Hierarchical Sampling for Active Learning

Sanjoy Dasgupta DASGUPTA@CS.UCSD.EDU
Daniel Hsu DJHSU@CS.UCSD.EDU
Department of Computer Science and Engineering, University of California, San Diego
9500 Gilman Drive, La Jolla, CA 92093-0404

Abstract

We present an active learning scheme that exploits cluster structure in data.

1. Introduction

The *active learning* model is motivated by scenarios in which it is easy to amass vast quantities of unlabeled data (images and videos off the web, speech signals from microphone recordings, and so on) but costly to obtain their labels. It shares elements with both supervised and unsupervised learning. Like supervised learning, the goal is ultimately to learn a classifier. But like unsupervised learning, the data come unlabeled. More precisely, the labels are hidden, and each of them can be revealed only at a cost. The idea is to query the labels of just a few points that are especially informative about the decision boundary, and thereby to obtain an accurate classifier at significantly lower cost than regular supervised learning. Indeed, there are canonical examples in which active learning provably yields exponentially lower label complexity than supervised learning (Cohn et al., 1994; Freund et al., 1997; Dasgupta, 2005; Balcan et al., 2006; Balcan et al., 2007; Castro & Nowak, 2007; Hanneke, 2007; Dasgupta et al., 2007). However, these examples are highly specific, and the wider efficacy of active learning remains to be characterized.

Sampling bias. A typical active learning heuristic might start by querying a few randomly-chosen points, to get a very rough idea of the decision boundary. It might then query points that are increasingly closer to its current estimate of the boundary, with the hope of rapidly honing in. Such heuristics immediately bring to the forefront the unique difficulty of active learning, the fundamental characteristic that separates it from other learning models: *sampling bias*. As training proceeds, and points are queried based on increasingly confident assessments of their informativeness, the training set quickly diverges from the underlying data distribution. It consists of an unusual subset of points, hardly a representative subsample; why should a classifier trained on these strange points do well on the overall distribution? In section 2, we make this intuition concrete, and show how ill-managed sampling bias causes many active learning heuristics to not be consistent: even with infinitely many labels, they fail to converge to a good hypothesis.

The two faces of active learning. The recent literature offers two distinct narratives for explaining when active learning is helpful. The first has to do with *efficient search through the hypothesis space*. Each time a new label is seen, the set of plausible classifiers (those roughly consistent with the labels seen so far) shrinks somewhat. Using active learning, one can explicitly select points whose labels will shrink this set as fast as possible. Most theoretical work in active learning attempts to formalize this intuition.

The second argument for active learning has to do with *exploiting cluster structure in data*. Suppose, for instance, that the unlabeled points form five nice clusters; with luck, these clusters will be "pure" and only five labels will be necessary! Of course, this is hopelessly optimistic. In general, there may be no nice clusters, or there may be viable clusterings at many different resolutions. The clusters themselves may only be mostly-pure, or they may not be aligned with labels at all. In this paper, we present a scheme for cluster-based active learning that is statistically consistent and never has worse label complexity than supervised learning. In cases where there exists cluster structure (at whatever resolution) that is loosely aligned with class labels, the scheme detects and exploits it.

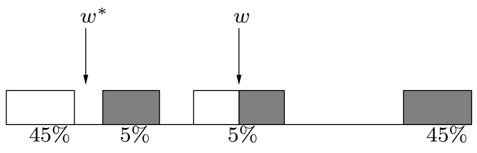
Our model. We start with a hierarchical clustering of the unlabeled points. This should be constructed so that some pruning of it is weakly informative of the

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

class labels. We describe an active learning strategy with good statistical properties, that will discover and exploit any informative pruning of the cluster tree. For instance, suppose it is possible to prune the cluster tree to m leaves (m unknown) that are fairly pure in the labels of their constituent points. Then, after querying just $O(m)$ labels, our learner will have a fairly accurate estimate of the labels of the *entire* data set. These can then be used as is, or as input to a supervised learner. Thus, our scheme can be used in conjunction with any hypothesis class, no matter how complex.

2. Active Learning and Sampling Bias

Many active learning heuristics start by choosing a few unlabeled points at random and querying their labels. They then repeatedly do something like this: fit a classifier $h \in H$ to the labels seen so far; and query the label of the unlabeled point closest to the decision boundary of h (or the one on which h is most uncertain, or something similar). Such schemes make intuitive sense, but do not correctly manage the bias introduced by adaptive sampling. Consider this 1-d example:



Here the data lie in four groups on the line, and are (say) distributed uniformly within each group. Filled blocks have a + label, while clear blocks have a − label. Most of the data lies in the two extremal groups, so an initial random sample has a good chance of coming entirely from these. Suppose the hypothesis class consists of thresholds on the line: $H = \{h_w : w \in \mathbb{R}\}$ where $h_w(x) = \mathbf{1}(x \geq w)$. Then the initial boundary will lie somewhere in the center group, and the first query point will lie in this group. So will every subsequent query point, forever. As active learning proceeds, the algorithm will gradually converge to the classifier shown as w. But this has 5% error, whereas classifier w^* has only 2.5% error. Thus the learner is not consistent: even with infinitely many labels, it returns a suboptimal classifier.

The problem is that the second group from the left gets overlooked. It is not part of the initial random sample, and later on, the learner is mistakenly confident that the entire group has a − label. And this is just in one dimension; in high dimension, the problem can be expected to be worse, since there are more places for this troublesome group to be hiding out. For a discussion of this problem in text classification, see the recent paper of Schutze et al. (2006).

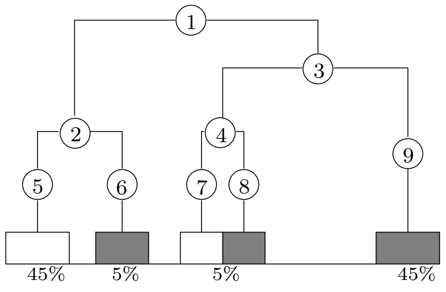


Figure 1. The top few nodes of a hierarchical clustering.

Sampling bias is the most fundamental challenge posed by active learning. This paper presents a broad framework for managing this bias that is provably sound.

3. A Clustering-Based Framework for Guiding Sampling

Our active learner starts with a hierarchical clustering of the data. Figure 1 shows how this might look for the example of the previous section.

Here only the top few nodes of the hierarchy are shown; their numbering is immaterial. At any given time, the learner works with a particular partition of the data set, given by a pruning of the tree. Initially, this is just $\{1\}$, a single cluster containing everything. Random points are drawn from this cluster and their labels are queried. Suppose one of these points, x, lies in the rightmost group. Then it is a random sample from node 1, but also from nodes 3 and 9. Based on these random samples, each node of the tree maintains statistics about the relative numbers of positive and negative instances seen. A few samples reveal that the top node 1 is very mixed while nodes 2 and 3 are substantially more pure. Once this transpires, the partition $\{1\}$ will be replaced by $\{2, 3\}$. Subsequent random samples will be chosen from either 2 or 3, according to a sampling strategy favoring the less-pure node. A few more queries down the line, the pruning will likely be refined to $\{2, 4, 9\}$. This is when the benefits of the partitioning scheme become most obvious; based on the samples seen, it can be concluded that cluster 9 is (almost) pure, and thus (almost) no more queries will be made from it until the rest of the space has been partitioned into regions that are similarly pure.

The querying can be stopped at any stage; then, each cluster in the current partition gets assigned the majority label of the points queried from it. In this way, the *entire* data set gets labeled, and the number of erroneous labels induced is kept to a minimum. If desired, these labels can be used for a subsequent round of supervised learning, with any learning algorithm and any hypothesis class.

3.1. Preliminary Definitions

The cost of a pruning. Say there are n unlabeled points, and we have a hierarchical clustering represented by a binary tree T with n leaves. For any node v of the tree, denote by T_v both the subtree rooted at v and also the data points contained in this subtree (at its leaves). A *pruning* of the tree is a subset of nodes $\{v_1, \ldots, v_m\}$ such that the T_{v_i} are disjoint and together cover all the data. At any given stage, the active learner will work with a partition of the data set given by a pruning of T. In the analysis, we will also deal with a *partial pruning*: a subset of a pruning.

A weight of a node $v \in T$ is the proportion of the data set in T_v: $w_v = (\text{number of leaves of } T_v)/n$. Likewise, the weight of a partial pruning is the fraction of the data set that it covers, $w(P) = \sum_{v \in P} w_v$. A full pruning has weight 1.

Suppose there are k possible labels, and that their proportions in T_v are $p_{v,l}$ for $l = 1, \ldots, k$. Then the error introduced by assigning all points in T_v their majority label is $\epsilon_v = 1 - \max_l p_{v,l}$. Consequently, the error induced by a particular pruning (or partial pruning) P—that is, the fraction of incorrect labels when each cluster of P is assigned its majority label—is

$$\epsilon(P) = \frac{1}{w(P)} \sum_{v \in P} w_v \epsilon_v$$

In pruning the tree, it always helps to go as far down as possible, *provided* we can accurately estimate the majority labels in those nodes.

Empirical estimates for individual nodes. Due of limited sampling, we will only have labels from some of the nodes, and even for those, we may not be able to correctly determine the majority label. If we assign label l to all the points in T_v, the induced error is $\epsilon_{v,l} = 1 - p_{v,l}$. Likewise, when each cluster v of pruning (or partial pruning) P is assigned label $L(v) \in \{1, 2, \ldots, k\}$, the error induced is

$$\epsilon(P, L) = \frac{1}{w(P)} \sum_{v \in P} w_v \epsilon_{v,L(v)}.$$

We will at any given time have only very imperfect estimates of the $p_{v,l}$'s and thus of these various error probabilities. Fix any node v, and suppose that at time t, we have queried $n_v(t)$ random points contained in that node. This gives us estimates of its class probabilities, $p_{v,l}(t)$. Correspondingly, our estimate of $\epsilon_{v,l}$ will be $\epsilon_{v,l}(t) = 1 - p_{v,l}(t)$.

Table 1. Key quantities in the algorithm and analysis. The indexing (t) specifies the empirical quantity at time t.

d_v	depth of node v in tree
d_P	maximum depth of nodes in P
w_v	weight of node v
$p_{v,l}$	fraction of label l in node v
$L^*(v)$	majority label of node v (that is, $\arg\max_l p_{v,l}$)
$n_v(t)$	number of points sampled from node v
$p_{v,l}(t)$	fraction of label l in points sampled from T_v
$\mathcal{A}(t)$	admissible (node,label) pairs
$\epsilon_{v,l}(t)$	$1 - p_{v,l}(t)$
$\widetilde{\epsilon}_{v,l}(t)$	$\epsilon_{v,l}(t)$ if $(v,l) \in \mathcal{A}(t)$; otherwise 1

The quality of these estimates can be assessed using generalization bounds. At any given time t, we can associate with each node v and label l a confidence interval $[p^{\text{LB}}_{v,l}, p^{\text{UB}}_{v,l}]$ within which we expect the true probability $p_{v,l}$ to lie. One possibility is to use $[\max(p_{v,l}(t) - \Delta_{v,l}(t), 0), \min(p_{v,l}(t) + \Delta_{v,l}, 1)]$, for $\Delta_{v,l}(t) \approx \frac{1}{n_v(t)} + \sqrt{\frac{p_{v,l}(t)(1-p_{v,l}(t))}{n_v(t)}}$. In Lemma 1, we will give a precise value for $\Delta_{v,l}(t)$ for which we are able to assert that (with high probability) *every* $p_{v,l}$ is *always* within this interval. However, there are other ways of constructing confidence intervals as well. The most accurate is simply to use the binomial (or hypergeometric) distribution directly.

When are we confident about the majority label of a subtree? As mentioned above, it is advantageous to descend as far as possible in the tree, provided we are confident about our estimate of the majority label. To this end, define

$$A_{v,l}(t) = \text{true} \quad \Leftrightarrow \quad (1 - p^{\text{LB}}_{v,l}(t)) < \beta \cdot \min_{l' \neq l} (1 - p^{\text{UB}}_{v,l'}(t)). \tag{1}$$

$A_{v,l}$ asserts that l is an *admissible* label for node v, in the weak sense that it incurs at most β times as much error as any other label. To see this, notice that label l gets at most $1 - p^{\text{LB}}_{v,l}(t)$ fraction of the points wrong, whereas l' gets at least $1 - p^{\text{UB}}_{v,l'}(t)$ fraction of the points wrong. In our experiments, we use $\beta = 2$, in which case

$$A_{v,l}(t) = \text{true} \quad \Leftrightarrow \quad p^{\text{LB}}_{v,l}(t) > 2p^{\text{UB}}_{v,l'}(t) - 1 \ \forall l' \neq l.$$

For any given v, t, several different labels l might satisfy this criterion, for instance if $p^{\text{LB}}_{v,l}(t) = p^{\text{UB}}_{v,l}(t) = 1/k$ for all labels l. When there are only two possible labels, the criterion further simplifies to $p^{\text{LB}}_{v,l}(t) > 1/3$.

We will maintain a set of (v, l) pairs for which the condition $A_{v,l}(t)$ is either true or was true sometime in the past:

$$\mathcal{A}(t) = \{(v, l) : A_{v,l}(t') \text{ for some } t' \leq t\}.$$

$\mathcal{A}(t)$ is the set of admissible (v, l) pairs at time t. We use it to stop ourselves from descending too far down tree T when only a few samples have been drawn. Specifically, we say pruning P and labeling L are *admissible* in tree T at time t if:

- $L(v)$ is defined for P and ancestors of P in T.
- $(v, L(v)) \in \mathcal{A}(t)$ for any node v that is a strict ancestor of P in T.
- For any node $v \in P$, there are two options:
  - either $(v, L(v)) \in \mathcal{A}(t)$;
  - or there is no l for which $(v, l) \in \mathcal{A}(t)$. In this case, if v has parent u, then $(u, L(v)) \in \mathcal{A}(t)$.

This final condition implies that if a node in P is not admissible (with any label), then it is forced to take on an admissible label of its parent.

Empirical estimate of the error of a pruning. For any node v, the empirical estimate of the error induced when all of subtree T_v is labeled l is $\epsilon_{v,l}(t) = 1 - p_{v,l}(t)$. This extends to a pruning (or partial pruning) P and a labeling L:

$$\epsilon(P, L, t) = \frac{1}{w(P)} \sum_{v \in P} w_v \epsilon_{v,L(v)}(t).$$

This can be a bad estimate when some of the nodes in P have been inadequately sampled. Thus we use a more conservative adjusted estimate:

$$\widetilde{\epsilon}_{v,l}(t) = \begin{cases} 1 - p_{v,l}(t) & \text{if } (v, l) \in \mathcal{A}(t) \\ 1 & \text{if } (v, l) \notin \mathcal{A}(t) \end{cases}$$

with $\widetilde{\epsilon}(P, L, t) = (1/w(P)) \sum_{v \in P} w_v \widetilde{\epsilon}_{v,L(v)}(t)$. The various definitions are summarized in Table 1.

Picking a good pruning. It will be convenient to talk about prunings not just of the entire tree T but also of subtrees T_v. To this end, define the *score of v at time t*—denoted $s(v, t)$—to be the adjusted empirical error of the best admissible pruning and labeling (P, L) of T_v. More precisely, $s(v, t)$ is

$$\min\{\widetilde{\epsilon}(P, L, t) : (P, L) \text{ admissible in } T_v \text{ at time } t\}.$$

Written recursively, $s(v, t)$ is the minimum of

- $\widetilde{\epsilon}_{v,l}(t)$, for all l;
- $\frac{w_a}{w_v} s(a, t) + \frac{w_b}{w_v} s(b, t)$, whenever v has children a, b and $(v, l) \in \mathcal{A}(t)$ for some l.

Algorithm 1 Cluster-adaptive active learning

```
Input: Hierarchical clustering of $n$ unlabeled points; batch size $B$
$P \leftarrow \{\text{root}\}$ (current pruning of tree)
$L(\text{root}) \leftarrow 1$ (arbitrary starting label for root)
for time $t = 1, 2, \ldots$ until the budget runs out do
  for $i = 1$ to $B$ do
    $v \leftarrow \text{select}(P)$
    Pick a random point $z$ from subtree $T_v$
    Query $z$'s label $l$
    Update empirical counts and probabilities $(n_u(t), p_{u,l}(t))$ for all nodes $u$ on path from $z$ to $v$
  end for
  In a bottom-up pass of $T$, update $\mathcal{A}$ and compute scores $s(u, t)$ for all nodes $u \in T$ (see text)
  for each (selected) $v \in P$ do
    Let $(P', L')$ be the pruning and labeling of $T_v$ achieving scores $s(v, t)$
    $P \leftarrow (P \setminus \{v\}) \cup P'$
    $L(v) \leftarrow L'(u)$ for all $u \in P'$
  end for
end for
for each cluster $v \in P$ do
  Assign each point in $T_v$ the label $L(v)$
end for
```

Starting from the empirical estimates $p_{v,l}(t), p^{\text{LB}}_{v,l}, p^{\text{UB}}_{v,l}$, it is possible to update the set $\mathcal{A}(t)$ and to compute all the $\widetilde{\epsilon}_{v,l}(t)$ and $s(v, t)$ values in a single linear-time, bottom-up pass through the tree.

3.2. The Algorithm

Algorithm 1 contains the active learning strategy. It remains to specify the the manner in which the hierarchical clustering is built and the procedure **select**. *Regardless of how these decisions are made*, the algorithm is statistically sound in that the confidence intervals $p_{v,l} \pm \Delta_{v,l}(t)$ are valid, and these in turn validate the guarantees for admissible prunings/labelings. This leaves a lot of flexibility to explore different clustering and sampling strategies.

The select procedure. This controls the selective sampling. Some options:

(1) Choose $v \in P$ with probability $\propto w_v$. This is similar to random sampling.

(2) Choose v with probability $\propto w_v(1 - p^{\text{UB}}_{v,L(v)}(t))$. This is an active learning rule that reduces sampling

in regions of the space that have already been observed to be fairly pure in their labels.

(3) For each subtree ($T_z, z \in P$), find the observed majority label, and assign this label to all points in the subtree; fit a classifier h to this data; and choose $v \in P$ with probability $\propto \min\{|\{x \in T_v : h(x) = +1\}|, |\{x \in T_v : h(x) = -1\}|\}$. This biases sampling towards regions close to the current decision boundary.

Building a hierarchical clustering. The scheme works best when there is a pruning P of the tree such that $|P|$ is small and a significant fraction of its constituent clusters are almost-pure. One option is to run a standard hierarchical clustering algorithm, like average linkage, perhaps with a domain-specific distance function (or one generated from a neighborhood graph). Another option is to use a bit of labeled data to guide the construction of the hierarchy.

3.3. Naive Sampling

First consider the naive sampling strategy in which a node $v \in P$ is selected in proportion to its weight w_v. We'll show that if there is an almost-pure pruning with m nodes, then only $O(m)$ labels are needed before the entire data is labeled almost-perfectly. Proofs are deferred to the full version of the paper.

Theorem 1 *Pick any $\delta, \eta > 0$ and any pruning Q with $\epsilon(Q) \leq \eta$. With probability at least $1 - \delta$, the learner induces a labeling (of the data set) with error $\leq (\beta + 1)\epsilon(Q) + \eta$ when the number of labels seen is*

$$Bt = O\left(\frac{\beta+1}{\beta-1} \cdot \frac{|Q|}{\eta} \log \frac{2^{d_Q} kB|Q|}{\eta\delta}\right).$$

Recall that β is used in the definition of an admissible label (equation (1)); we use $\beta = 2$ in our experiments.

The number of prunings with m nodes is about 4^m; and these correspond to roughly $(4k)^m$ possible classifications (each of the m clusters can take on one of k labels). Thus this result is what one would expect if one of these classifiers were chosen by supervised learning. In our scheme, we do not evaluate such classifiers directly, but instead evaluate the subregions of which they are composed. We start our analysis with confidence intervals for $p_{v,l}$ and n_v.

Lemma 1 *Pick any $\delta > 0$. With probability at least $1 - \delta$, the following holds for all nodes $v \in T$, all labels l, and all times t.*

(a) $|p_{v,l} - p_{v,l}(t)| \leq \Delta_{v,l} \leq \Delta_{v,l}(t)$, where

$$\Delta_{v,l} = \frac{2}{3n_v(t)} \log \frac{1}{\delta'} + \sqrt{\frac{2p_{v,l}(1-p_{v,l})}{n_v(t)} \log \frac{1}{\delta'}}.$$

$$\Delta_{v,l}(t) = \frac{5}{n_v(t)} \log \frac{1}{\delta'} + \sqrt{\frac{9p_{v,l}(t)(1-p_{v,l}(t))}{2n_v(t)} \log \frac{1}{\delta'}}.$$

for $\delta' = \delta/(kBt^2 d_v^2)$.

(b) $n_v(t) \geq Btw_v/2$ if $Btw_v \geq 8 \log(t^2 2^{2d_v}/\delta)$.

Our empirical assessment of the quality of a pruning P is a blend of sampling estimates $p_{v,l}(t)$ and perfectly known values w_v. Next, we examine the rate of convergence of $\epsilon(P, L, t)$ to the true value $\epsilon(P, L)$.

Lemma 2 *Assume the bounds of Lemma 1 hold. There is a constant c such that for all prunings (or partial prunings) $P \subset T$, all labelings L, and all t,*

$$w(P) \cdot |\epsilon(P, L, t) - \epsilon(P, L)| \ \leq c \cdot \left(\frac{|P|}{Bt} \log \frac{kBt^2 2^{d_P}}{\delta} + \sqrt{w(P)\epsilon(P,L)\frac{|P|}{Bt} \log \frac{kBt^2 2^{d_P}}{\delta}}\right).$$

Lemma 2 gives useful bounds on $\epsilon(P, L, t)$. Our algorithm uses the more conservative estimate $\widetilde{\epsilon}(P, L, t)$, which is identical to $\epsilon(P, L, t)$ except that it automatically assigns an error of 1 to any $(v, L(v)) \notin \mathcal{A}(t)$, that is to say, any (node, label) pair for which insufficiently many samples have been seen. We need to argue that for nodes v of reasonable weight, and their majority labels $L^*(v)$, we will have $(v, L^*(v)) \in \mathcal{A}(t)$.

Lemma 3 *There is a constant c' such that $(v, l) \in \mathcal{A}(t)$ for any node v with majority label l and*

$$w_v \ \geq \ \max\left(\frac{8}{Bt} \log \frac{t^2 2^{2d_v}}{\delta}, \ \frac{\beta+1}{\beta-1} \cdot \frac{c'}{Bt} \log \frac{kBt^2 d_v^2}{\delta}\right).$$

The purpose of the set $\mathcal{A}(t)$ is to stop the algorithm from descending too far in the tree. We now quantify this. Suppose there is a good pruning that contains a node q whose majority label is $L^*(q)$. However, our algorithm descends far below q, to some pruning P (and associated labeling L) of T_q. By the definition of admissible pruning, this can only happen if $(q, L(q))$ lies in $\mathcal{A}(t)$. Under such circumstances, it can be proved that (P, L) is not too much worse than $(q, L^*(q))$.

Lemma 4 *For any node q, let (P, L) be the admissible pruning and labeling of T_q found by our algorithm at time t. If $(q, L(q)) \in \mathcal{A}(t)$, then $\epsilon(P, L) \leq (\beta + 1)\epsilon_q$.*

Proof sketch of Theorem 1. Let Q, t be as in the theorem statement, and let L^* denote the optimal labeling (by majority label) of each node. Define V to be the set of all nodes v with weight exceeding the bound in Lemma 3. As a result, $(v, L^*(v)) \in \mathcal{A}(t)$ for all $v \in V$.

Suppose that at time t, the learning scheme is using some pruning P with labeling L. We will decompose P and Q into three groups of nodes each: (i) $P_a \subset P$ are strict ancestors of $Q_a \subset Q$; (ii) $P_d \subset P$ are strict descendants of $Q_d \subset Q$; and (iii) the remaining nodes are common to P and Q.

Since nodes of P_a were never expanded to Q_a, we can show $w(P_a)\epsilon(P_a, L) \leq w(P_a)\epsilon(Q_a, L^*) + 2\eta/3 + w(Q_a \setminus V)$. Meanwhile, from Lemma 4 we have $w(P_d)\epsilon(P_d, L) \leq (\beta+1)w(Q_d)\epsilon(Q_d, L^*) + w(Q_d \setminus V)$. Putting it all together, we get $\epsilon(P, L) - \epsilon(Q, L^*) \leq \eta + (\beta+1)\epsilon(Q)$, under the conditions on t. ▮

3.4. Active Sampling

Suppose our current pruning and labeling are (P, L). So far we have only discussed the naive strategy of choosing query nodes $u \in P$ with probability proportional to w_u. For active learning, a more intelligent and adaptive strategy is needed. A natural choice is to pick u with probability proportional to $w_u \epsilon^{\mathrm{UB}}_{u,L(u)}(t)$, where $\epsilon^{\mathrm{UB}}_{u,l} = 1 - p^{\mathrm{LB}}_{u,l}(t)$ is an upper bound on the error associated with node u. This takes advantage of large, pure clusters: as soon as their purity becomes evident, querying is directed elsewhere.

Fallback analysis. Can the adaptive strategy perform worse than naive random sampling? There is one problematic case. Suppose there are only two labels, and that the current pruning P consists of two nodes (clusters), each with 50% probability mass; however, cluster A has impurity (minority label probability) 5% while B has impurity 50%. Under our adaptive strategy, we will query 10 times more from B than from A. But suppose B cannot be improved: any attempts to further refine it lead to subclusters which are also 50% impure. Meanwhile, it might be possible to get the error in A down to zero by splitting it further. In this case, random sampling, which weighs A equally to B, does better than the active learning scheme.

Such cases can only occur if the best pruning has high impurity, and thus active learning still yields a pruning that is not much worse than optimal. To see this, pick any good pruning Q (with optimal labeling L^*), and let's see how adaptive sampling fares with respect to Q. Suppose our scheme is currently working with a pruning P and labeling L. Divide P into two regions: $P_0 = \{p \in P : p \in T_v \text{ for some } v \in Q\}$ and $P_1 = P \setminus P_0$. The danger is that we will sample too much from P_0, where no further improvement is needed (relative to Q), and not enough from P_1. But it can be shown that *either* the active strategy samples from P_1 at least half as often as the random strategy would, *or* the current pruning is already pretty good, in that

$$\epsilon(P, L) \;\leq\; 2\epsilon(Q, L^*) + \text{terms involving sampling error.}$$

Benefits of active learning. Active sampling is sure to help when the hierarchical clustering has some large, fairly-pure clusters near the top of the tree. These will be quickly identified, and very few queries will subsequently be made in those regions. Consider an idealized example in which there are only two possible labels and each node in the tree is either pure or $(1/3, 2/3)$-impure. Specifically: (i) each node has two children, with equal probability mass; and (ii) each impure node has a pure child and an impure child. In this case, active sampling can be seen to yield a convergence rate $1/n^2$ in contrast to the $1/n$ rate of random sampling.

The example is set up so that the selected pruning P (with labeling L) always consists of pure nodes $\{a_1, a_2, \ldots, a_d\}$ (at depths $1, 2, \ldots, d$) and a single impure node b (at depth d). These nodes have weights $w_{a_i} = 2^{-i}$, $i = 1, \ldots, d$, and $w_b = 2^{-d}$; the impure node causes the error of the best pruning to be $\varepsilon = 2^{-d}/3$. The goal, then, is to sample enough from node b to cut this error in half (say, because the target error is $\varepsilon/2$). This can be achieved with a constant number of queries from node b, since this is enough to render the majority label of its pure child admissible and thus offer a superior pruning.

If we were to completely ignore the pure nodes, then the next several queries could all be made in node b; we thus halve the error with only a constant number of queries. Continuing this way leads to an exponential improvement in convergence rate. Such a policy of neglect is fine in our present example, but this would be imprudent in general: after all, the nodes we ignore may actually turn out impure, and only further sampling would reveal them as such. We instead select a node u with probability proportional to $w_u \epsilon^{\mathrm{UB}}_{u,L(u)}(t)$, and thus still select a pure node a_i with probability roughly proportional to $w_{a_i}/n_{a_i}(t)$. This allows for some cautionary exploration while still affording an improved convergence rate.

The chance of selecting the impure node b is

$$\frac{w_b \epsilon^{\mathrm{UB}}_{b,L(b)}}{w_b \epsilon^{\mathrm{UB}}_{b,L(b)} + \sum_{i=1}^d w_{a_i} \epsilon^{\mathrm{UB}}_{a_i,L(a_i)}} \geq \Omega\left(\frac{\varepsilon}{\varepsilon + \frac{d}{\sum_{i=1}^d n_{a_i}(t)}}\right).$$

The inequality follows (with high probability) because the error bound for b is always at least the true error ε (up to constants), while another argument shows that

$$\sum_{i=1}^{d} w_{a_i} \epsilon^{\mathrm{UB}}_{a_i,L(a_i)} = O\left(\sum_{i=1}^{d} \frac{w_{a_i}}{n_{a_i}(t)}\right) = O\left(\frac{d}{\sum_{i=1}^{d} n_{a_i}(t)}\right).$$

We need to argue that the pure nodes do not get queried too much. Well, if they have been queried at least $\sqrt{d/\varepsilon} = O(\sqrt{(1/\varepsilon)\log 1/\varepsilon})$ times, the chance of selecting b is $\Omega(\sqrt{\varepsilon/d})$; another $O(\sqrt{d/\varepsilon})$ queries with active sampling suffice to land a constant number in node b—just enough to cut the error in half. Overall, the number of queries needed is then $O(\sqrt{(1/\varepsilon)\log(1/\varepsilon)})$, considerably less than the $O(1/\varepsilon)$ required of random sampling.

4. Experiments

How many label queries can we save by exploiting cluster structure with active learning? Our analysis suggests that the savings is tied to *how well the cluster structure aligns with the actual labels*. To evaluate how accommodating real world data is in this sense, we studied the performance of our active learner on several natural classification tasks.

4.1. Classification Tasks

When used for classification, our active learning framework decomposes into three parts: (1) unsupervised hierarchical clustering of the unlabeled data, (2) cluster-adaptive sampling (Algorithm 1, with the second variant of **select**), and (3) supervised learning on the resulting fully labeled data. We used standard statistical procedures, Ward's average linkage clustering and logistic regression, for the unsupervised and supervised components, respectively, in order to assess just the role of the cluster-adaptive sampling method.

We compared the performance of our active learner to two baseline active learning methods, random sampling and margin-based sampling, that only train a classifier on the subset of queried labeled data. Random sampling chooses points to label at random, and margin-based sampling chooses to label the points closest to the decision boundary of the current classifier (as described in Section 2). Again, we used logistic regression with both of these methods.

A few details: We ran each active learning method 10 times for each classification task, allowing the budget of labels to grow in small increments. For each budget size, we evaluated the resulting classifier on a test set, computed its misclassification error, and averaged this error over the repeated trials. Finally, we used ℓ_2-regularization with logistic regression, choosing the trade-off parameter with 10-fold cross validation.

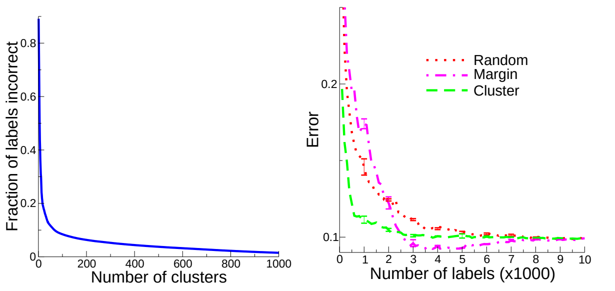


Figure 2. Results on OCR digits. Left: Errors of the best prunings in the OCR digits tree. Right: Test error curves on classification task.

OCR digit images. We first considered multi-class classification of the MNIST handwritten digit images.[1] We used 10000 training images and 2000 test images.

The tree produced by Ward's hierarchical clustering method was especially accommodating for cluster-adaptive sampling. Figure 2 (left) depicts this quantitatively; it shows the error of the best k-pruning of the tree for several values of k. For example, the tree had a pruning of 50 nodes with about 12% error. Our active learner found such a pruning using just 400 labels.

Figure 2 (right) plots the test errors of the three active learning methods on the multi-class classification task. Margin-based sampling and cluster-adaptive sampling both outperformed random sampling, with margin-based sampling taking over a little after 2000 label queries. The initial advantage of cluster-adaptive sampling reflects its ability to discover and subsequently ignore relatively pure clusters at the onset of sampling. Later on, it is left sampling from clusters of easily confused digits (e.g. 3's, 5's, and 8's).

The test error of the margin-based method appeared to actually dip below the test error of classifier trained using all of the training data (with the correct labels). This appears to be a case of fortunate sampling bias. In contrast, cluster-adaptive sampling avoids this issue by concentrating on converging to the same result as if it had all of the correct training labels.

Newsgroup text. We also considered four pairwise binary classification tasks with the 20 Newsgroups data set. Following Schohn and Cohn (2000), we chose four pairs of newsgroups that varied in difficulty. We used a version of the data set that removes duplicates and some newsgroup-identifying headers, but other-

[1] http://yann.lecun.com/exdb/mnist/

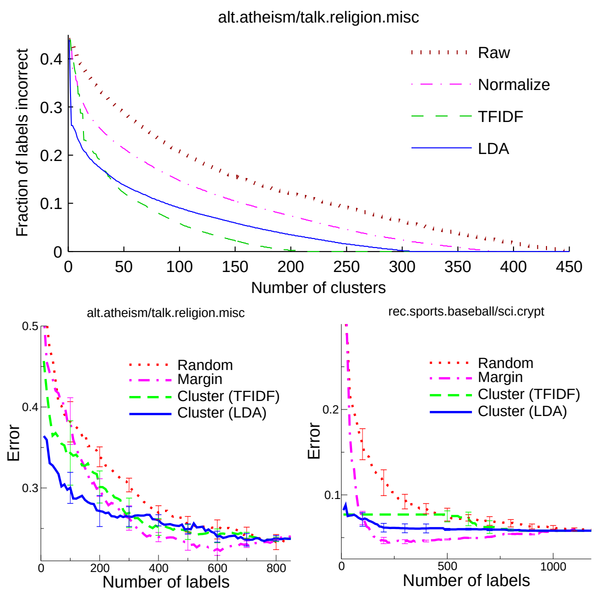


Figure 3. Results on newsgroup text. Top: Errors of the best prunings in various trees for atheism/religion pair. Bottom: Test error curves on newsgroup tasks.

wise represents each document as a simple word count vector.[2] Each newsgroup had about 1000 documents, and the data for each pair were partitioned into training and test sets at a 2:1 ratio. We length-normalized the count vectors before training the logistic regression models in order to speed up the training and improve classification performance.

The initial word count representation of the newsgroup documents yielded poor quality clusterings, so we tried various techniques for preprocessing text data before clustering with Ward's method: (1) normalize each document vector to unit length; (2) apply TF/IDF and length normalization to each document vector; and (3) infer a posterior topic mixture for each document using a Latent Dirichlet Allocation model trained on the same data (Blei et al., 2003). For the last technique, we used Kullback-Leibler divergence as the notion of distance between the topic mixture representations. Figure 3 (top) plots the errors of the best prunings. Indeed, the various changes-of-representation and specialized notions of distance help build clusterings of greater utility for cluster-adaptive active learning.

In all four pairwise tasks, both margin-based sampling and cluster-adaptive sampling outperformed random sampling. Figure 3 (bottom) shows the test errors on two of these newsgroup pairs. We observed the same effects regarding cluster-adaptive sampling and margin-based sampling as in the OCR digits data.

[2]http://people.csail.mit.edu/jrennie/20Newsgroups/

4.2. Rare Category Detection

To demonstrate its versatility, we applied our cluster-adaptive sampling method to a rare category detection task. We used the Statlog Shuttle data, a set of 43500 examples from seven different classes; the smallest class comprises a mere 0.014% of the whole. To discover at least one example from each class, random sampling needed over 8000 queries (averaged over several trials). In contrast, cluster-adaptive sampling needed just 880 queries; it sensibly avoided sampling much from clusters confidently identified as pure, and instead focused on clusters with more potential.

Acknowledgements. Support provided by the Engineering Institute at Los Alamos National Laboratory and the NSF under grants IIS-0347646 and IIS-0713540.

References

Balcan, M.-F., Beygelzimer, A., & Langford, J. (2006). Agnostic active learning. *ICML*.

Balcan, M.-F., Broder, A., & Zhang, T. (2007). Margin based active learning. *COLT*.

Blei, D., Ng, A., & Jordan, M. (2003). Latent dirichlet allocation. *JMLR*, *3*, 993–1022.

Castro, R., & Nowak, R. (2007). Minimax bounds for active learning. *COLT*.

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, *15*, 201–221.

Dasgupta, S. (2005). Coarse sample complexity bounds for active learning. *NIPS*.

Dasgupta, S., Hsu, D., & Monteleoni, C. (2007). A general agnostic active learning algorithm. *Neural Information Processing Systems*.

Freund, Y., Seung, H., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, *28*, 133–168.

Hanneke, S. (2007). A bound on the label complexity of agnostic active learning. *ICML*.

Schohn, G., & Cohn, D. (2000). Less is more: active learning with support vector machines. *ICML*.

Schutze, H., Velipasaoglu, E., & Pedersen, J. (2006). Performance thresholding in practical text classification. *ACM International Conference on Information and Knowledge Management*.