Active Learning with an ERM Oracle

Alina Beygelzimer IBM Research
Daniel Hsu UC San Diego
John Langford Yahoo! Research
Tong Zhang Rutgers University

1 Introduction

An active learning algorithm adaptively chooses which unlabeled data to obtain labels for. The hope is that the active learner can query for labels on just a small fraction of the data set, and otherwise perform as well as a fully supervised learning method. In this work, we are interested in agnostic active learning algorithms for binary classification that are provably consistent.

One technique that has proved theoretically profitable is to maintain a candidate set of hypotheses (sometimes called a version space), and to query the label of a point if and only if there is disagreement within this set about how to label the point. The criteria for membership in this candidate set need to be carefully defined so that the optimal hypothesis h^* is always included, but otherwise this set can be quickly whittled down as more labels are queried. This technique is rather straightforward in the realizable setting [3], and it can be extended to the agnostic setting with the use of confidence bounds and hard example constraints [1, 4]. One of the algorithms in [2] uses large importance weights to essentially achieve the same effect.

A drawback of this version space approach is its computational intractability. Maintaining a version space and guaranteeing that *only* valid hypotheses are returned is difficult for linear predictors and appears intractable for interesting nonlinear predictors. Here, we develop a new strategy which is computationally tractable given an oracle that returns an empirical error minimizing hypothesis. As this oracle matches our abstraction of many supervised learning algorithms, we believe active learning algorithms built using this oracle are immediately and widely applicable.

This approach also has a substantial secondary advantage. Version space algorithms must be very conservative in defining the version space, because any error could result in discarding the optimal hypothesis h^*. This conservative behavior implies that in many practical applications, version space algorithms fail to improve on the label complexity of supervised learning. We expect an active learning algorithm based on an empirical error minimization oracle to be substantially more robust (and hence, aggressive) in practice as errors can be recovered from.

In this work, we present an algorithm that avoids the rigidity of hard example constraints and the like. It is similar to the algorithm of [4] in that it only accesses hypotheses via a supervised learning oracle. However, the oracle required is simpler than that used in [4] and also avoids strict adherence to a candidate set of hypotheses. We prove that the algorithm is consistent and has a label complexity analysis comparable to that of previous consistent active learning algorithms.

2 Preliminaries

Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{\pm 1\}$, where $\mathcal{X}$ is the input space and the labels are ± 1. An active learning algorithm receives a stream of examples (x, y) drawn i.i.d. from $\mathcal{D}$ with y hidden unless it is explicitly queried. Let H be a set of hypotheses mapping from $\mathcal{X}$ to $\{\pm 1\}$. For simplicity, we assume H is finite; our results can be extended to infinite H with finite VC dimension. Denote the (true) error of a hypothesis h by $\mathrm{err}(h) := \Pr_{(x,y)\sim\mathcal{D}}[h(x) \neq y]$; and denote the empirical error of h on a sample $Z \subset \mathcal{X} \times \{\pm 1\}$ by $\mathrm{err}(h, Z) := (1/|Z|) \sum_{(x,y)\in Z} I[h(x) \neq y]$, where $I[\cdot]$ is the 0-1 indicator function. Let $h^* := \arg\min\{\mathrm{err}(h) : h \in H\}$ be a hypothesis of minimum error in H.

3 Algorithm

Our algorithm is as follows. Initially, $\tilde{Z}_0 := \emptyset$.
For $t = 1, 2, \ldots$:

1. Get x_t, sampled from the distribution $\mathcal{D}$ (with label y_t hidden).
2. Let $h_t := \arg\min\{\mathrm{err}(h, \tilde{Z}_{t-1}) : h \in H\}$
   and $h'_t := \arg\min\{\mathrm{err}(h, \tilde{Z}_{t-1}) : h \in H \ \wedge \ h(x_t) \neq h_t(x_t)\}$.
3. If $\mathrm{err}(h'_t, \tilde{Z}_{t-1}) - \mathrm{err}(h_t, \tilde{Z}_{t-1}) > \Delta_{t-1}$
   (a) Then: Assign $\tilde{y}_t := h_t(x_t)$, and let $\tilde{Z}_t := \tilde{Z}_{t-1} \cup \{(x_t, \tilde{y}_t)\}$.
   (b) Else: Query true label y_t, and let $\tilde{Z}_t := \tilde{Z}_{t-1} \cup \{(x_t, y_t)\}$.

Note that h_t minimizes the empirical error on $\tilde{Z}_{t-1}$ (the previous $t-1$ examples), and h'_t does the same except over just those hypotheses that disagree with h_t on x_t. The decision to query the t-th label or not is based on the outcome of a simple test involving the error difference $d_t := \mathrm{err}(h'_t, \tilde{Z}_{t-1}) - \mathrm{err}(h_t, \tilde{Z}_{t-1})$ and a threshold Δ_{t-1}. The threshold, specified in the next section, is based on deviation bounds for i.i.d. samples.

Here we give some intuition behind the above algorithm. First, because h'_t minimizes the empirical error on $\tilde{Z}_{t-1}$ among all hypotheses that disagree with h_t on x_t, a sufficiently large error difference d_t implies that any hypothesis disagreeing with h_t on x_t must have larger true error than h_t. The optimal hypothesis h^* cannot have larger true error than h_t, so it must agree with h_t on x_t. In this way, we can deduce $\tilde{y}_t = h_t(x_t) = h^*(x_t)$ whenever d_t is large. Second, substituting any true labels y_t with the labels assigned by $h_t(x_t)$ only works to bias a learner towards h^*, since hypotheses that disagree with h^* on these examples are penalized by the substitution. Finally, if a hypothesis h' disagrees with h^* on x_t, and d_t is large enough, then h' will never minimize the error on $\tilde{Z}_\tau$ at any time $\tau > t$ in the future. Therefore we are sure that h_t always agrees with h^* on examples with synthesized labels, which is essential to avoid using an explicit candidate set of hypotheses (*c.f.* [4]).

4 Analysis (sketch)

Let $\tilde{S}_n \subseteq \tilde{Z}_n$ denote the set of examples $(x_t, \tilde{y}_t)$ for which the algorithm synthesizes the label $\tilde{y}_t$, and let $Q_n := \tilde{Z}_n \setminus \tilde{S}_n$ be the remaining examples (x_t, y_t) for which the algorithm queries the label y_t. Also, let $S_n := \{(x_t, y_t) : (x_t, \tilde{y}_t) \in \tilde{S}_n\}$ be the same as $\tilde{S}_n$ except with the synthesized labels replaced with the true labels. Note that $Z_n := S_n \cup Q_n$ is an i.i.d. sample from $\mathcal{D}$ while $\tilde{Z}_n$ is generally not.

As mentioned before, the threshold Δ_n will be based on (non-uniform) deviation bounds $\beta_n : H \to \mathbb{R}_+$ that guarantee, with probability at least $1 - \delta$,

$$|(\mathrm{err}(h) - \mathrm{err}(h^*)) - (\mathrm{err}(h, Z_n) - \mathrm{err}(h^*, Z_n))| \leq \beta_n(h)$$

for all $h \in H$ and all $n \geq 1$. Using non-uniform bounds allows for tighter guarantees when h is close to h^*, which is crucial to obtain our label complexity bounds. Applying Chernoff bounds to the error differences $\mathrm{err}(h, Z_n) - \mathrm{err}(h^*, Z_n)$, we can set

$$\beta_n(h) := O\left(\alpha_n + \sqrt{\alpha_n \cdot \frac{1}{n}\sum_{t=1}^{n} I[h(x_t) \neq h^*(x_t)]}\right)$$

where $\alpha_n := \log(|H|n/\delta)/n$. Note that $\beta_n(h)$ interpolates between $O(1/n)$, when $h = h^*$, and $O(1/\sqrt{n})$, when h and h^* disagree on all examples.

Let $H_n := \{h \in H : h(x) = h^*(x) \ \forall (x, \tilde{y}) \in \tilde{S}_n\}$. This serves as a candidate set for the sake of analysis, but note that the algorithm never explicitly restricts its attention to H_n. We consider H_n in the analysis because the $h \in H_n$ enjoy a tighter deviation bound. Indeed, let $\bar{\beta}_n := \max\{\beta_n(h) : h \in H_n\}$; we will show that $\bar{\beta}_n$ is roughly bounded by $\alpha_n + \sqrt{\alpha_n |Q_n|/n}$, so it suffices to define the threshold as $\Delta_n := 5\bar{\beta}_n \approx 5(\alpha_n + \sqrt{\alpha_n |Q_n|/n})$.

The first lemma makes clear the intuition that replacing labels with those assigned by h^* favorably biases us towards h^*.

Lemma 1. *Assume* $\tilde{y}_t = h^*(x_t)$ *for all* $(x_t, \tilde{y}_t) \in \tilde{S}_n$. *Then* $\mathrm{err}(h, \tilde{Z}_n) - \mathrm{err}(h^*, \tilde{Z}_n) \geq \mathrm{err}(h, Z_n) - \mathrm{err}(h^*, Z_n)$ *for all* $h \in H$.

This bias essentially lets us think of $\tilde{Z}_n$ as an i.i.d. sample in the subsequent analysis.

The next lemma captures several intuitions: when the error difference $\mathrm{err}(h'_n, \tilde{Z}_{n-1}) - \mathrm{err}(h_n, \tilde{Z}_{n-1})$ is large, then the algorithm correctly determines $h_n(x_n) = h^*(x_n)$ (so the previous lemma applies); and h_n always agrees with h^* where labels are synthesized on previous examples.

Lemma 2. *With probability at least* $1 - \delta$,

1. *For* $n \geq 1$: $(x_n, \tilde{y}_n) \in \tilde{S}_n \Rightarrow h_n(x_n) = h^*(x_n)$;
2. *For* $n \geq 0$: $h' \notin H_n \Rightarrow \mathrm{err}(h') - \mathrm{err}(h^*) > 3\beta_n(h')$;
3. *For* $n \geq 0$: $h_{n+1} \in H_n$.

Some consequences of this lemma are: (1) $\bar{\beta}_n \leq O(\alpha_n + \sqrt{\alpha_n |Q_n|/n})$, which justifies the setting of the threshold Δ_n, and (2) $\mathrm{err}(h_n) \leq \mathrm{err}(h^*) + \beta_{n-1}(h_n)$. The latter in turn implies the following consistency guarantee.

Theorem 1. *For any* T, H, *and* δ, *with probability at least* $1 - \delta$,

$$\mathrm{err}(h_{T+1}) \leq \mathrm{err}(h^*) + O\left(\frac{\log(T|H|/\delta)}{T} + \sqrt{\mathrm{err}(h^*) \cdot \frac{\log(T|H|/\delta)}{T}}\right).$$

In other words, the label complexity of the active learning algorithm is no worse than the sample complexity of supervised learning.

Finally, we give a bound on the label complexity of the algorithm in terms of the *disagreement coefficient* [5], defined as follows. Let $d(h, h') = \Pr_{(x,y)\sim\mathcal{D}}[h(x) \neq h'(x)]$ and $B(h, r) = \{h' \in H : d(h, h') \leq r\}$. Let $\mathrm{Disagree}(r) = \{x : \exists h \in B(h^*, r) \text{ s.t. } h(x) \neq h^*(x)\}$. The disagreement coefficient is

$$\theta := \sup\left\{\frac{\Pr_{(x,y)\sim\mathcal{D}}[x \in \mathrm{Disagree}(r)]}{r} : r > 0\right\}.$$

Theorem 2. *For any* T, H, *and* δ, *with probability at least* $1 - \delta$, *the expected number of labels queried by the algorithm after seeing* T *unlabeled examples is bounded as*

$$\mathbb{E}|Q_T| \leq O\left(\theta \cdot \mathrm{err}(h^*) \cdot T + \theta^2 \cdot \log(|H|/\delta) \cdot \log^3 T\right).$$

In comparison, the number of labels requested by the algorithms of [1] and [4] is $O(\theta \cdot \mathrm{err}(h^*) \cdot T + \theta \cdot \log(|H|/\delta) \cdot \log^2 T)$. So, we are worse by a factor of $\theta \cdot \log T$ in the second term. Note that the factor $\mathrm{err}(h^*) \cdot T$ in the first term is generally unavoidable due to lower bounds for any active learner [2]. However, in some cases, the quantity

$$\kappa := \sup\left\{\frac{d(h, h^*)}{\mathrm{err}(h) - \mathrm{err}(h^*)} : h \in H\right\}$$

may be bounded, in which case we can provide an improved guarantee of

$$\mathbb{E}|Q_T| \leq O\left(\kappa \cdot \theta^2 \cdot \log(|H|/\delta) \cdot \log^3 T\right).$$

So, when the leading terms are small, the active learner is exponentially more efficient than a supervised learner that requests all T labels.

References

[1] M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *ICML*, 2006.

[2] A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *ICML*, 2009.

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

[4] S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *NIPS*, 2007.

[5] S. Hanneke. A bound on the label complexity of agnostic active learning. In *ICML*, 2007.