf{p}' \\ \frac{g_j}{|g_j|}, & \text{if } \mathbf{p}' \to \mathbf{p} \\ 0, & \text{if } \mathbf{p} \to \mathbf{p} \\ 0, & \text{if } \mathbf{p}' \to \mathbf{p}' \end{cases} . \tag{3.5}$$

4 Experiments

We begin by describing the three datasets we will consider: a synthetic binary classification task, and two real-world data sets in document classification and sequency tagging. We the present our results. We first show that active over-labeling improves accuracy over standard active or passive learning. We then study how HAL's performance performance varies with the relative labeling cost between coarse (root) and fine (lower-level) labels and overall budget (number of training examples). Finally, we show how our adaptive bandit-based over-labeling scheme, BANDIT, is robust to changes in labeling cost and budget.

4.1 Synthetic Dataset 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 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 $\{[12,13),[14,15),[16,17)\}$. 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) [13], which is an ensemble of regression trees. We set the maximum depth in GBRT to be 1 so that each tree maps to an 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 3.2. 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. For this task, learners started with 100 initial training examples to ensure that all learners had initial instances in all fine-grained classes. In each round of iteration, each learner purchases a label of one instance from a pool of 10000 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 8000 examples uniformly distriubuted in $[0, 18)$.

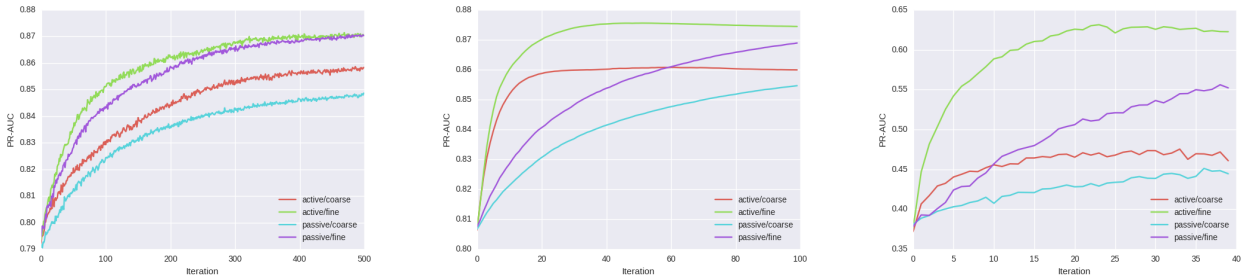
4.2 Document Classification The RCV1 data set [6] contains 23149 training documents and 781265 test documents labeled with a 117-node hierarchy of Reuters Topics categories. Each document was represented in cosine-normalized log TF-IDF [15] format. Our coarse-based and fine-based learners used logistic regression [16] as the base learner, with L2 regularization ($\lambda = 0.1$).

We used ECAT as the coarse-grained class, which contains 119920 positive examples and 33 sub-classes in multiple levels underneath it. For this task we started with a seed set of 2000 randomly-selected instances and ran 100 iterations of active learning with 120 labels acquired per iteration from the pool of the remaining 22949 instances using both coarse-based and fine-based methods. To eliminate trivially small classes, 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 50 rounds.

4.3 Sequence Tagging 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 to a two-level hierarchical labeling scheme [17]. 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) [18] 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 [19] 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



(a) Synthetic dataset (b) Document classification (c) Sequence tagging

Figure 3: Learning curves comparing combinations of fine/coarse and active/passive PR-AUC

learning, fine-based classifiers via passive learning, coarse-based via active learning, and coarse-based via passive learning. First, we set aside 20000 sentences for the test set and 10000 sentences for candidate set. The experiment starts with a initial training set of 1600 sentences. For this task our experiments proceeded in 40 iterations, with batch sizes of 100 sentences each iteration. When computing the uncertainty of a sentence s, we took the maximum uncertainty across all tokens in s.

4.4 Results on active over-labeling For each of the three tasks described above, we built four learning curves. The experiments compare four settings, for the four combinations of *active* vs. *passive* learning,[4] and standard labeling vs. over-labeling with HAL. Because standard learning solicits coarse-grained labels for the coarse-grained task, we refer to that setting as *coarse*. The coarse algorithms are representative of the state of the art in the document classification and sequence tagging. HAL uses finer-grained labels, so we refer to the over-labeling setting as *fine*.

The results appear in Figure 4.1. As all three figures show, active over-labeling with *active coarse* is the best method across all three data sets. These results show how HAL improves accuracy by querying for examples at a finer granularity than that targeted for classification.

4.5 Results as cost varies The above 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, which means we might not be able to afford to purchase purely fine-grained labels.

We first provide an analysis indicating that active over-labeling is likely to provide value at varying ratios of cost between coarse and fine-grained labels. We examine the learning curve of varying fixed ratio of instances labeled at the fine and coarse levels in the experiments (FFR). For example, in a setting of fine cost 16 and iteration budget of 32, FFR[0.5] allocates each of its 50% budget to coarse and fine, which corresponds to picking 16 coarse instances and 1 fine instance per iteration. This curve 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.

We switch from the fixed number of purchases per iteration to a fixed budget per iteration, re-run the three experiments from Section 4.1–Section 4.3, and then compare the AUC scores achieved for active FFR learners in across experiments.

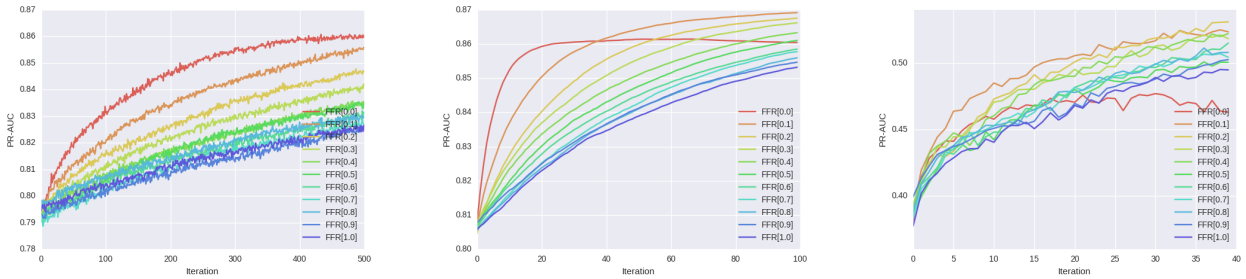
In Figure 4(a), the synthetic dataset experiment, lower FFR achieves better AUC in all 500 iterations. The synthetic problem is easy to learn which means the benefit of fine-grained labels can not compensate a fine cost is as high as 16. It is more cost effective to use a pure coarse-based classifier and not utilize fine-grained labels at all.

In Figure 4(b), the document classification experiment, FFR[0.0] rises fast but quickly reaches a bottleneck, which is surpassed by FFR[0.1] in a later iteration. This problem is less easy to learn and fine-grained labels may be worth their cost, depending on the overall budget. If the budget is limited to fewer than 38 rounds, FFR[0.0] is the best choice; otherwise, FFR[0.1] delivers better value than FFR[0.0].

In Figure 4(c), the sequence tagging experiment, the problem is harder, so FFR[0.1] has higher AUC than FFR[0.0] starting from the beginning. Then FFR[0.2] catches up after round 35. Higher FFR ratio is more affordable in the long run. If the budget is less than 35 rounds then FFR[0.1] is preferred over FFR[0.2].

From Figure 4, we can see that the choice of FFR to achieve high AUC depends on multiple factors, like the overall budget (the number rounds of iterations before termination), the cost of fine-grained labels and the nature of

[4] 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.



(a) Synthetic dataset (b) Document classification (c) Sequence tagging

Figure 4: Learning curves comparing active FFR method PR-AUC on fine cost 16

the problem target itself.

4.6 Results with BANDIT We now turn to evaluating BANDIT, which chooses purchase proportions dynamically. We configure the two strategies that BANDIT selects between to be the all-coarse (FFR[0.0]) and all-fine (FFR[1.0]) strategies. We again set aside 20000, 4000 and 10000 unlabeled examples for the synthetic task, document classification, and sequence tagging, and re-run the experiment in Section 4.5 with BANDIT. To evaluate the robustness of the algorithm, we test each approach for fine-grain cost varying within $[1.0, 1.1, 1.2, 1.5, 2.0, 4.0, 8.0, 16.0, 32.0, 64.0]$

Figure 5 shows the AUC for FFR and BANDIT in the end. We see that AUC in FFR[0.0] is not affected by the fine cost. And the AUCs for the other fixed ratio methods FFR[0.1]-FFR[1.0] decrease as fine grained labels become more expensive, because that results in these methods acquiring fewer labels. FFR with a lower fine-grained ratio achieves higher AUC when fine cost is high, and vice vesa. But the BANDIT curve is almost always among the top curves, regardless fine cost. This shows how BANDIT is robust to changes in cost.

Tables 1–3 quantifies the observations in Figure 5, by measuring how close each learner is to the top-scoring learner as fine cost varies. The metric *diff* gives the learner's absolute difference from the top learner. The *rank* metric is the learner's relative rank ordered by AUC. We caculate the minimum, maximum, mean and standard deviation for both metrics. In Table 1, the diff of BANDIT is in the range of [0.001, 0.004] and averages 0.002 away from the top curve. and the rank of BANDIT ranges from 1 to 5 and averages 2.625. BANDIT's mean for diff and rank is the lowest among all learners, and it has a low standard deviation, indicating that BANDIT scores close to the top curve most of the time. Similar results are shown in Table 2 and Table 3, where the mean diff of BANDIT is the lowest and the mean rank is the second lowest (as highlighted). These results illustrate tell that BANDIT successfully tunes the fine-grained ratio itself to cope with variable cost/benefit settings, which is important in real-world settings where the costs and benefits from querying at different levels of granualrity are not known in advance.

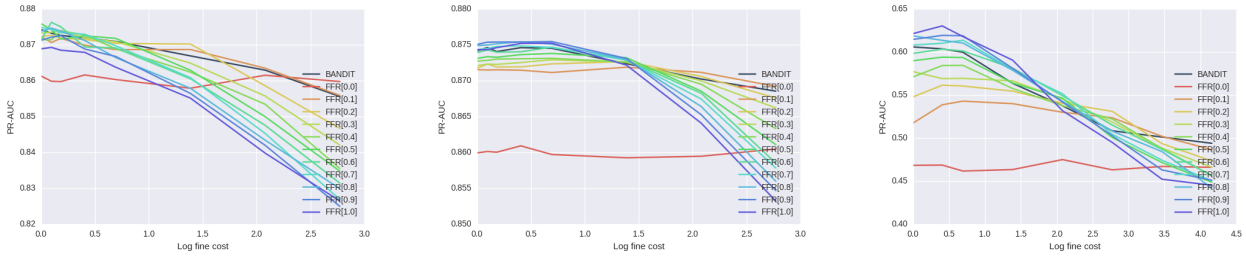
Table 1: Aggregated PR AUC for synthetic dataset

	diff				rank			
	min	max	mean	std	min	max	mean	std
algorithm								
BANDIT	0.001	0.004	**0.002**	0.001	1	5	**2.625**	1.598
FFR[0.0]	0.000	0.016	0.010	0.006	0	11	8.125	4.549
FFR[0.1]	0.000	0.006	0.003	0.002	0	9	4.875	3.603
FFR[0.2]	0.000	0.013	0.004	0.004	0	7	4.000	2.204
FFR[0.3]	0.001	0.018	0.005	0.006	1	4	3.250	1.165
FFR[0.4]	0.000	0.024	0.007	0.008	1	8	4.875	2.357
FFR[0.5]	0.000	0.025	0.006	0.009	0	9	3.750	3.151
FFR[0.6]	0.000	0.028	0.008	0.010	0	8	5.250	3.370
FFR[0.7]	0.000	0.033	0.008	0.012	0	9	4.250	3.240
FFR[0.8]	0.001	0.030	0.009	0.011	1	9	6.000	3.251
FFR[0.9]	0.002	0.035	0.011	0.012	6	11	8.750	1.669
FFR[1.0]	0.005	0.033	0.013	0.010	10	11	10.250	0.463

Table 2: Aggregated PR AUC for document classification

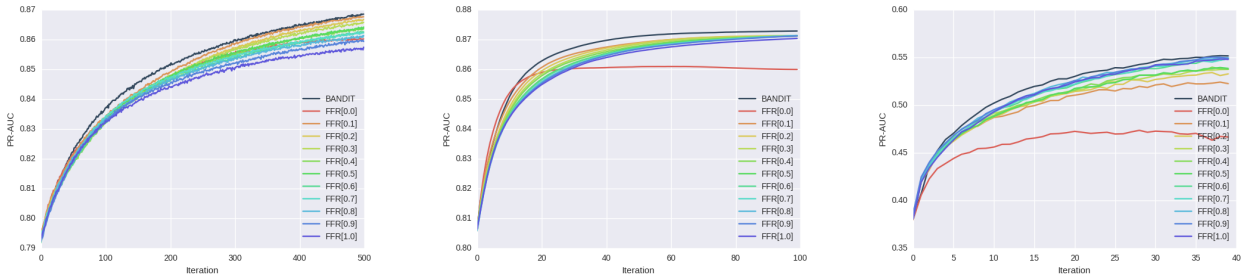
	diff				rank			
	min	max	mean	std	min	max	mean	std
algorithm								
BANDIT	0.001	0.001	**0.001**	0.000	1	8	3.750	2.188
FFR[0.0]	0.009	0.016	0.014	0.002	6	11	10.375	1.768
FFR[0.1]	0.000	0.004	0.003	0.002	0	10	7.500	4.629
FFR[0.2]	0.001	0.004	0.002	0.001	1	9	6.500	3.381
FFR[0.3]	0.001	0.003	0.002	0.001	3	9	6.750	2.375
FFR[0.4]	0.001	0.006	0.003	0.002	4	7	6.125	1.356
FFR[0.5]	0.000	0.008	0.003	0.002	0	6	5.000	2.070
FFR[0.6]	0.000	0.011	0.002	0.003	3	7	4.875	1.356
FFR[0.7]	0.000	0.011	0.002	0.004	2	8	4.250	2.121
FFR[0.8]	0.000	0.013	0.002	0.005	0	9	3.000	3.464
FFR[0.9]	0.000	0.015	0.003	0.005	0	10	**2.625**	4.274
FFR[1.0]	0.000	0.016	0.003	0.006	1	11	5.250	4.062

Finally, we examine how BANDIT performs as budget changes. Figure 6 shows the learning curves of how AUC increases for FFR and BANDIT, averaging over different costs. On different tasks, different fine-grained ratios may be preferable for different budgets—e.g., FFR[0.0] performs



(a) Synthetic dataset experiment (b) Document classification experiment (c) Sequence tagging experiment

Figure 5: Comparing active FFR and BANDIT method PR-AUC on different fine cost



(a) Synthetic dataset experiment (b) Document classification experiment (c) Sequence tagging experiment

Figure 6: Learning curves comparing active FFR and BANDIT method PR-AUC on mixed fine cost

Table 3: Aggregated PR AUC for sequence tagging

	diff				rank			
	min	max	mean	std	min	max	mean	std
algorithm								
BANDIT	0.000	0.027	**0.016**	0.011	0	8	4.250	2.712
FFR[0.0]	0.028	0.162	0.101	0.056	4	11	9.875	2.475
FFR[0.1]	0.000	0.104	0.045	0.041	0	10	6.500	4.840
FFR[0.2]	0.000	0.074	0.035	0.029	0	9	5.750	3.845
FFR[0.3]	0.006	0.061	0.030	0.020	3	8	5.000	2.330
FFR[0.4]	0.009	0.050	0.030	0.016	2	8	6.000	2.138
FFR[0.5]	0.005	0.045	0.029	0.011	2	9	6.500	2.268
FFR[0.6]	0.002	0.038	0.019	0.011	1	6	**3.875**	1.885
FFR[0.7]	0.000	0.044	0.018	0.014	0	8	4.125	2.850
FFR[0.8]	0.003	0.052	0.018	0.015	1	11	4.625	3.114
FFR[0.9]	0.000	0.043	0.018	0.016	0	10	4.375	3.662
FFR[1.0]	0.000	0.050	0.019	0.022	0	11	5.125	5.249

best for the first few purchases in Figure 6(b), but is much worse for larger budgets. However, in all figures, BANDIT almost always maintains the top AUC score after the first few rounds. Thus, we expect BANDIT to perform well against fixed ratio methods across a variety of budgets.

5 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 [20]. 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 [2, 3] or hierarchies [4]. 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 [21]. 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., Merialdo [22]); other measures include uncertainty in the parameters of probabilistic models [23] or the size of a model's decision boundary [24].

In previous work, active learning has also been shown to reduce sampling bias by utilizing the hierarchical structure of input features [25, 26]. By contrast, our work focuses on active learning over hierarchically structured output labels.

Luo et al. [27] 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.

6 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. Finally we proposed a more sophisticated algorithm which dynamically adjust the mix of different levels of labels, which benefit from more detailed cost analyses.

In the roadmp, it would be interesting to consider other hierarchically labeled data sets with multiple layers, e.g., that labeled by the Gene Ontology [1].

References

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

[2] M. Fleischman and E. Hovy, "Fine-grained classification of named entities," in *Proc. 19th Int. Conf. on Comp. Linguistics-Vol 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] P. Auer, N. Cesa-Bianchi, P. Fischer, "Finite-time analysis of the multi-armed bandit problem," *Machine Learning* 47, 235-256 (2002).

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

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

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

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

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

[20] 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.

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

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

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

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

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

[26] C. Symons et al., "Multi-Criterion Active Learning in Conditional Random Fields," *ICTAI*, pp. 323–331, 2006.

[27] W. Luo, A. Schwing, and R. Urtasun, "Latent structured active learning," in *NIPS*, 2013.