Activized Learning: Transforming Passive to Active with Improved Label Complexity

Steve Hanneke
Department of Statistics
Carnegie Mellon University
shanneke@stat.cmu.edu

Introduction

Active learning methods often achieve improved performance using fewer labels compared to passive learning methods. A variety of practically successful active learning algorithms use a passive learning algorithm as a subroutine, and the essential role of the active component is to construct data sets to feed into the passive subroutine. This general idea is appealing for a variety of reasons, as it may be able to inherit the tried-and-true learning bias of a given passive algorithm for a particular type of learning problem, while gaining further savings from focusing on informative examples. With few exceptions, so far these reduction-based active learning algorithms have been designed largely on a case-by-case basis, and are often based on rough heuristics such as ranking the unlabeled points by informal notions of "confidence." This raises the question of whether there exist general strategies for constructing active learning algorithms based on passive algorithms, which allow for theoretical guarantees of improved performance. In this extended abstract, we formalize this objective, explore the failings of existing methods, and propose a new method which provably succeeds in a very general sense.

First, let us fix a particular formal model for active learning, similar in style to the PAC model of passive learning. In this scenario, we have an *instance space* $\mathcal{X}$, and a set $\mathbb{C}$ of classifiers $h : \mathcal{X} \to \{-1, +1\}$ known as the *concept space*. We denote by d the VC dimension of $\mathbb{C}$, and throughout this work we will assume $d < \infty$ (such a $\mathbb{C}$ is called a *VC class*). There is additionally an unknown distribution $\mathcal{D}$ on $\mathcal{X}$, known as the *target distribution*, and a special classifier, $h^* \in \mathbb{C}$, known as the *target function*. For any $h \in \mathbb{C}$, we define $er(h) = \mathbb{P}_{X\sim\mathcal{D}}(h(X) \neq h^*(X))$, the *error rate* of h.

For the learning problem, there is a sequence of unlabeled examples $X_1, X_2, \ldots$ independent and identically distributed according to $\mathcal{D}$. In this active learning framework, the learning algorithm can observe this sequence, select any index i, and receive the value $h^*(X_i)$, then select another index j, and receive the value $h^*(X_j)$, etc. Each time it selects an index, we refer to that as a "label request." This process continues sequentially until the algorithm halts, at which time it outputs a classifier h. The algorithm will be provided (as an argument) with a predefined value n, representing its budget. An **active learning algorithm** is defined as any procedure guaranteed to halt and output a classifier without the number of label requests exceeding this budget. We are interested in the budget size necessary for the algorithm to achieve a given error rate, a quantity known as the *label complexity*, defined formally as follows.

Definition 1. *An active learning algorithm $\mathcal{A}_a$ achieves a* **label complexity** $\Lambda_a(\cdot,\cdot,\cdot)$ *if, for any distribution $\mathcal{D}$, $h^* \in \mathbb{C}$, and $\epsilon > 0$, $\forall n \geq \Lambda_a(\epsilon, h^*, \mathcal{D})$, if $h_n = \mathcal{A}_a(n)$ is the algorithm's output after at most n label requests (under target distribution $\mathcal{D}$ and target function h^*), then $\mathbb{E}[er(h_n)] \leq \epsilon$.*

A **passive algorithm** $\mathcal{A}_p$ is any method that takes as input a finite sequence of labeled examples $\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, and produces as output a classifier h. We say a passive algorithm $\mathcal{A}_p$ achieves a label complexity $\Lambda_p(\cdot,\cdot,\cdot)$ if, for any target distribution $\mathcal{D}$, target function $h^* \in \mathbb{C}$, and $\epsilon > 0$, $\forall n \geq \Lambda_p(\epsilon, h^*, \mathcal{D})$, if $h_n = \mathcal{A}_p(\{(X_1, h^*(X_1)), (X_2, h^*(X_2)), \ldots, (X_n, h^*(X_n))\})$, then $\mathbb{E}[er(h_n)] \leq \epsilon$. Additionally, by an **active meta-algorithm** $\mathcal{A}_a$, we simply mean a procedure such that, for any passive algorithm $\mathcal{A}_p$, it is true that $\mathcal{A}_a(\mathcal{A}_p, \cdot)$ is an active learning algorithm. With these definitions, we can now formalize the objective of this work.

Definition 2. *We say an active meta-algorithm $\mathcal{A}_a$* **activizes** *a passive algorithm $\mathcal{A}_p$ for $\mathbb{C}$ if, for any label complexity Λ_p achieved by A_p, the active learning algorithm $\mathcal{A}_a(\mathcal{A}_p, \cdot)$ achieves a label complexity Λ_a such that, for all distributions $\mathcal{D}$ and $h^* \in \mathbb{C}$, $\Lambda_p(\epsilon, h^*, \mathcal{D}) = O(polylog(1/\epsilon)) \Rightarrow \Lambda_a(\epsilon, h^*, \mathcal{D}) = O(polylog(1/\epsilon))$, and if $polylog(1/\epsilon) \ll \Lambda_p(\epsilon, h^*, \mathcal{D}) < \infty$, then there exists a finite constant c such that*

$$\Lambda_a(c\epsilon, h^*, \mathcal{D}) = o(\Lambda_p(\epsilon, h^*, \mathcal{D})).$$

A meta-algorithm $\mathcal{A}_a$ that activizes every *passive algorithm $\mathcal{A}_p$ for $\mathbb{C}$ is called an* **activizer** *for $\mathbb{C}$.*

In words, the objective is to asymptotically strictly dominate the passive algorithm on all nontrivial target functions and distributions. To be clear about the semantics, h^* and $\mathcal{D}$ are considered fixed constants in the $o(\cdot)$, and the asymptotics are considered as $\epsilon \downarrow 0$. The conditions involving the $polylog(1/\epsilon)$ are necessary to handle trivial scenarios, such as single-point distributions ($\exists x : \mathbb{P}(X = x) = 1$).

This objective is related to some recent results in the literature; most notably, in my prior work (joint with Maria-Florina Balcan and Jennifer Wortman) [1], we included a result of a similar type; however, in that work, we allowed the algorithms to depend on the target distribution $\mathcal{D}$ in arbitrary ways, and the algorithm constructed in the proof of that main result leverages this capability to such an extent that it seems unlikely that there is any way to remove the dependence without fundamentally altering the nature of the algorithm. As dependence on the distribution is generally unrealistic, the positive results below for this objective (which does not allow dependence on $\mathcal{D}$) represent a significant step forward in the pursuit of a theory of the advantages of active learning applicable to real learning scenarios.

A Basic Activizer

At present, one of the best-understood families of active learning algorithms are those choosing their label requests based on disagreement among classifiers in a version space; the canonical algorithm of this type is CAL [e.g., 2, 4]. To precisely characterize scenarios where disagreement-based active learning improves over passive, we will require a bit of notation. For a finite set $\mathcal{L}$ of labeled examples, and $h \in \mathbb{C}$, let $er_{\mathcal{L}}(h) = \frac{1}{|\mathcal{L}|} \sum_{(x,y)\in\mathcal{L}} \mathbb{1}[h(x) \neq y]$. For any $V \subseteq \mathbb{C}$, $x \in \mathcal{X}$, and $y \in \{-1, +1\}$, define $V_{(x,y)} = \{h \in V : h(x) = y\}$. For $r > 0$, define $B_{\mathcal{D},V}(h^*, r) = \{h \in V : \mathbb{P}_{\mathcal{D}}(h(X) \neq h^*(X)) \leq r\}$. Also define the *region of disagreement* $DIS(V) = \{x \in \mathcal{X} : \exists h_1, h_2 \in V \text{ s.t. } h_1(x) \neq h_2(x)\}$. Finally, define the *boundary* of the target function h^*, denoted $\partial_{\mathcal{D},V} h^* = \lim_{r\downarrow 0} DIS(B_{\mathcal{D},V}(h^*, r))$.

Additionally, we adopt the convention that any set of classifiers V *shatters* $\{\}$ iff $V \neq \{\}$ (and otherwise, shattering is defined as in [6], as usual). Also, for convenience, define $\mathcal{X}^0 = \{\{\}\}$.

Informally, if V is the set of classifiers consistent with all labels observed so far in a disagreement-based algorithm, then $DIS(V) \to \partial_{\mathcal{D},\mathbb{C}} h^*$, so we expect improvements from disagreement-based algorithms when $\mathbb{P}(\partial_{\mathcal{D},\mathbb{C}} h^*) = 0$; in this case, the queries become focused as the algorithm progresses. On the other hand, when $\mathbb{P}(\partial_{\mathcal{D},\mathbb{C}} h^*) > 0$, we expect the queries will not become focused, so that disagreement-based algorithms will not provide better than constant-factor improvements over passive learning. This is the case we must try to improve if we are to succeed in general. However, note that after a large enough number of label requests, a random point x_1 in $DIS(V)$ will likely be in $\partial_{\mathcal{D},\mathbb{C}} h^*$. We can exploit this fact by using x_1 to split V into two subsets: $V_{(x_1,+1)}$ and $V_{(x_1,-1)}$. Now, if $x_1 \in \partial_{\mathcal{D},\mathbb{C}} h^*$, then $\inf_{h\in V_{(x_1,+1)}} er(h) = \inf_{h\in V_{(x_1,-1)}} er(h) = 0$. Therefore, for almost every point $x \notin DIS(V_{(x_1,+1)})$, we can infer a label for this point (the one agreed upon by $V_{(x_1,+1)}$), which will agree with the target function, and similarly for almost every point $x \notin DIS(V_{(x_1-1)})$ (using the label agreed upon by $V_{(x_1,-1)}$). Thus, we can focus the region for which we must request labels down to $DIS(V_{(x_1,+1)}) \cap DIS(V_{(x_1,-1)})$, which may be significantly smaller than $DIS(V)$ (the region where disagreement-based algorithms query). Now, if $\mathbb{P}(DIS(V_{(x_1,+1)}) \cap DIS(V_{(x_1,-1)})) \to 0$, the queries will become focused, and we can improve over passive learning in this situation. On the other hand, if the queries do not become focused, then $DIS(V_{(x_1,+1)}) \cap DIS(V_{(x_1,-1)})$ converges to a region $\partial_{\mathcal{D},V_{(x_1,+1)}} h^* \cap \partial_{\mathcal{D},V_{(x_1,-1)}} h^*$, which has nonzero probability. So after a large enough number of label requests, a random x_2 in $DIS(V_{(x_1,+1)}) \cap DIS(V_{(x_1,-1)})$ will likely be in $\partial_{\mathcal{D},V_{(x_1,+1)}} h^* \cap \partial_{\mathcal{D},V_{(x_1,-1)}} h^*$. In this case, we can repeat this argument, splitting V into four sets with 0 infimum error rates, and we can further focus the query region to the intersections of their regions of disagreement, etc. We can repeat this process as needed, until we get a partition of V with a shrinking intersection of regions of disagreement.

Note that this argument can be written more concisely in terms of *shattering*. That is, any $x \in DIS(V)$ is simply one such that V shatters $\{x\}$; a point $x \in DIS(V_{(x_1,+1)}) \cap DIS(V_{(x_1,-1)})$ is simply one such that V shatters $\{x_1, x\}$, and for any $x \notin DIS(V_{(x_1,+1)}) \cap DIS(V_{(x_1,-1)})$, the label y we infer about x has the property that the set $V_{(x,-y)}$ does not shatter $\{x_1\}$, and so on for further iterations. Also, in order to reduce the probability of a mistake due to $x_1 \notin \partial_{\mathcal{D},\mathbb{C}} h^*$ (or similarly for later x_i), we can simply replace each single x_i with multiple samples, and then take a majority vote over whether to infer the label, and which label to infer if we do so. Generally, we can think of this as estimating a certain probability; for brevity, in the algorithm below I omit the details of this estimator, and simply write $\hat{\mathbb{P}}$, which can be instantiated in a variety of ways using only *unlabeled* examples; the particulars of $\hat{\mathbb{P}}$ are not essential to the results (see my doctoral dissertation [5] for one possible instantiation of this estimator). The resulting algorithm, referred to as Meta-Algorithm 1 below, is stated as follows.

Meta-Algorithm 1: $Activizer(\mathcal{A}_p, n)$
Input: passive algorithm $\mathcal{A}_p$, label budget n
Output: classifier $\hat{h}$

0. Request the first $\lfloor n/3 \rfloor$ labels and let Q denote these $\lfloor n/3 \rfloor$ labeled examples
1. Let $V = \{h \in \mathbb{C} : er_Q(h) = 0\}$
2. For $k = 1, 2, \ldots, d+1$
3. $\quad \hat{\Delta}^{(k)} \leftarrow \hat{\mathbb{P}}\left(x : \hat{\mathbb{P}}\left(S \in \mathcal{X}^{k-1} : V \text{ shatters } S \cup \{x\} | V \text{ shatters } S\right) \geq 1/2\right)$
4. $\quad$ Let $\mathcal{U}_k$ denote the next $\lfloor n/(6 \cdot 2^k \hat{\Delta}^{(k)}) \rfloor$ unlabeled examples, and $\mathcal{L}_k \leftarrow \{\}$
5. $\quad$ For each $x \in \mathcal{U}_k$,
6. $\quad\quad$ If $\hat{\mathbb{P}}\left(S \in \mathcal{X}^{k-1} : V \text{ shatters } S \cup \{x\} | V \text{ shatters } S\right) \geq 1/2$
7. $\quad\quad\quad$ Request the label y of x, and let $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup \{(x, y)\}$
8. $\quad\quad$ Else, let $y = \underset{y \in \{-1,+1\}}{\operatorname{argmax}} \hat{\mathbb{P}}\left(S \in \mathcal{X}^{k-1} : V_{(x,-y)} \text{ does not shatter } S | V \text{ shatters } S\right)$
$\quad\quad\quad$ and let $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup \{(x, y)\}$
9. Return $ActiveSelect(\{\mathcal{A}_p(\mathcal{L}_1), \mathcal{A}_p(\mathcal{L}_2), \ldots, \mathcal{A}_p(\mathcal{L}_{d+1})\}, \lfloor n/3 \rfloor)$

Subroutine: $ActiveSelect(\{h_1, h_2, \ldots, h_N\}, m)$
0. For each $j, k \in \{1, 2, \ldots, N\} : j < k$,
1. $\quad$ Take the next $\lfloor m / \binom{N}{2} \rfloor$ examples x s.t. $h_j(x) \neq h_k(x)$ (if such examples exist)
2. $\quad$ Let m_{jk} and m_{kj} respectively denote the number of mistakes h_j and h_k make on these
3. Return $h_{\hat{k}}$, where $\hat{k} = \arg\min_k \max_j m_{kj}$

Theorem 1. *For any VC class $\mathbb{C}$, with a suitable choice for the estimators $\hat{\mathbb{P}}$, Meta-Algorithm 1 is an activizer for $\mathbb{C}$.*

For the proof, see my doctoral dissertation [5]. We immediately get the following simple corollary.

Corollary 1. *For any VC class $\mathbb{C}$, there exists an active learning algorithm that achieves a label complexity Λ_a such that, for all target distributions $\mathcal{D}$ and target functions $h^* \in \mathbb{C}$,*

$$\Lambda_a(\epsilon, h^*, \mathcal{D}) = o(1/\epsilon).$$

Open Problems

One natural question is how large of a gap between passive and active learning we might expect. We know the gap can sometimes be large or small [1], but we are still missing a general characterization of this gap, beyond Theorem 1 and the various analyses of verifiable label complexity [e.g., 3, 4]. Meta-Algorithm 1 has been presented in a minimalistic style; however, several of its steps can be performed in a sequential CAL-like manner, updating the version space after each query, and a detailed analysis of such a variant of Meta-Algorithm 1 may help to quantify this gap. Another interesting open problem is to generalize these results to situations with noisy labels or model misspecification. Some preliminary results along these lines are presented in my dissertation [5], but many questions remain. In particular, I conjecture that for any VC class, it is possible to activize some *empirical risk minimization* passive algorithm. Yet another great open problem is to design activizers guaranteed to preserve certain nice properties of the passive algorithm, such as dimension independence from a margin-based learning bias, or weak dependence on dimensionality from a sparsity-based learning bias.

References

[1] Balcan, M.-F., Hanneke, S., and Wortman, J. (2008). The true sample complexity of active learning. In *Proceedings of the 21st Conference on Learning Theory*.

[2] Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, **15**(2), 201–221.

[3] Dasgupta, S. (2005). Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems 18*.

[4] Hanneke, S. (2009a). Rates of convergence in active learning. *In Submission*.

[5] Hanneke, S. (2009b). *Theoretical Foundations of Active Learning*. Ph.D. thesis, Machine Learning Department, School of Computer Science, Carnegie Mellon University.

[6] Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley & Sons, Inc.