Analysis of a greedy active learning strategy

Sanjoy Dasgupta[*]
University of California, San Diego
dasgupta@cs.ucsd.edu

January 3, 2005

Abstract

We abstract out the core search problem of active learning schemes, to better understand the extent to which adaptive labeling can improve sample complexity. We give various upper and lower bounds on the number of labels which need to be queried, and we prove that a popular greedy active learning rule is approximately as good as any other strategy for minimizing this number of labels.

1 Introduction

An increasingly common phenomenon in classification tasks is that unlabeled data is abundant, whereas labels are considerably harder to come by. Genome sequencing projects, for instance, are producing vast numbers of peptide sequences, but reliably labeling even one of these with structural information requires time and close attention.

This distinction between labeled and unlabeled data is not captured in standard models like the PAC framework, and has motivated the field of *active learning*, in which the learner is able to ask for the labels of specific points, but is charged for each label. These query points are typically chosen from an unlabeled data set, a practice called *pool-based learning* [10]. There has also been some work on creating query points synthetically, including a rich body of theoretical results [1, 2], but this approach suffers from two problems: first, from a practical viewpoint, the queries thus produced can be quite unnatural and therefore bewildering for a human to classify [3]; second, since these queries are not picked from the underlying data distribution, they might have limited value in terms of generalization. In this paper, we focus on pool-based learning.

We are interested in active learning with generalization guarantees. Suppose the hypothesis class has VC dimension d and we want a classifier whose error rate on distribution P over the joint (input, label) space, is less than $\epsilon > 0$. The theory tells us that in a *supervised* setting, we need some $m = m(\epsilon, d)$ *labeled* points drawn from P (for a fixed level of confidence, which we will henceforth ignore). Can we get away with substantially fewer than m labels if we are given unlabeled points from P and are able to adaptively choose which points to label? How much fewer, and what querying strategies should we follow?

[*] This work was funded by the NSF, under grant IIS-0347646. A summary of the results appeared in *Advances in Neural Information Processing Systems*, 2004.

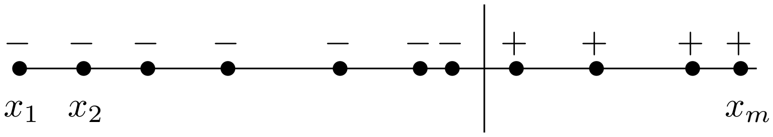

Figure 1: Finding the boundary between − and + requires just $O(\log m)$ labels, using binary search.

Here is a toy example illustrating the potential of active learning. Suppose the data lie on the real line, and the classifiers are simple thresholding functions, $H = \{h_w : w \in \mathbf{R}\}$:

$$h_w(x) = \begin{cases} 1 & \text{if } x \geq w \\ 0 & \text{if } x < w \end{cases}$$

VC theory tells us that if the underlying distribution P can be classified perfectly by some hypothesis in H (called the *realizable* case), then it is enough to draw $m = O(1/\epsilon)$ random labeled examples from P, and to return any classifier consistent with them. But suppose we instead draw m *unlabeled* samples from P. If we lay these points down on the line, their hidden labels are a sequence of 0's followed by a sequence of 1's, and the goal is to discover the point w at which the transition occurs (Figure 1). This can be accomplished with a simple binary search which asks for just $\log m$ labels. Thus active learning gives us an *exponential* improvement in the number of labels needed: by adaptively querying $\log m$ labels, we can automatically infer the rest of them.

The various active learning schemes in the literature differ significantly in the kinds of generalization guarantees which accompany their final hypotheses. It is important to distinguish between these. For instance, the hypothesis h produced by the above (binary search) active learning procedure has the following property: "in the realizable case, this h is exactly the hypothesis we would have returned if we knew all m labels", which implies that "if some $h^* \in H$ perfectly classifies P, then (with high probability) this particular h's performance on P is ϵ-close to that of h^*".

Generalized binary search?

So far we have only looked at an extremely simple learning problem. It is a tantalizing possibility that even for more complicated hypothesis classes H, a sort of a generalized binary search is possible. What would the search space look like? For supervised learning, in the realizable case, the usual bounds specify a sample complexity of (very roughly) $m \approx d/\epsilon$ labeled points if the target error rate is ϵ. So let's pick this many unlabeled points, and then try to find a hypothesis consistent with all the hidden labels by adaptively querying just a few of them. We know via Sauer's lemma that H can classify these m points (considered jointly) in at most $O(m^d)$ different ways – in effect, the size of H is reduced to $O(m^d)$. This finite set is the *effective hypothesis class* $\widehat{H}$. (For instance, in Figure 1, $\widehat{H}$ has size $m+1$, and corresponds to the intervals into which the points x_i split the real line.) The most we can possibly learn about the target hypothesis, even if all the labels are revealed, is to narrow it down to one of these regions. Is it possible to pick among these $O(m^d)$ possibilities using $o(m)$ labels? If binary search were possible, just $O(d \log m)$ labels would be needed.

Unfortunately, we cannot hope for a generic positive result of this kind. The toy example above is a one-dimensional linear separator ($d = 1$). We show that for $d \geq 2$, the situation is very different:

> Pick any collection of m (unlabeled) points on the unit sphere in $\mathbf{R^d}$, for $d \geq 2$, and assume their hidden labels correspond perfectly to some linear separator. Then there are target hypotheses in $\widehat{H}$ which cannot be identified without querying *all* the labels.

What if the active learner is not required to identify exactly the right hypothesis, but something close to it? This and other little variations don't help much.

Therefore, even in the most benign situations, we cannot expect that *every* target hypothesis will be identifiable using $o(m)$ labels. To put it differently, in the worst case over target hypotheses, active learning gives no improvement in sample complexity.

But hopefully, *on average* (with respect to some distribution over target hypotheses), the number of labels needed is small. For instance, when $d = 2$ in the bad case above, a target hypothesis chosen uniformly at random from $\widehat{H}$ can be identified by querying just $O(\log m)$ labels in expectation. This motivates the main model of this paper.

An average-case model

We will count the expected number of labels queried when the target hypothesis is chosen from some distribution π over $\widehat{H}$. This can be interpreted as a Bayesian setting, but it is more accurate to think of π merely as a device for averaging query counts, which has no bearing on the final generalization bound. A natural choice is to make π uniform over $\widehat{H}$.

Most existing active learning schemes work with H rather than $\widehat{H}$; but $\widehat{H}$ reflects the underlying combinatorial structure of the problem, and it can't hurt to deal with it directly. Often π can chosen to mask the structure of $\widehat{H}$; for instance, if H is the set of linear separators, then $\widehat{H}$ is a set of convex regions of H, and π can be made proportional to the volume of each region. This makes the problem continuous rather than combinatorial.

What is the expected number of labels needed to identify a target hypothesis chosen from π? In this average-case setting, is it always possible to get away with $o(m)$ labels, where m is the sample complexity of the supervised learning problem as defined above? We show that the answer, once again, is sadly no. Thus the benefit of active learning is really a function of the specific hypothesis class and the particular pool of unlabeled data. Depending on these, the expected number of labels needed lies in the following range (within constants):

ideal case:	$d \log m$	perfect binary search
worst case:	m	all labels, or randomly chosen queries

Notice the exponential gap between the top and bottom of this range. Is there some simple querying strategy which *always* achieves close to the minimum (expected) number of labels, whatever this minimum number might be?

Our main result is that this property holds for a variant of a popular greedy scheme: always ask for the label which most evenly divides the current effective version space weighted by π. This doesn't necessarily minimize the number of queries, just as a greedy decision tree algorithm need not produce trees of minimum size. However:

> When π is uniform over $\widehat{H}$, the expected number of labels needed by this greedy strategy is at most $O(\ln |\widehat{H}|)$ times that of any other strategy.

We also give a bound for arbitrary π, and show corresponding lower bounds in both the uniform and non-uniform cases.

Variants of this greedy scheme underlie many active learning heuristics, and are often described as optimal in the literature. This is the first rigorous validation of the scheme in a general setting. The performance guarantee is significant: recall $\log |\widehat{H}| = O(d \log m)$, the minimum number of queries possible.

Relation to other active learning work

Existing active learning methods are in large part amalgams of three orthogonal techniques:

1. generalized binary search, the subject of this paper;

2. opportunistic priors, or algorithmic luckiness;

3. Bayesian assumptions – knowledge of a prior upon which the generalization bound is based.

The first technique is specific to active learning, and holds great potential for saving labels. In this paper, we have abstracted it out in order to study it rigorously. In the final section of the paper, we also discuss the other two techniques briefly.

2 Preliminaries

Let $\mathcal{X}$ be the input space, $\mathcal{Y} = \{0, 1\}$ the space of labels, and P an unknown underlying distribution over $\mathcal{X} \times \mathcal{Y}$. We want to select a hypothesis (a function $\mathcal{X} \to \mathcal{Y}$) from some class H of VC dimension $d < \infty$, which will accurately predict labels of points in $\mathcal{X}$. We will assume that the problem is *realizable*, that is, there is some hypothesis in H which gives a correct prediction on every point. Suppose that points $(x_1, y_1) \ldots, (x_m, y_m)$ are drawn randomly from P. Standard bounds give us a function $m(\epsilon, d)$ such that if we want a hypothesis of error $\leq \epsilon$ (on P, modulo some fixed confidence level), and if $m \geq m(\epsilon, d)$, then we need only pick a hypothesis $h \in H$ which is consistent with these labeled points [9].

Now suppose just the pool of unlabeled data $x_1, \ldots, x_m$ is available. The possible labelings of these points form a subset of $\{0, 1\}^m$, the *effective hypothesis class*

$$\widehat{H} \;\cong\; \{(h(x_1), \ldots, h(x_m)) : h \in H\}.$$

Alternatively, we can think of the unlabeled points as carving H into a finite number of regions; any two hypotheses within a region are indistinguishable. Sauer's lemma [9] then tells us $|\widehat{H}| = O(m^d)$. We want to pick some $h \in H$, or equivalently the unique $h \in \widehat{H}$, which is consistent with all the hidden labels, by querying just a few of them.

Any deterministic search strategy can be represented as a binary tree whose internal nodes are queries ("what is the x_i's label?"), and whose leaves are elements of $\widehat{H}$. Henceforth we will blur the distinction between a tree and the particular querying strategy that it embodies.[1]

3 Some bad news

Our initial hope was that for hypothesis classes of interest, such as linear separators, it might always be possible to construct query trees of height $o(m)$, at least for benign data distributions. This is not the case.

Claim 1 *Let H be the hypothesis class of linear separators in $\mathbf{R}^2$. For any set of m distinct data points on the perimeter of the unit circle, there are always some target hypotheses in $\widehat{H}$ which cannot be identified without querying all m labels.*

[1] We can accommodate randomization – for instance, to allow a random choice of query point – by letting internal nodes of the tree be random coin flips. Our main result, Theorem 3, is unaffected by this generalization.

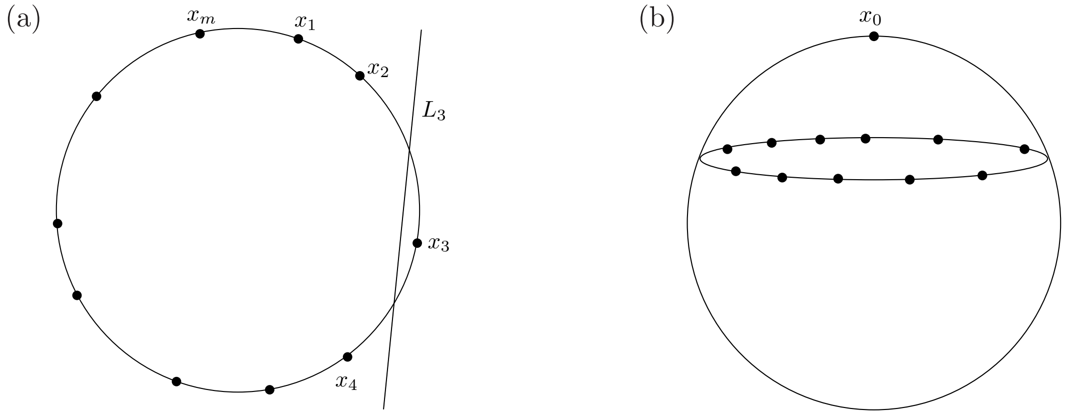

Figure 2: (a) To identify target hypotheses like L_3, we need to see *all* the labels. (b) Same thing, but in three dimensions, and with an additional point x_0 which takes up half of the total probability mass (eg. repeat the same point m times). Think of x_0 as having a positive label.

Proof. To see this, consider the following realizable labelings (Figure 2(a)):

- Labeling L_0: all points are negative.
- Labeling L_i $(1 \leq i \leq m)$: all points are negative except x_i.

It is impossible to distinguish these cases without seeing *all* the labels.[2] ▮

Remark. To rephrase this example in terms of our original goal of learning a linear separator with error $\leq \epsilon$, suppose the input distribution $P(\mathcal{X})$ is a density over the perimeter of the unit circle. Then no matter what this density is, there are always certain target hypotheses in H which force us to ask for $\Omega(1/\epsilon)$ labels: no improvement over the sample complexity of supervised learning.

In this example, the bad target hypotheses have a large imbalance in probability mass between their positive and negative regions. Is this what makes active learning difficult? Not so – by adding an extra dimension and an extra point, exactly the same example can be modified to make the bad hypotheses balanced (Figure 2(b)).

So let's return to the original 2-d case. We know that some of the labelings $L_0, \ldots, L_m \in \widehat{H}$ must lie at depth m in any query tree. But these are just a small fraction of $\widehat{H}$: what about the rest? Well, suppose for convenience that $x_1, \ldots, x_m$ are in clockwise order around the unit circle. Then $\widehat{H} = \{h_{ij} : 1 \leq i \neq j \leq m\} \cup \{h_0, h_1\}$, where h_{ij} labels $x_i \cdots x_{j-1}$ positive (if $j < i$ it wraps around) and the remaining points negative, and h_0, h_1 are everywhere negative/positive. It is quite easy to construct a query tree in which each h_{ij} lies at depth $\leq 2(m/|j-i| + \log|j-i|)$. Thus, if the target hypothesis is chosen uniformly from $\widehat{H}$, the *expected* number of labels queried is at most

$$\frac{1}{m(m-1)+2}\left\{2m + \sum_{i \neq j} 2(m/|j-i| + \log|j-i|)\right\} \;=\; O(\log m).$$

This is why we place our hopes in an average-case analysis.

[2] What if the final hypothesis – considered as a point in $\{0,1\}^m$ – doesn't have to be exactly right, but within Hamming distance k of the correct one? Then a similar example forces $\Omega(m/k)$ queries.

4 Main result

Let π be any distribution over $\widehat{H}$; we will analyze search strategies according to the number of labels they require, averaged over target hypotheses drawn from π. In terms of query trees, this is the average depth of a leaf chosen according to π. Specifically, let T be any tree whose leaves include the support of π. The quality of this tree is

$$Q(T,\pi) \;=\; \sum_{h\in\widehat{H}} \pi(h)\cdot(\#\text{ labels needed for } h) \;=\; \sum_{h\in\widehat{H}} \pi(h)\cdot \text{leaf-depth}(h).$$

We will occasionally consider trees whose leaves are a superset of π's support.

Is there always a tree of average depth $o(m)$? The answer, once again, is sadly no.

Claim 2 *Pick any $d \geq 2$ and any $m \geq 2d$. There is an input space $\mathcal{X}$ of size m and a hypothesis class H of VC dimension d, defined on domain $\mathcal{X}$, with the following property: if π is chosen to be uniform over $H = \widehat{H}$, then any query tree T has $Q(T,\pi) \geq m/8$.*

Proof. Let $\mathcal{X}$ consist of any m points $x_1, \ldots, x_m$, and let H consist of all hypotheses $h : \mathcal{X} \to \{0,1\}$ which are positive on exactly d inputs. In order to identify a particular element $h \in H$, any querying method must discover exactly the d points x_i on which h is nonzero. By construction, the order in which queries are asked is irrelevant – it might as well be $x_1, x_2, \ldots$. Simple probability calculations then show the average tree height to be at least $m/8$. ∎

In our average-case model, we have now seen two examples, one in which intelligent querying results in an exponential improvement in the number of labels required, and one in which it is no help at all. In general, the benefit of active learning will depend upon the particular hypothesis class and particular pool of unlabeled points. Therefore, our best hope is that there is some simple, generic scheme which *always* comes close to minimizing the number of queries, whatever the minimum number might be.

Here is a natural greedy approach.

> **Greedy strategy.** Let $S \subseteq \widehat{H}$ be the current version space. For each unlabeled x_i, let S_i^+ be the hypotheses which label x_i positive and S_i^- the ones which label it negative. Pick the x_i for which these sets are most nearly equal in π-mass, that is, the x_i for which $\min\{\pi(S_i^+), \pi(S_i^-)\}$ is largest.

We show that this simple deterministic strategy is almost as good at minimizing queries as *any* other strategy.

Theorem 3 *Let π be any distribution over $\widehat{H}$. Suppose that the optimal query tree requires Q^* labels in expectation, for target hypotheses chosen according to π. Then the expected number of labels needed by the greedy strategy is at most $4Q^* \ln 1/(\min_h \pi(h))$.*

For the case of uniform π, the approximation ratio is thus at most $4 \ln |\widehat{H}|$. We also show almost-matching lower bounds in both the uniform and non-uniform cases.

5 Analysis of the greedy active learner

5.1 Lower bounds on the greedy scheme

At the root of any query tree, when all of $\widehat{H}$ is possible, the entropy of the distribution over possible hypotheses is $\mathbf{H}(\pi)$. At other nodes, the version space is some $S \subseteq \widehat{H}$, with associated entropy $\mathbf{H}(\pi_S)$, where π_S is the restriction of π to S: for $h \in S$, $\pi_S(h) = \pi(h)/\pi(S)$. The greedy approach builds a query tree top-down, trying to reduce entropy as quickly as possible.

This approach is not optimal because it doesn't take into account the way in which a query reshapes the search space – specifically, the effect of a query on the quality (extent of entropy reduction) of *other* queries. For instance, $\widehat{H}$ might consist of several dense clusters, each of which permits rapid binary search. However, the version space must first be whittled down to one of these subregions, and this process, though ultimately optimal, might initially be slower at decreasing entropy than more shortsighted alternatives. A concrete example of this type gives rise to the following lower bound.

Claim 4 *For any integer $n \geq 16$ which is a power of two, there is a concept class $\widehat{H}_n$ of size n with the following property: under uniform π, the optimal tree has average height at most $q_n = \Theta(\log n)$, but the greedy active learning strategy produces a tree of average height $\Omega(q_n \cdot \frac{\log n}{\log \log n})$.*

What about non-uniform π? In this case, the additional knowledge about the distribution of target hypotheses might make active learning easier, but what impact does it have on the approximation ratio of a greedy learner? It turns out that the greedy scheme can deviate more substantially from optimality.

Claim 5 *Pick any $n \geq 2$. There exists a hypothesis class $\widehat{H}$ with $2n+1$ elements and a distribution π over $\widehat{H}$, such that: (a) π ranges in value from $1/2$ to $1/2^{n+1}$; (b) the optimal query tree has average depth less than 3; (c) the greedy query tree has average depth at least $n/2$.*

The proofs of both lower bounds can be found in the appendix.

5.2 Upper bound: overview

The two lower bounds on the quality of a greedy active learner are sobering, but fortunately, things cannot get too much worse than this. Here's the basic argument for the case of uniform π: we start by showing that if the optimal tree T^* requires Q^* queries in expectation, then there must be some good query which (again in expectation) "cuts off" a chunk of $\widehat{H}$ whose π-mass is at least $\Omega(1/Q^*)$. Therefore, we can be sure that the root query of the greedy tree T_G is at least this good (cf. Johnson's set cover analysis [8]). Things get trickier when we try to apply this same argument to show that the rest of T_G is also good. The problem is that although T^* uses just Q^* queries *on average*, there may well be some particular hypotheses for which it uses lots more queries. Therefore, subtrees of T_G could correspond to version spaces for which more than Q^* queries are needed, and the roots of these subtrees might not cut down the version space much. Care is needed to limit the effect of such subtrees.

For a worst-case model, a proof of approximate optimality is known in a related context [6]; as we saw in Claim 1, that model is trivial in our situation. The average-case model, and especially the use of arbitrary weights π, require more care.

5.3 Upper bound: details

The key concept we have to define is the *quality* of a query, and it turns out that we need this to be monotonically decreasing, that is, it should only go down as active learning proceeds and the version space shrinks. This rules out some natural entropy-based notions.

Suppose we have narrowed the candidate hypotheses down to some version space $S \subseteq \widehat{H}$, and a possible next query is x_j. How many elements are eliminated by this query, or more precisely, what probability mass (measured by π) is eliminated on average? If S^+ is the subset of S which labels x_j positive, and S^- are the ones that label it negative, then on average the probability mass eliminated is

$$\frac{\pi(S^+)}{\pi(S)}\pi(S^-) + \frac{\pi(S^-)}{\pi(S)}\pi(S^+) \;=\; \frac{2\pi(S^+)\pi(S^-)}{\pi(S)}.$$

We say x_j *shrinks* (S, π) by this much, with the understanding that this is in expectation.

We now confirm that shrinkage indeed has the monotonicity property we need.

Lemma 6 *Suppose query x_j shrinks $(\widehat{H}, \pi)$ by Δ. Then for any $S \subseteq \widehat{H}$, x_j shrinks (S, π) by at most Δ.*

Proof. This follows by rewriting the expression for shrinkage as

$$\frac{2}{\frac{1}{\pi(S^+)} + \frac{1}{\pi(S^-)}}.$$

Therefore, the shrinkage decreases monotonically as $\pi(S^+), \pi(S^-)$ decrease. ▮

We would expect that if the optimal tree is short, there must be at least one query which shrinks $(\widehat{H}, \pi)$ considerably. More concretely, the definition of shrinkage seems to suggest that if all queries provide shrinkage at most Δ, and the current version space has mass $\pi(S)$, then at least about $\pi(S)/\Delta$ more queries are needed. This isn't entirely true, because of a second effect – if there are just two hypotheses left in S, then we need just one query, regardless of $\pi(S)$.

Roughly speaking, when there are lots of hypotheses with significant mass left in S, the first effect dominates; thereafter the second takes over. To smoothly incorporate both effects, we use the notion of *collision probability*. For a distribution ν over support $\mathcal{Z}$, this is

$$\mathsf{CP}(\nu) = \sum_{z \in \mathcal{Z}} \nu(z)^2,$$

the chance that two random draws from ν are identical.

Lemma 7 *Suppose that every query shrinks $(\widehat{H}, \pi)$ by at most some $\Delta > 0$. Then for any $S \subseteq \widehat{H}$, and any query tree T whose leaves include S,*

$$Q(T, \pi_S) \;\geq\; \frac{\pi(S) \cdot (1 - \mathsf{CP}(\pi_S))}{\Delta}.$$

Proof. We use induction on $|S|$. When $|S| = 1$, $\mathsf{CP}(\pi_S) = 1$ and so the claim holds trivially. Now consider any $S \subseteq \widehat{H}$ of size > 1. Suppose the optimal tree for π_S starts with query x_j, which splits S into S^+ and S^- with masses $p\pi(S), (1-p)\pi(S)$ respectively. By the induction hypothesis, the average number of queries needed by subtrees corresponding to S^+ and S^- are at least $\pi(S^+)(1-\mathsf{CP}(\pi_{S^+}))/\Delta$ and $\pi(S^-)(1-\mathsf{CP}(\pi_{S^-}))/\Delta$, respectively. Therefore the expected number of queries needed for π_S is at least

$$\begin{aligned} & 1+p\cdot\frac{\pi(S^+)(1-\mathsf{CP}(\pi_{S^+}))}{\Delta}+(1-p)\cdot\frac{\pi(S^-)(1-\mathsf{CP}(\pi_{S^-}))}{\Delta} \\ =\ & 1+\frac{\pi(S)}{\Delta}\left\{p^2+(1-p)^2-p^2\mathsf{CP}(\pi_{S^+})-(1-p)^2\mathsf{CP}(\pi_{S^-})\right\} \\ =\ & 1+\frac{\pi(S)}{\Delta}\left\{1-2p(1-p)-\mathsf{CP}(\pi_S)\right\} \end{aligned}$$

From the previous lemma, x_j shrinks (S,π) by at most Δ, that is, $2p(1-p)\pi(S)\leq\Delta$. Combining this with the formula above, we find that the expected number of queries needed for S is at least $\pi(S)(1-\mathsf{CP}(\pi_S))/\Delta$, completing the induction. ▮

Corollary 8 *Pick any $S\subseteq\widehat{H}$ and any tree T whose leaves include all of S. Then there must exist a query which shrinks (S,π_S) by at least $(1-\mathsf{CP}(\pi_S))/Q(T,\pi_S)$.*

This last corollary is very helpful in analyzing greedy active learning. It tells us that if the current version space $S\subseteq\widehat{H}$ is such that π_S has small collision probability, then there must be some query which splits off a reasonable chunk of S. This principle can form the basis of a proof by induction.

But what if π_S has high collision probability, say greater than 1/2? In this case, the mass of some particular hypothesis $h_0\in S$ exceeds that of all the others combined, and S could shrink by just an insignificant amount during the subsequent greedy query, or even during the next few iterations of greedy queries. We will argue, however, that within roughly the number of iterations that the optimal tree needs for target h_0, the greedy procedure will either reject h_0 or identify it as the target. If it is rejected, then *by that time* S will have shrunk considerably.

Our main theorem is an immediate consequence of the following lemma.

Lemma 9 *Let T^* denote any particular query tree for π (for instance, the optimal tree), and let T be the greedily-constructed query tree. For any $S\subseteq\widehat{H}$ which corresponds to a subtree T_S of T,*

$$Q(T_S,\pi_S)\ \leq\ 4Q(T^*,\pi_S)\ln\frac{\pi(S)}{\min_{h\in S}\pi(h)}.$$

Proof. Again, we proceed by induction on $|S|$. When $|S|=1$, the claim holds trivially. So consider any $S\subseteq\widehat{H}$ of size >1. We will look at two cases:

Case one: $\mathsf{CP}(\pi_S)\leq 1/2$

Suppose the greedy query splits S into S^+ and S^-, with $p=\pi(S^+)/\pi(S)$. We'll use $Q^*(A)$ as a shorthand for $Q(T^*,\pi_A)$, and $\min(A)$ as a shorthand for $\min_{h\in A}\pi(h)$. As an example of this notation, observe that $Q^*(S)=pQ^*(S^+)+(1-p)Q^*(S^-)$. The number of queries needed by the greedy learner is

$$\begin{aligned} Q(T_S,\pi_S) &= 1+pQ(T_{S^+},\pi_{S^+})+(1-p)Q(T_{S^-},\pi_{S^-}) \\ &\leq 1+4pQ^*(S^+)\ln\frac{\pi(S^+)}{\min(S^+)}+4(1-p)Q^*(S^-)\ln\frac{\pi(S^-)}{\min(S^-)} \\ &\leq 1+4\left(pQ^*(S^+)+(1-p)Q^*(S^-)\right)\ln\frac{\max\{\pi(S^+),\pi(S^-)\}}{\min(S)} \end{aligned}$$

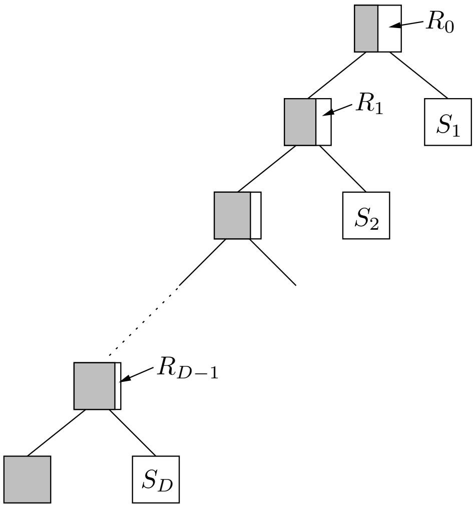

Figure 3: The greedily-constructed tree T_S. The shaded region denotes the single hypothesis h_0 which dominates S. At depth d, the hypotheses which remain to be distinguished from h_0 are R_d.

$$\begin{aligned} &= 1 + 4Q^*(S) \ln \frac{\pi(S)}{\min(S)} + 4Q^*(S) \ln \max\{p, 1-p\} \\ &\leq 1 + 4Q^*(S) \ln \frac{\pi(S)}{\min(S)} - 4Q^*(S) \min\{p, 1-p\} \end{aligned}$$

where the first inequality follows from the induction hypothesis and the last one follows from $\ln(1-x) \leq -x$. Since $\mathsf{CP}(\pi_S) \leq 1/2$, we have by Corollary 8 that

$$\min\{p, 1-p\} \;\geq\; p(1-p) \;\geq\; \frac{1 - \mathsf{CP}(\pi_S)}{2Q^*(S)} \;\geq\; \frac{1}{4Q^*(S)}.$$

Plugging this into the equation above, we find $Q(T_S, \pi_S) \leq 4Q^*(S) \ln \frac{\pi(S)}{\min(S)}$, as desired.

Case two: $\mathsf{CP}(\pi_S) > 1/2$

Since $\mathsf{CP}(\pi_S) \leq \max_{h \in S} \pi_S(h)$, it follows that there is a single hypothesis h_0 which takes up more than half the probability mass of S. Suppose that h_0 lies at depth D in the greedy tree T_S. At depth $d < D$ of T_S, let R_d be the other hypotheses which still remain to be distinguished from h_0 (Figure 3). At this node of the tree, the greedy learner is simply trying to cut off as much of R_d as possible. Let S_{d+1} be the amount it succeeds in removing, ie. $R_d = R_{d+1} \cup S_{d+1}$. Since h_0 can be uniquely identified using $Q^*(h_0)$ queries, we know that it is always possible to cut off a $1/Q^*(h_0)$ fraction of R_d:

$$\pi(R_{d+1}) \;=\; \pi(R_d) - \pi(S_{d+1}) \;\leq\; \pi(R_d) - \frac{\pi(R_d)}{Q^*(h_0)}$$

Hence for $d \geq 0$

$$\pi(R_d) \;\leq\; \left(1 - \frac{1}{Q^*(h_0)}\right)^d \pi(R_0) \;\leq\; \pi(R_0) e^{-d/Q^*(h_0)}.$$

This tells us that $D \leq Q^*(h_0) \ln(\pi(R_0)/\min(R_0))$.

By considering h_0 separately from the other elements of S, we get

$$\begin{aligned}
Q(T_S, \pi_S) &= \frac{1}{\pi(S)} \left\{ \pi(h_0) \cdot D + \sum_{d=1}^{D} \pi(S_d) \cdot (d + Q(T_{S_d}, \pi_{S_d})) \right\} \\
&\leq \frac{1}{\pi(S)} \left\{ \pi(h_0) \cdot D + \sum_{d=1}^{D} \pi(S_d) \cdot d + \sum_{d=1}^{D} \pi(S_d) \cdot 4Q^*(S_d) \ln \frac{\pi(S_d)}{\min(S_d)} \right\} \\
&\leq \frac{1}{\pi(S)} \left\{ \pi(h_0) \cdot D + \sum_{d=1}^{D} \pi(S_d) \cdot d + 4\pi(R_0)Q^*(R_0) \ln \frac{\pi(R_0)}{\min(R_0)} \right\}.
\end{aligned}$$

Since at each depth d at least a $1/Q^*(h_0)$ fraction of the mass of R_d is cut away, the sum $\sum_d \pi_{R_0}(S_d)d$ is at most $Q^*(h_0)$, the expected number of trials before seeing "heads" when tossing a coin with heads probability $1/Q^*(h_0)$. Therefore, $\sum_d \pi(S_d)d \leq \pi(R_0)Q^*(h_0)$, and pulling in our earlier bound on D,

$$\begin{aligned}
Q(T_S, \pi_S) &\leq \frac{1}{\pi(S)} \left\{ \pi(h_0) \cdot D + \pi(R_0)Q^*(h_0) + 4\pi(R_0)Q^*(R_0) \ln \frac{\pi(R_0)}{\min(R_0)} \right\} \\
&\leq \frac{1}{\pi(S)} \left\{ \pi(h_0)Q^*(h_0) \ln \frac{\pi(R_0)}{\min(R_0)} + \pi(R_0)Q^*(h_0) + 4\pi(R_0)Q^*(R_0) \ln \frac{\pi(R_0)}{\min(R_0)} \right\} \\
&\leq \frac{1}{\pi(S)} \left\{ \pi(h_0)Q^*(h_0) \ln \frac{\pi(S)}{\min(S)} + \pi(h_0)Q^*(h_0) + 4\pi(R_0)Q^*(R_0) \ln \frac{\pi(S)}{\min(S)} \right\} \\
&\leq \frac{1}{\pi(S)} \left\{ 4\pi(h_0)Q^*(h_0) + 4\pi(R_0)Q^*(R_0) \right\} \ln \frac{\pi(S)}{\min(S)} \\
&= 4Q^*(S) \ln \frac{\pi(S)}{\min(S)}
\end{aligned}$$

where we have made use of $\pi(S) \geq 2\min(S)$ and $\pi(h_0) \geq \pi(R_0)$. This completes case two, and with it the inductive proof. ▮

6 Related work

Rather than attempting to summarize the wide range of proposed active learning methods, for instance [5, 7, 10, 13, 14], we will discuss three basic techniques upon which they rely.

Greedy search

This is the technique which we have abstracted and rigorously validated in this paper. It is the foundation of most of the schemes cited above. Algorithmically, the main problem is that the query selection rule is not immediately tractable, so approximations are necessary and often require considerable ingenuity. For instance, for linear separators, $\widehat{H}$ consists of convex sets, and if π is chosen to be proportional to volume, query selection involves estimating volumes of convex regions, which is tractable but (using present techniques) inconvenient. Tong and Koller [13] investigate margin-based approximations which are efficiently computable using standard SVM technology.

Opportunistic priors, or algorithmic luckiness

This is a trick in which the learner takes a look at the unlabeled data and then places bets on hypotheses. A uniform bet over all of $\widehat{H}$ leads to standard VC generalization bounds. But if the algorithm places more weight on certain hypotheses (for instance, those with large margin), then its final error bound is excellent if it guessed right, and worse-than-usual if it guessed wrong. This technique is not specific to active learning, and has been analyzed elsewhere (for instance, [12]). One interesting line of work has investigated a flexible family of such priors, specified by pairwise similarities between data points, eg. [14].

Bayesian assumptions

In our analysis, although π can be seen as some sort of prior belief, there is no assumption that nature shares this belief; in particular, the generalization bound does not depend on it. A Bayesian assumption has an immediate benefit for active learning: if at any stage the remaining version space (weighted by prior π) is largely in agreement on the unlabeled data, it is legitimate to stop and output one of these remaining hypotheses [7]. In a non-Bayesian setting this would be unthinkable.

When the hypothesis class consists of probabilistic classifiers, the Bayesian assumption has also been used in another way: to approximate the greedy selection rule using the MAP estimate instead of an expensive summation over the posterior (eg. [11]).

In terms of theoretical results, another work which considers the tradeoff between labels and generalization error is [7], in which a greedy scheme, realized using sampling, is analyzed in a Bayesian setting. The authors show that it is possible to achieve an exponential improvement in the number of labels needed to learn linear separators, when both data and target hypothesis are chosen uniformly from the unit sphere. It is an intriguing question whether this holds for more general data distributions.

7 Promising directions

We have looked at the case where the acceptable error rate is fixed and the goal is to minimize the number of queries. What about the dual question of fixing the number of queries and asking for the best (average) error rate possible? In other words, the query tree has a fixed depth, and each leaf is annotated with some subset $S \subseteq \widehat{H}$, the remaining version space at that leaf. If each element of S is thought of as a point in $\{0,1\}^m$ (its predictions on the pool of data), then the error at this leaf depends on the Hamming diameter of S. What is a good querying strategy for producing low-diameter leaves?

Perhaps the most widely-used specific hypothesis class is that of linear separators. Existing active learning schemes ignore the rich algebraic structure of $\widehat{H}$, an *arrangement of hyperplanes* [4].

Acknowledgements

I am very grateful to the anonymous NIPS reviewers for their careful and detailed feedback.

References

[1] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[2] D. Angluin. Queries revisited. *Proceedings of the Twelfth International Conference on Algorithmic Learning Theory*, pages 12–31, 2001.

[3] E.B. Baum and K. Lang. Query learning can work poorly when a human oracle is used. *International Joint Conference on Neural Networks*, 1992.

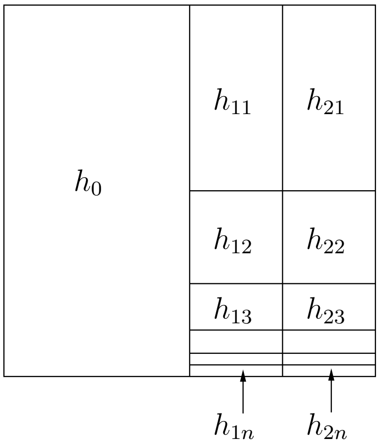

Figure 4: A hypothesis class with $2n+1$ elements. Area is proportional to π-mass.

[4] A. Bjorner, M. Las Vergnas, B. Sturmfels, N. White, and G. Ziegler. *Oriented matroids.* Cambridge University Press, 1999.

[5] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[6] S. Dasgupta, P.M. Long, and W.S. Lee. A theoretical analysis of query selection for collaborative filtering. *Machine Learning*, 51:283–298, 2003.

[7] Y. Freund, S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.

[8] D.S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[9] M.J. Kearns and U.V. Vazirani. *An introduction to computational learning theory.* MIT Press, 1993.

[10] A. McCallum and K. Nigam. Employing em and pool-based active learning for text classification. *Fifteenth International Conference on Machine Learning*, 1998.

[11] N. Roy and A. McCallum. Toward optimal active learning through sampling of error reduction. *Twentieth International Conference on Machine Learning*, 2003.

[12] J. Shawe-Taylor, P. Bartlett, R. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 1998.

[13] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2001.

[14] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. *ICML workshop*, 2003.

8 Appendix on lower bounds

8.1 Claim 5: the non-uniform case

Fix any $n \geq 2$. Consider a hypothesis class with $2n+1$ elements $H = \{h_0, h_{1i}, h_{2i} : i = 1, \ldots, n\}$, and corresponding weights π:

$$\pi(h_0) = 1/2, \;\; \pi(h_{1i}) = \pi(h_{2i}) = 1/2^{i+2} \text{ for } i < n, \;\; \pi(h_{1n}) = \pi(h_{2n}) = 1/2^{n+1}.$$

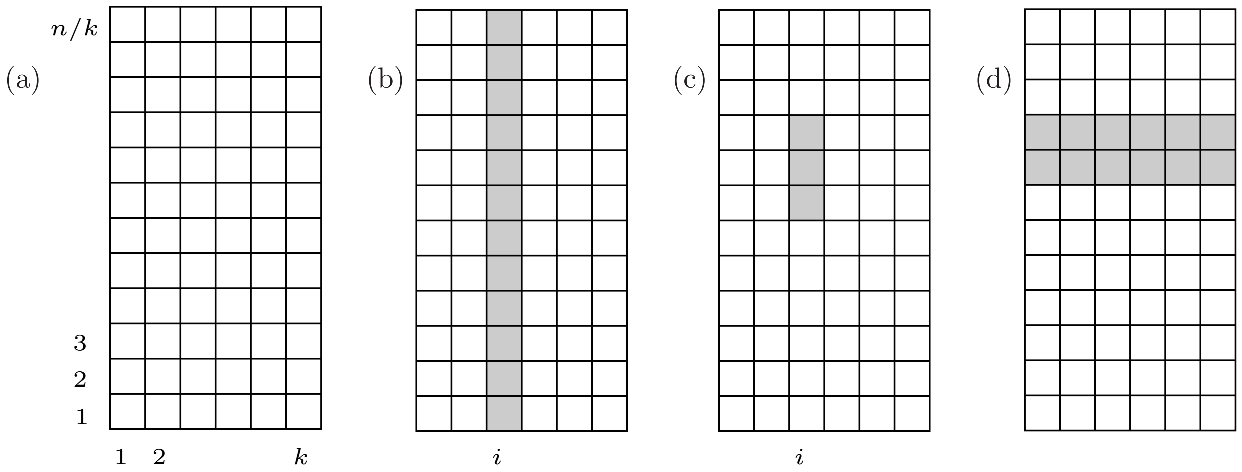

Figure 5: (a) The hypothesis class is $\{1, 2, \ldots, k\} \times \{1, 2, \ldots, n/k\}$. (b) Queries $Q^{(1)}$ decide whether the target lies in a given column. (c) Queries $Q^{(2)}$ enable binary search within a column. (d) Queries $Q^{(3)}$ test certain subsets of contiguous rows.

This information is summarized in Figure 4. Suppose that the unlabeled data correspond to the following set of queries:

Query q_1 returns 1 if the target is some h_{1i}.

Query q_2 returns 1 if the target is some h_{2i}.

Query q_{3i} ($i = 1, \ldots, n$) returns 1 if the target is either h_{1i} or h_{2i}.

A sensible strategy is to first use q_1, q_2 to figure out which "column" (of Figure 4) contains the target, and then, if it turns out to be one of the right-hand columns, to narrow down the choices using $q_{31}, q_{32}, \ldots$. The expected number of labels queried is then:

$$\begin{aligned}
\pi(h_0) \cdot \#\text{labels}(h_0) + \sum_{i=1}^{n} \pi(h_{1i}) \cdot \#\text{labels}(h_{1i}) + \sum_{i=1}^{n} \pi(h_{2i}) \cdot \#\text{labels}(h_{2i}) \\
&\leq \frac{1}{2} \cdot 2 + \sum_{i=1}^{\infty} \frac{1}{2^{i+2}} \cdot (1+i) + \sum_{i=1}^{\infty} \frac{1}{2^{i+2}} \cdot (2+i) \\
&= 1 + \frac{3}{4} + 1 \quad < \quad 3
\end{aligned}$$

A greedy learner, however, could wander down a different path: it might start with $q_{31}, q_{32}, q_{33}, \ldots$. If any of these succeeds, it would then distinguish between the remaining two cases h_{1i}, h_{2i} using q_1. If none of the q_{3i} succeed, it would conclude h_0. This latter event requires n queries and occurs with probability 1/2. Therefore the expected number of queries is at least $n/2$.

8.2 Claim 4: the uniform case

Pick any integers $2 \leq k < n$ such that k divides n. Consider a class of n hypotheses $H = \{h_{ij} : 1 \leq i \leq k, 1 \leq j \leq n/k\}$, with uniform weights π over them. Suppose that the unlabeled data is such that there are three types of queries (Figure 5):

Queries $Q^{(1)} = \{q_1, \ldots, q_k\}$; q_i returns 1 if the target is in column i (see figure).

Queries $Q^{(2)}$ permit binary search within each column.

Queries $Q^{(3)} = \{q_{ij} : \text{selected } (i,j)\}$; q_{ij} returns 1 if the target lies somewhere in rows i to j. The available pairs (i,j) are described below.

Queries $Q^{(2)}$ tell us that the hypothesis class consists of k clusters (the columns), each of which is rapidly searchable. A good strategy is therefore to first identify the correct column, which requires the relatively inefficient queries $Q^{(1)}$, and to then perform a fast binary search. The expected number of queries for such a scheme is at most $(k+1)/2 + \lg n/k + 1$.

A greedy learner, however, does not have any lookahead capability, and will not understand the long-term potential of queries $Q^{(1)}$. It might instead use $Q^{(3)}$, which always allow a $1/k$ fraction of the remaining rows to be pruned away. For instance, there is a query q_{ij} for the interval $[i,j] = [1, \lceil n/k^2 \rceil]$, the first $1/k$ fraction of rows. This query has two outcomes; for either one, there is a corresponding query which encompasses $1/k$ of what remains (making sure to always round up) – and so on.

How many of these queries $Q^{(3)}$ are needed (in expectation) before the row containing the target is uniquely identified? Let's call this number $N(r)$, where r is the number of rows left (initially $r = n/k$). Then, ignoring rounding effects,

$$N(r) = 1 + \tfrac{1}{k} N(r/k) + (1 - \tfrac{1}{k}) N(r(1 - 1/k)), \text{ and } N(1) = 0.$$

The solution to this recurrence is

$$N(r) \;=\; \frac{\log_2 r}{\mathbf{H}(1/k)} \;\geq\; \frac{k \log r}{2 \log k},$$

where $\mathbf{H}(p)$ denotes the entropy of a coin with bias p. Rounding changes this only slightly – we can show $N(r) \geq (k \log r/k)/(4 \log k/2)$. Thus the expected number of queries needed by the greedy strategy is at least

$$\frac{k \log n/k^2}{4 \log k/2} + \frac{k}{2}.$$

Comparing this to the optimal strategy given above, and choosing $k \approx \lg n$ (if n is a power of two, then some number in the range $[\lg n, 2 \lg n]$ must divide n), we get Claim 4.