Understanding the Effects of Batching in Online Active Learning

Kareem Amin
Google, New York

Corinna Cortes
Google, New York

Giulia DeSalvo
Google, New York

Afshin Rostamizadeh
Google, New York

Abstract

Online active learning (AL) algorithms often assume immediate access to a label once a query has been made. However, due to practical constraints, the labels of these queried examples are generally only available in "batches". In this work, we present an analysis for a generic class of batch online AL algorithms, which reveals that the effects of batching are in fact mild and only result in an additional label complexity term that is quasi-linear in the batch size. To our knowledge, this provides the first theoretical justification for such algorithms and we show how they can be applied to batch variants of three canonical online AL algorithms: IWAL, ORIWAL, and DHM. Finally, we also present empirical results across several benchmark datasets that corroborate these theoretical insights.

1 Introduction

Large labeled datasets are often used to train models in supervised learning. However, in some domains, such as those that require domain experts, labeling is a very costly. Active learning directly tackles the important task of training accurate models while at the same time minimizing the number of labeled points.

Previous work in active learning often analyzes the online, or streaming, setting where a learner observes a single unlabeled example at a time and decides whether or not to request the label of the example, receiving the label immediately if queried, and typically updating the learner with the additional label before receiving the next sample point. Generally, obtaining the label of just a single point is entirely impractical due to, for example, the overhead of assembling a pool of qualified raters and assigning enough work to each rater in order for their time to be well spent. Additionally, there is a considerable overhead in updating an active learner, e.g. computing a new version space, with one additional instance at a time. Thus, in practice, labels are requested only once a large enough batch of requests has been queued. As an example, consider remote sensors with a finite buffer that process a stream of unlabeled data, a subset of which may be useful for training a machine learning model. Due to the practical reasons discussed above, as well as potential communication costs, the sensor only sends batches of points to a labeling service once the buffer is full.

Motivated by these practical constraints, we analyze the *batch online active learning* setting. A common approach is to convert off-the-shelf online active learning algorithms to operate in the batch setting. This can be accomplished by delaying label feedback to the algorithm until a sufficiently large number of label requests are made. However, the effect of batching on the active learning algorithm's generalization and label complexity guarantees is not well understood. Since the batch framework is a substantially more restrictive setting, the main questions we seek to answer are: in what ways will batching impact known theoretical guarantees? How strongly do label complexity and generalization guarantees depend on the batch size?

To that end, we present a label complexity analysis, that is a bound on the number of requested labels, for a generic batch online active meta-algorithm that is assumed to satisfy a mild *time-decreasing labeling rate* condition. This condition states that the probability of requesting the label of point decreases as a function of time, which is a natural property for any active learning algorithm that admits non-trivial bounds on label complexity. Crucially, our theoretical analysis shows that the label complexity of such batch active learning algorithms, ignoring logarithmic terms, admit a linear dependence on the batch size. This reveals that the effects of batching are minimal as long as the batch size is a constant independent of the total number of observations. We show that this result can be applied to batch variants of three well studied online

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

active learning algorithms: IWAL, ORIWAL, and DHM [Beygelzimer et al., 2009, Cortes et al., 2019a, Dasgupta et al., 2008]. Moreover, we prove that the theoretical guarantees of these algorithms are not affected by batch size and empirically verify the insights provided by both the label complexity and generalization bounds. To our knowledge, this is the first work proving theoretical guarantees for batch online active learning.

Below, we review related work. In Section 2, we present our generic batch online active learning framework and in Section 3, we provide a novel theoretical analysis of its label complexity. Then, we show applications of this generic batch online AL framework along with the derivations of the generalization guarantees in Sections 4 and an empirical verification in Section 5. Most of the proofs of our analysis are found in the appendix.

Related Work: Most theoretical work in (non-batch) active learning considers the online setting with a focus on proving generalization guarantees for the hypothesis returned by an active learning algorithm and bounds on the active learner's label complexity. In the case of separable data, Cohn et al. [1994] derived an algorithm that exhibits an exponential decrease in label complexity when compared to passive learning. The main idea of this algorithm is to trim the hypothesis set of all classifiers that are inconsistent with the currently labeled data and to only ask for the labels of points the hypotheses in this set disagree on. The amount of disagreement among a set of hypothesis can be characterized by the *disagreement coefficient*, which was first introduced by Hanneke [2007]. These ideas on disagreement are the core of many active learning algorithms and the label complexity guarantees of these algorithms are often in terms of the disagreement coefficient [Balcan et al., 2006, Dasgupta et al., 2008, Beygelzimer et al., 2009, 2010, Cortes et al., 2019a,b]. Similar quantities will appear in the bounds we present, although we arrive at them in a significantly different fashion. Another line of work has focused on algorithms based on requesting labels along the margin of a linear separator, which only under certain distributional assumptions admit theoretical guarantees [Dasgupta et al., 2005, Balcan et al., 2007, Balcan and Long, 2013, Awasthi et al., 2014, 2015, Zhang, 2018].

Active learning has also been analyzed in pool-based setting where the entire pool of unlabeled data is available and the algorithm must choose subsets of this pool to be sent to raters for labeling. Unlike the batch online active learning setting, the algorithms in the pool based setting do not have limited memory and thus are qualitatively very different than the algorithms analyzed in this paper. Several authors have studied the pool-based setting, but the focus was primarily on finding solutions for specific tasks. For example, Kurihara and Sugiyama [2012] derives sampling objectives tailored to linear regression for choosing a single batch of examples to be labeled, Bach [2007] presents an asymptotic analysis for misspecified generalized linear models, and McCallum and Nigam [1998], Hoi et al. [2006a,b, 2008] focus on text and image classification tasks. Other work has focused on incorporating different definitions of diversity and informativeness, but without deriving any generalization and label complexity guarantees [Brinker, 2003, Xu et al., 2007, Guo and Schuurmans, 2008]. Dasgupta and Hsu [2008] develops an active learning algorithm with theoretical guarantees under specific assumptions on the ability to cluster the data.

Chen and Krause [2013] analyze the batch active learning problem, albeit in the pool based setting, and show that a greedy batch construction strategy is competitive with an optimal batch selection when the problem exhibits an adaptive submodularity condition. Our work is significantly different, in that we consider the online setting and make no submodularity assumption.

Online learning with delayed feedback has been studied in the general partial-monitoring setting. Joulani et al. [2013] demonstrate that in non-adversarial settings the price of delayed feedback is an additive regret term that is linear (ignoring log factors) in the length of the feedback delay. This strongly mirrors the label-complexity result we achieve in our setting as we bound the number of additional label requests made by an amount that is also linear (ignoring log factors) in the length of feedback delay. However, we emphasize that one setting does not subsume the other. In online learning with delayed feedback, the learner eventually receives feedback for every decision, while in active learning, there are some rounds where the learner never receives feedback at all (i.e., when a label is not requested). Moreover, in contrast to online learning, where there is a single objective (minimizing regret), active learning admits a bicriterion of bounding label complexity and generalization error.

2 Batch Active Learning Framework

Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ denote an example space and $\mathcal{Z}_\perp = \mathcal{X} \times (\mathcal{Y} \cup \{\perp\})$ denote the same example space except where examples can be label-free (denoted by $\perp$). We assume that the data is drawn stochastically, that is the data points are drawn i.i.d. from an unknown distribution $\mathcal{D}$ over $\mathcal{Z}$ and define a hypothesis set H where each function $h \in H$ maps from $\mathcal{X}$ to $\mathcal{W} \subseteq \mathbb{R}$. The quality of a hypothesis function is measured by a loss function, $\ell \colon \mathcal{W} \times \mathcal{Y} \to [0, 1]$, and we denote by $L(h) = \mathbb{E}[\ell(h(x), y)]$ the expected loss of hypothesis h.

Let A denote an online active learning algorithm. At each time step $t \in [T] := \{1, \ldots, T\}$, algorithm A

maintains an internal state $\omega_t \in \Omega$, where Ω denotes a universe of possible internal states. Given an example $x_t \in \mathcal{X}$, the algorithm first decides whether or not to receive the true label of x_t. We denote by $\bar{y}_t \in \mathcal{Y} \cup \{\perp\}$ either the true revealed label y_t or the decision not to request the label, $\perp$, at time t. Given this new information, the algorithm then updates its state to ω_{t+1}. Both the decision of requesting the label and the update of its state may be a probabilistic process.

An online active learning algorithm can be fully characterized by its state, a function that decides whether to request for a label, and a function that updates the state of the algorithm. More formally, we define two functions $\texttt{Labeler}\colon \Omega \times \mathcal{X} \to [0,1]$ which maps a state ω_t and an example x_t to the probability of requesting a label and $\texttt{Updater}\colon \Omega \times \mathcal{Z}_\perp \to \Delta(\Omega)$ which maps the feedback received during a timestep to a distribution over next states, where $\Delta(\Omega)$ denotes the probability simplex over Ω. Given an initial state ω_1, an online active learning algorithm is then defined by the following triplet: $A = (\omega_1, \texttt{Labeler}, \texttt{Updater})$.

In order to convert such an online active learning algorithm A into an algorithm A_B for the batch setting, we "freeze" the state of A until B labels have been requested. That is, the algorithm makes decisions on a sequence of points without updating its state until B labels have been requested. We call a sequence of timesteps where the state remains unchanged a *round*. At the end of the round, the algorithm receives the B labels of the requested points and it updates its state accordingly.[1] This continues until the time horizon T is met, at which point the algorithm selects a hypothesis using information from the final state. Note that in the sequel, during the execution of the algorithm, we denote the current round index using the variable r. Algorithm 1 defines the batch algorithm in detail.

In Section 3 we will provide a sufficient condition in order to analyze the label complexity of Algorithm 1. Then in Section 4, we argue that this condition holds for batch variants of many existing online active learning algorithms and therefore our label complexity bounds can be applied directly. In Section 4, we also show that for these same algorithms, generalization guarantees are completely unaffected by batching.

Algorithm 1 Batch Online AL Algorithm A_B.

Inputs : $A = (\omega_1, \texttt{Labeler}, \texttt{Updater})$, batch size $B \geq 1$, time horizon $T \geq 1$.
Set : current round $r = 1$, number labels requested in round $C_1 = 0$, previous round boundary $\tau_0 = 0$.
for $t = 1, 2, \ldots, T$ **do**
  Receive x_t.
  Draw $Q_t \sim \text{Bernoulli}(\texttt{Labeler}(\omega_r, x_t))$.
  Update $C_r = C_r + Q_t$.
  #B requests or final timestep, end round.
  if $C_r = B$ **or** $t = T$ **then**
    #Receive batched labels.
    for $t' = \tau_{r-1} + 1, \ldots, t$ **do**
      if $Q_{t'} = 1$ **then**
        Receive $y_{t'}$.
        Set $\hat{y}_{t'} = y_{t'}$.
      else
        Set $\hat{y}_{t'} = \perp$.
    #Perform batched state updates.
    Initialize $\omega' = \omega_r$.
    for $t' = \tau_{r-1} + 1, \ldots, t$ **do**
      Draw $\omega' \sim \texttt{Updater}(\omega', x_{t'}, \hat{y}_{t'})$
    Update : round $r = r + 1$, previous round boundary $\tau_{r-1} = t$, frozen state $\omega_r = \omega'$.
    Reset label count $C_r = 0$.
return $\widehat{h}_T$ hypothesis learned with state ω_r.

3 Label Complexity

In this section, we show that converting an online active algorithm to a batch online active algorithm results in a label complexity bound that contains only a mild dependence on batch size. More specifically, the additional label complexity cost over the online active learning algorithm is an *additive* $\tilde{O}(B)$ term in the batch size B. Ignoring log factors, this is the best one can hope for, since a batch active algorithm must request $\Omega(B)$ labels in order to receive any labels at all. In order for this general result to hold, the active learning algorithm must satisfy a natural condition, which we called *time-decreasing labeling rate*.

3.1 Time-Decreasing Labeling Rate

Let $r(t)$ be a random variable denoting the round corresponding to timestep t, that is $r(t) = \min\{s \mid \tau_s \geq t\}$, and let $\mathcal{F}_t$ denote the sigma algebra containing all random variables up to time t. A batch active learning algorithm has time-decreasing labeling rate if the probability of requesting a label can be upper bounded by a decreasing function p^+_δ that depends only on the timesteps required for the previous round to elapse, $\tau_{r(t)-1}$ and a failure parameter δ. Note that $\tau_{r(t)-1}$ is known at time $t-1$, meaning it is $\mathcal{F}_{t-1}$-measurable (see the appendix for proof of this statement).

Definition 1 (Time-Decreasing Labeling Rate) *For any $\delta > 0$, we say A_B has time-decreasing labeling rate if there exists a non-negative strictly decreasing function $p^+_\delta(t)\colon \mathbb{N} \to [0,1]$ such that for all $t \geq 1$ with*

[1] The only exception is the last round, which allows for a batch of fewer than B labels. One could instead ignore the requests made in the last round. This does not materially change the results, but would complicate the presentation.

probability $1-\delta$,

$$\mathbb{P}(Q_t = 1 \mid \mathcal{F}_{t-1}) \leq p^+_\delta(\tau_{r(t)-1}).$$

In the following, we will elide the dependence on δ and write p^+, unless we wish to make the dependence explicit. The requesting probability of the active learning algorithm could be non-monotonic at each iteration, but the condition implies there is a monotonic decreasing upper bound, which is natural for any active learning algorithm with nontrivial label complexity.

3.2 Theoretical Analysis

In this section, we conduct a label complexity analysis for algorithm A_B that admits the time-decreasing labeling rate property and which was derived from an arbitrary online active algorithm A via Algorithm 1. As we will see, the label complexity analysis of batch algorithms departs significantly from the standard analysis for online active learning algorithms since the length of a round is itself a random variable with a particularly intricate dependence on all previous rounds.

Given a horizon $T \geq 1$, the number of labels requested by algorithm A_B will be bounded by $r(T)B$ since A_B requests exactly B labels every time a round elapses and the algorithms halts after round $r(T)$ (except the last round, which may request fewer than B labels). Thus, the goal of this analysis is to prove a high probability upper bound on $r(T)$.

Understanding $r(T)$, depends on analyzing the boundaries $\tau_0, \tau_1, \tau_2, \ldots$ of the subsequent rounds. If this sequence grows quickly, it takes relatively few rounds (and therefore few labels) to reach T timesteps. We henceforth focus on the length-of-round, $W_r = \tau_r - \tau_{r-1}$, where W_r is a random variable denoting the waiting time for the rth round to elapse. The crux of the proof focuses on showing a lower bound on W_r that holds with high probability. Equivalently, we argue that the lengths of rounds become increasingly longer.

Concretely, the argument proceeds in three stages. We first relate the length-of-round, W_r, to a simpler conservative random process, $\tilde{W}_r$, that can be directly defined as a sequence of dependent negative binomial variables. The process is conservative in the sense that the round lengths will, with high probability, be smaller compared to the W_r necessitating additional rounds and therefore more labels before seeing T examples. We then relate this conservative process to an idealized deterministic sequence, which loosely corresponds to the mean of the stochastic conservative process. Finally, we analyze the behavior of this deterministic sequence. At a high level, this analysis reduces the problem of lower bounding W_r to the problem of lower bounding the growth of a deterministic process.

Step 1: Relating W_r to conservative process $\tilde{W}_r$
First recall N/q is the expected value of a negative binomial distribution $\mathrm{NB}(q, N)$ which counts the number of independent Bernoulli trials with success parameter p until exactly B successes occur. Now, note that for any round r, if we condition on the value of τ_{r-1}, then Definition 1 implies with high probability that for any t occurring in round r, $\mathbb{P}(Q_t = 1 \mid \tau_{r-1}) \leq p^+_\delta(\tau_{r-1})$. Since the length of the round W_r is equal to the number of Bernoulli trials (with success parameter at most $p^+_\delta(\tau_{r-1})$) before B labels are queried, its conditional expectation is lower bounded as follows $\mathbb{E}[W_r \mid \tau_{r-1}] \geq B/p^+_\delta(\tau_{r-1})$.

We construct a new process $\tilde{W}_r$ defined explicitly in terms of a negative binomial distribution with parameter $p^+_\delta(\tilde{\tau}_r)$, specifically

$$\tilde{W}_1 = B \qquad \tilde{\tau}_r = \sum_{s=1}^{r} \tilde{W}_s \qquad \tilde{W}_{r+1} \sim \mathrm{NB}(p^+_\delta(\tilde{\tau}_r), B)\,,$$

and relate it to W_r. The subsequent lemma proves that the conservative process $\tilde{W}_r$ is upper bounded by the length-of-round W_r.

Lemma 1 *Fix $\delta > 0$, and suppose that A_B has time-decreasing labeling rate. Then with probability at least $1-\delta$, for all r, it holds that $\tilde{W}_r \leq W_r$, where $\tilde{W}_r$ has a dependence on δ via the parameter $p^+_\delta(\tilde{\tau}_r)$.*

The lemma is proven via a coupling argument along with the fact that A_B has time-decreasing labeling rate.

Step 2: Relating $\tilde{W}_r$ to deterministic seq. w_r

Although the distribution of $\tilde{W}_r$ conditioned on a fixed $\tilde{\tau}_{r-1}$ can be directly related to a negative binomial distribution, the unconditioned distribution does not have this direct relationship. To help with this, we consider the trajectory of the process $\tilde{W}_1, \tilde{W}_2, \ldots$ that occurs if every variable $\tilde{W}_r$ is equal to its mean conditioned on the past. In particular define the following deterministic sequence,

$$w_1 = \hat{B} \qquad w_1^r = \sum_{s=1}^{r} w_s \qquad w_{r+1} = \frac{\hat{B}}{p^+(w_1^r)}\,, \quad (1)$$

then letting $\hat{B} = B$ recovers the trajectory just described. We stress that even the expectation of the stochastic process $\tilde{W}_r$ does not follow the particular deterministic trajectory of w_r, i.e. $\mathbb{E}[\tilde{W}_r] \neq w_r$ in general.[2] However, if we let $\hat{B} = B/4$, we can show that with high probability the deterministic process w_r lower bounds the stochastic process $\tilde{W}_r$.

First, we define a collection of independent non-identical negative binomial random variables

[2] In particular, even after two rounds, we have $\mathbb{E}[\tilde{W}_3] = \mathbb{E}[\mathbb{E}[\tilde{W}_3 \mid \tilde{\tau}_2]] \neq \mathbb{E}[\tilde{W}_3 \mid \tilde{\tau}_2 = w_1 + w_2] = w_3$.

$N(1), \ldots, N(T)$, where $N(t) \sim \text{NB}(p^+(t), B)$ and where $\mu(t) = \mathbb{E}[N(t)] = B/p^+(t)$ is the expected value of $N(t)$ and also define $\tilde{r}(T) = \min\{r \mid \tilde{\tau}_r \geq T\}$. Then by definition we have $\tilde{W}_1 = B$, and $\tilde{W}_r = N(\tilde{\tau}_{r-1})$ for up to round $\tilde{r}(T)$.

Given that $\{\tilde{W}_r\}$ can be defined in terms of this collection of independent negative binomials $N(\cdot)$, we argue that the growth of $\tilde{W}_r$ is well-behaved as long as $N(\cdot)$ is well-behaved, specifically each $N(t)$ is not much smaller than its mean. To this end, let T_{bad} count the number of negative binomials that are significantly smaller than their means, that is, $T_{\text{bad}} = \sum_{t=1}^{T} \mathbf{1}[N(t) < \frac{1}{4}\mu(t)]$. Moreover, consider the deterministic sequence w_r defined in equation (1) with $\hat{B} = B/4$. As long as the process $N(\cdot)$ is well-behaved, $\tilde{W}_r$ grows faster than w_r. The next lemma states for a horizon R, in the worst case, $\tilde{W}_r$ grows like w_r for the first $R - T_{\text{bad}}$ rounds, and then like B for the final T_{bad} rounds (since the outcome of any $N(\cdot)$ is at least B).

Lemma 2 *Fix a horizon $T \geq 1$. Let w_r be the sequence defined in equation 1 with $\hat{B} = B/4$. On any outcome of $N(1), \ldots, N(T)$, and any R satisfying $T_{\text{bad}} \leq R \leq \tilde{r}(T)$, $w_1^{R-T_{\text{bad}}} + T_{\text{bad}}B \leq \tilde{\tau}_R$, which implies $w_1^{R-T_{\text{bad}}} \leq \tilde{\tau}_R$.*

Next, we bound T_{bad} with high probability. Specifically, we apply a Chernoff argument to bound the probability than an individual $N(t)$ takes a value of less than $1/4$ of its mean and then use Bernstein's inequality to bound the number of times that this can occur.

Lemma 3 *For any $\delta < \sqrt{1/e}$, and $B \geq 2\log(T)$, it follows that $P(T_{\text{bad}} > 1 + 3\log(1/\delta)) < \delta$.*

We now state a main theorem, which relates the total number of labels requested by a batch active learner A_B with time decreasing labeling rate p^+ to the deterministic process w_r generated by p^+. In particular, we relate the label complexity to R^*, the number of rounds sufficient for the deterministic process to satisfy $w_1^{R^*} \geq T$, which we analyze in the final step.

Theorem 1 *Fix $\delta > 0$, and time horizon $T \geq 1$. Let A_B be a batch active sampling algorithm with time decreasing labeling rate p^+, and batch size $B \geq 2\log(T)$. Let w_r be the deterministic sequence defined in equation 1 with $\hat{B} = B/4$. Let R^* be a number large enough such that $w_1^{R^*} \geq T$, then with probability at least $1 - 2\delta$, the total labels requested by A_B is bounded by:*

$$Br(T) \leq BR^* + 3B\log(1/\delta) + 2B.$$

Proof. We want to show an upper bound on $r(T)$ that depends on R^* and that holds with high probability. We first relate $\tilde{r}(T)$ to $r(T)$ by considering the event $\mathcal{E}'$ that $\tilde{W}_r \leq W_r$ for all r. This event implies that $\tilde{\tau}_r \leq \tau_r$ since $\tilde{\tau}_r$ and τ_r equal the sum of the length-of-rounds $\tilde{W}_{r'}$ and $W_{r'}$ for $r' \in [r]$, respectively. This in turn implies that $r(T) \leq \tilde{r}(T)$ by definition. Next, we focus on proving an upper bound on $\tilde{r}(T)$.

Define the event $\mathcal{E}$ as $T_{\text{bad}} \leq Z$ and consider outcomes where it holds. For the sake of contradiction, suppose that $R^* + Z < \tilde{r}(T) - 1$ where $Z = 1 + 3\log(1/\delta)$. By definition of R^*, we have $T \leq w_1^{R^*}$. Combining this with the fact that w_1^r is monotonic and that $T_{\text{bad}} \leq Z$, it then follows that $T \leq w_1^{R^*} \leq w_1^{R^*+(Z-T_{\text{bad}})}$ (call this Fact-1). Again by the event $\mathcal{E}$ and the contradiction assumption, it holds $T_{\text{bad}} \leq Z \leq R^* + Z < \tilde{r}(T) - 1 \leq \tilde{r}(T)$. Then, we can apply Lemma 2 by taking $r = R^* + Z$ to conclude that $w_1^{R^*+Z-T_{\text{bad}}} \leq \tilde{\tau}_{R^*+Z}$ (call this Fact-2). Combining Fact-1, Fact-2 and the contradiction assumption, respectively, we have $T \leq w_1^{R^*+Z-T_{\text{bad}}} \leq \tilde{\tau}_{R^*+Z} \leq \tau_{\tilde{r}(T)-1}$. The inequality $T \leq \tau_{\tilde{r}(T)-1}$ contradicts the definition of $\tilde{r}(T) = \min\{r \mid \tilde{\tau}_r \geq T\}$ and thus, on the event $\mathcal{E}$, it holds that $\tilde{r}(T) \leq R^* + Z + 1$.

Taking a union bound and using Lemmas 1 and 3, with probability at least $1 - 2\delta$, both $\mathcal{E}$ and $\mathcal{E}'$ hold, and therefore $r(T) \leq \tilde{r}(T) \leq R^* + Z + 1 = R^* + 3\log(1/\delta) + 2$. Observing that the label complexity of A_B is at most $Br(T)$ completes the proof. □

Next, we show that BR^* is on the same order as standard active learning bounds; therefore, the above theorem shows that the effects of batching only costs an additional $3B\log(1/\delta) + 2B$ in label complexity.

Step 3: Behavior of deterministic seq. w_r

First, let us recall the label complexity bounds of standard online active learning algorithms, which can be written in the form of $a^*T + f(T)$. Here, $f(T)/T$ is $o(1)$ and $a^* \in [0, 1]$ is a problem-specific constant that typically contains quantities such as the disagreement coefficient and/or the loss L^* of the best hypothesis. Most active learning algorithms admit a time-decreasing labeling rate either of $p^+(t) = O(1/\sqrt{T})$ or $p^+(t) = O(1/T)$ resulting in label complexity bounds of $a^*T + O(\sqrt{T})$ and $a^*T + O(\log T)$, respectively. In our analysis, we study these two labeling rates and show that the cost of batching on labeling is at most a single additive $\tilde{O}(B)$ on top of the standard rate.

Specifically, we consider $p^+(t) = a + bt^{-\alpha}$, where $\alpha \in \{1/2, 1\}$, $a \in [0, 1]$ and $b \geq 0$. Returning to the deterministic sequence defined in equation (1), recall that w_1 begins at B and then converges to B/a as $t \to \infty$. In the appendix, we give a general theorem for studying deterministic sequences that asymptote, but exhibit non-trivial growth before convergence. Utilizing this theorem, we can bound the number of rounds R^* before $w_1^{R^*} \geq T$, thus allowing us to apply Theorem

1 when the functional form of p^+ is known.

Theorem 2 *Fix $\delta < \sqrt{1/e}$, time horizon $T \geq 1$, and $B > \max\{16b, 16, 2\log(T)\}$. Let w_t be the deterministic sequence defined in equation (1) by taking $\hat{B} = B/4$, and $p^+_\delta(t) = a + bt^{-\alpha}$ for values $a \in [0,1]$, $b \geq 0$ and $\alpha > 0$, where a, b may depend on δ, B, T (but not t). Then, for $\alpha = 1$ and*

$$R^*_1 = \frac{8aT}{B} + \log_2(T),$$

it holds that $w_1^{R^*_1} \geq T$. Furthermore, for $\alpha = \frac{1}{2}$, and $b' = \max\{b, 1\}$:

$$R^*_{1/2} = \frac{1}{B}\Big(8aT + 32b'\sqrt{T} + 4b^2\Big) + \log\log(B/4) + 2,$$

it holds that $w_1^{R^*_{1/2}} \geq T$.

The theorem shows that BR^*_1 and $BR^*_{1/2}$ are, indeed, of the same order as standard active learning bounds. This implies that in the batch setting we pay only an additive $\tilde{O}(B)$ cost over the standard non-batch label complexity. Using this theorem in combination with Theorem 1, we attain the label complexity for several canonical algorithms in the following section.

4 Applications

We analyze the IWAL, ORIWAL, and DHM algorithms where for each algorithm, we show how to extend it to the batch setting via Algorithm 1. For these batch variants, we prove that their generalization guarantees are of the same order as the original non-batch versions and that the theorems of the previous section can be leveraged to bound the label complexity with only a modest dependence on the batch size. In the next subsection, we provide a full description of the batch variant of the IWAL algorithm, generalization guarantee, bound on time-decreasing labeling rate, and resulting label complexity bound. Due to space constraints, we only provide the label complexity for the batch variants of the ORIWAL, and DHM algorithms in the body of the paper, but still provide the full algorithm description and corresponding guarantees in the appendix.

4.1 The IWAL algorithm

We start by recalling the IWAL algorithm of [Beygelzimer et al., 2009]. At each time $t \in [T]$, the IWAL algorithm observes a single point x_t and to determine whether to request its label, the algorithm flips a coin $Q_t \in \{0, 1\}$ with bias $p_t = \mathbb{P}(Q_t = 1)$. If $Q_t = 1$, then the algorithm requests the label of the point x_t while, if $Q_t = 0$, it passes on this request. The bias probability is defined as $p_t = \max_{f,g \in H_t} \max_{y \in \mathcal{Y}} |\ell(f(x_t), y) - \ell(g(x_t), y)|$ where $H_t \subseteq H$ is the version space at time t maintained by the algorithm. At each time t, the algorithm reduces the version space by removing any hypothesis far from the empirical best-in-class: $H_t = \big\{h \in H_{t-1} \colon L_{t-1}(h) \leq \min_{h' \in H_{t-1}} L_{t-1}(h') + \Delta_{t-1}\big\}$, where $L_t(f) = \frac{1}{t}\sum_{s=1}^t \frac{Q_s}{p_s}\ell(f(x_s), \bar{y}_s)$ is the weighted empirical loss and Δ_t is a slack term.

Algorithm 2 `Labeler`(H_r, x_t) for B-IWAL

$p_r(x_t) \leftarrow \max_{f,g \in H_r} \max_{y \in \mathcal{Y}} |\ell(f(x_t), y) - \ell(g(x_t), y)|$

return $p_r(x_t)$

Algorithm 3 `Updater`$(H_r, \mathcal{Z}^r_\perp)$ for B-IWAL

$H_{r+1} \leftarrow \big\{h \in H_r \colon L_{\tau_r}(h) \leq \min_{h' \in H_r} L_{\tau_r}(h') + \Delta_{\tau_r}\big\}$

return H_{r+1}

The B-IWAL algorithm extends the IWAL algorithm to our setting by freezing the version space for the length-of-round. More concretely, by recalling Algorithm 1, the state for the B-IWAL algorithm is defined in terms of the version space, that is $\omega_r = H_r$ for round r and initially, we set $H_1 = H$. The `Labeler`, which returns the probability of requesting a point, and `Updater`, which updates the state, are defined in Pseudocode 2 and Pseudocode 3. Due to a technicality, the slack term used in the version space is defined as $\Delta_t = \sqrt{8\log(2T^2(T+1)|H|^2/\delta)/t}$, which deviates slightly from slack term of IWAL as it contains T^2 instead of T. The following theorem provides generalization guarantees for the B-IWAL algorithm.

Theorem 3 *Let $\widehat{h}_T$ denote the hypothesis returned by* B-IWAL *after T time steps and let $h^* = \operatorname{argmin}_{h \in H} L(h)$. For any $\delta > 0$, with probability at least $1 - \delta$, $L(\widehat{h}_T) \leq L(h^*) + O\Big(\sqrt{\frac{\log(T|H|/\delta)}{T}}\Big)$.*

The theorem states that the expected loss of the best-in-class h^* is close to that of the hypothesis returned by the algorithm. As T grows, the difference in the expected loss of these two hypotheses decreases at the typical rate of $O(1/\sqrt{T})$. Thus, despite the fact that the version space is updated less frequently, the generalization bound of the B-IWAL algorithm is of the same order as that of the IWAL algorithm. At a high level, this follows from the fact that, although the version space is updated less frequently, when it is updated it will still "catch up" to the analogous hypothesis class that is updated immediately after each time step. However, the algorithm may request more labels due to maintaining a frozen state.

The label complexity of the active learning algorithm will depend on the disagreement coefficient $\theta(\mathcal{D}_\mathcal{X}, H)$,

which is defined as the infimum value of $\theta > 0$ such that for all $\Lambda \geq 0$:

$$\mathop{\mathbb{E}}_{x\in\mathcal{D}_{\mathcal{X}}}\Big[\max_{h\in\mathcal{B}(h^*,\Lambda)}\max_{y\in\mathcal{Y}}|\ell(h(x),y)-\ell(h^*(x),y)|\Big]\leq\theta\Lambda\,,$$

where $\mathcal{B}(h',\Lambda)=\{h\in H\colon \rho(h,h')\leq\Lambda\}$ is the ball of radius $\Lambda\geq 0$ and $\rho(h,h')=\mathbb{E}[|\ell(h(x),y)-\ell(h'(x),y)|]$ is the distance between two functions in $h,h'\in H$. Note, this definition of ρ is taken from Cortes et al. [2019b] which allows for a tighter label complexity. For simplicity, we use θ instead of $\theta(\mathcal{D}_{\mathcal{X}},H)$ in this section. The next lemma bounds the probability of the B-IWAL algorithm requesting a point and thereby implies that the time-decreasing labeling rate property is satisfied.

Lemma 4 *For $\delta>0$, with probability at least $1-\delta$, at any round r, $\mathbb{E}_x[p_r(x)|\tau_{r-1}]\leq 4\theta(L(h^*)+\Delta_{\tau_{r-1}})$.*

Now, to attain the label complexity bound, we apply the general theory from Section 3. Lemma 4 implies an upper bound on the sampling probability of the form $p_\delta^+(t)=a+bt^{-\alpha}$ with $a=4\theta L(h^*)$, $b=4\theta\sqrt{8\log(2T^2(T+1)|H|^2/\delta)}$, and $\alpha=1/2$. We then apply Theorem 2 with $R^*_{1/2}$ and Theorem 1 along with simplifying terms to prove the following corollary.

Corollary 1 *Fix $\delta<\sqrt{1/e}$, time horizon $T\geq 1$, and batch size $B>\max\{16b,16,2\log(T)\}$. Then with probability at least $1-2\delta$, the total labels requested by* B-IWAL *is bounded by: $\widetilde{O}\big(\theta L(h^*)T+\theta\sqrt{T}+B\big)$, where $\widetilde{O}(\cdot)$ is hiding absolute constants, $\log(T|H|)$ and $\log(1/\delta)$.*

Notice that an additive $\Omega(B)$ term is necessary since the algorithm must request at least B points to see any labels. Thus, for practical label batch sizes, the bound is nearly optimal except for constants and log terms.

4.2 The ORIWAL algorithm

At a high level, the ORIWAL algorithm of Cortes et al. [2019a] works by partitioning the space into regions and running a separate active learning algorithm in each region while carefully allocating the labeling resources across regions. Specifically, in each region, the ORIWAL runs the algorithm EIWAL, which is an enhanced version of IWAL with stronger theoretical guarantees.

We first present some needed notation and recall the algorithm. We denote by $\mathcal{X}_k$ for $k\in[n]$ the regions that partition in the input $\mathcal{X}$ and by H_k the hypothesis used in each region. The ORIWAL algorithm returns a hypothesis from the following region-based hypothesis set: $H_{[n]}=\{\sum_{k=1}^n 1_{x\in\mathcal{X}_k}h_k(x)\colon h_k\in H_k\}$. We define $L_k^*=\min_{h\in H_k}\mathbb{E}[\ell(h(x),y)|x\in\mathcal{X}_k]$ be the regional best-in-class and $\theta_k=\theta(\mathcal{D}_{\mathcal{X}_k},H_k)$ to be the regional disagreement coefficient where $\mathcal{D}_{\mathcal{X}_k}$ is the conditional distribution of x given region k.

At each time $t\in[T]$, ORIWAL receives the points x_t, finds the region k_t it belongs to, and decides whether to pass this point to the sub-routine EIWAL in region k_t by flipping a coin $A_t\in\{0,1\}$ with bias α_{k_t}. This bias probability carefully chosen to minimize the label complexity across the regions: $\alpha_k=\frac{(c_k/\mathtt{p}_k)^{1/3}}{\max_{k\in[n]}(c_k/\mathtt{p}_k)^{1/3}}$ where $c_k=\log[\frac{16T^2|H_k|^2\log(T)n}{\delta}]$ and where $\mathtt{p}_k=\mathbb{P}[\mathcal{X}_k]$. If $A_t=1$, then the point x_t is passed to the EIWAL instance in region k_t, which decides whether to request the label y_t, and updates its internal state.

In the appendix, we present the pseudo-code used to convert ORIWAL to its batch version, B-ORIWAL, and show that it exhibits a time-decreasing labeling rate of the order $O(1/T)$ in the case that $L_k^*=0$ for all k and $O(1/\sqrt{T})$ otherwise. This results in the following label complexity guarantee for B-ORIWAL.

Corollary 2 *Fix $\delta<\sqrt{1/e}$, time horizon $T\geq 1$, and batch size $B>\max\{16b,16,2\log(T),\frac{4\log(2n/\delta)}{\min_{k\in[n]}\mathtt{q}_k}\}$. Then with probability at least $1-2\delta$, the total labels requested by* B-ORIWAL *is bounded as follows:*

when $L_k^*=0$ for all $k\in[n]$,

$$O\big(B\log_2(T)+B\log(1/\delta)+B\big)$$

and when $\exists k\in[n]$ such that $L_k^*>0$,

$$\widetilde{O}\big(\textstyle\sum_{k=1}^n\mathtt{q}_k\theta_kL_k^*T+\theta_k\sqrt{T}+B\big)\,,$$

where $\mathtt{q}_k=\mathtt{p}_k\alpha_k/\sum_{k'=1}^n\mathtt{p}_{k'}\alpha_{k'}$.

The above label complexity matches that of ORIWAL modulo additive $\tilde{O}(B)$ terms. Crucially, as in the case of B-IWAL, the generalization guarantee is unaffected by batching, which is proven in the appendix.

4.3 The DHM algorithm

We extend the DHM algorithm [Dasgupta et al., 2008] to the batch setting. Given a point x_t, the DHM algorithm decides to either request the label or assigns it a carefully chosen pseudolabel. Specifically, it constructs two sets, $\widehat{S}_t$ and T_t, such that $\widehat{S}_t$ contains examples with pseudolabels consistent with the best-in-class h^* and T_t contains examples with requested labels. The union of these two sets is thus an i.i.d. sample from the underlying marginal distribution on the input space. To decide whether pseudo-label or request the label for given a point x_t, the algorithm checks if the difference of the empirical error on $(\widehat{S}_{t-1},T_{t-1})$ of two hypothesis learned via $h_{\widehat{y}}=\mathrm{LEARN}_H(\widehat{S}_{t-1}\cup\{x_t,\widehat{y}\},T_{t-1})$ for $\widehat{y}\in\{\pm1\}$ is large enough. The $\mathrm{LEARN}_H(A,B)$ denotes a learning algorithm that either returns hypothesis $h\in H$ consistent with A and with minumum error on B.

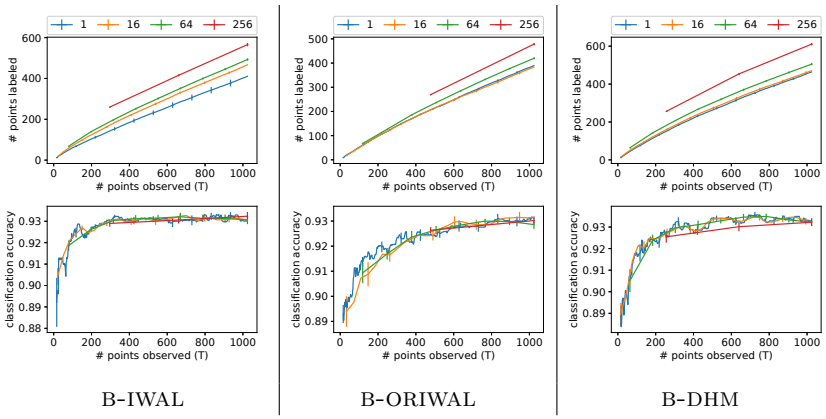

Figure 1: Each column displays the behavior of a different online active learning algorithm on the `phishing` dataset under various batch size constraints. A batch size of 1 (blue curve) corresponds to the setting where the active learner receives a label as soon as it is requested. The top row shows the mean number of points labeled by the respective algorithm, while the bottom row measures the mean test accuracy of the selected model, and error bars indicate the standard error.

The pseudocode for the batch version of DHM, called B-DHM, and the proof that this algorithm satisfies the time-decreasing labeling rate of $O(1/T)$ is in the appendix. Then, the following label complexity bound holds where θ' is given by Definition 2 in Dasgupta et al. [2008].

Corollary 3 *Fix $\delta < \sqrt{1/e}$, constant $c > 0$, time horizon $T \geq 1$, and batch size $B > \max\{16b, 16, 2\log(T)\}$. Then with probability at least $1 - 2\delta$, the total labels requested by* B-DHM *is bounded by: $O\big(c\theta' L(h^*)T + B\log_2(T) + B\log(1/\delta) + B\big)$.*

Once again, the label complexity of this batch algorithm admits, ignoring logarithmic terms, an additive linear dependence on the batch size B. In the appendix, we prove that the generalization guarantee of B-DHM is of the same order as DHM.

5 Empirical Verification

We empirically measure the effect of batching on online active learning algorithms, in the particular case of IWAL, ORIWAL, and DHM algorithms discussed in the previous section. We conduct the evaluation using six different publicly available benchmark datasets: `a9a`, `cod-rna`, `covtype`, `HIGGS`, `mnist`, and `phishing`.[3] For each dataset, the features are normalized to have zero mean and unit variance and subsequently scaled to ensure the maximum feature vector has unit norm. Furthermore, for each dataset, a finite hypothesis set of logistic regression models is generated to serve as the hypothesis set H. Each evaluation averages 10 trials, each with a random unlabeled pool and test set split. For details on the size of unlabeled pool and test fold, the number of features, the numbers of hypotheses, and hyperparameter settings used for each dataset, please refer to Appendix C. Details regarding plotting methodology are also found in the same appendix.

Figure 1 displays performance on the `phishing` dataset. Ignoring lower-order dependencies on T and logarithmic factors, Corollaries 1, 2 and 3 predict that the cumulative number of labels requested should appear as $aT + B$ for some problem-dependant constant a. Indeed, for each algorithm, the number of labels requested increases with at most an additive dependence on the label query batch size, as is suggested by the additive dependence found in their respective label complexity bounds. Furthermore, we also observe that the test accuracy is essentially unaffected by the batch size, as suggested by the corresponding generalization guarantees. Due to space constraints, results for the remainder of the datasets for all three algorithms, which show similar behavior, are presented in Appendix C. Overall, we find that the empirical results corroborate the theoretical results of the previous sections.

6 Conclusion

We presented an analysis of the batch online active learning setting, which is directly motivated by practical constraints. We bound the label complexity of a generic batch online active learning algorithm, showed that the result can be applied to several well-studied online active learning algorithms, and verified the findings empirically. Future directions include analyzing batching effects in pool-based settings. In such settings, additional requirements, such as enforcing diversity of examples within a batch, may be necessary.

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

References

Pranjal Awasthi, Maria-Florina Balcan, and Philip M. Long. The power of localization for efficiently learning linear separators with noise. In *Proceedings of the forty-sixth annual ACM symposium on theory of computing*, pages 449–458. ACM, 2014.

Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Ruth Urner. Efficient learning of linear separators under bounded noise. In *Conference on Learning Theory*, pages 167–190, 2015.

Francis Bach. Asymtotic analysis of generalized linear models. *Advances in Neural Information Processing Systems*, 2007.

Maria-Florina Balcan and Phil M. Long. Active and passive learning of linear separators under log-concave distributions. In *Conference on Learning Theory*, pages 288–316, 2013.

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In *Proceedings of the 23th International Conference on Machine Learning*, 2006.

Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *Conference on Learning Theory*, pages 35–50. Springer, 2007.

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *Proceedings of the 26th International Conference on Machine Learning*, pages 49–56. ACM, 2009.

Alina Beygelzimer, Daniel J. Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. In *Advances in Neural Information Processing Systems*, pages 199–207, 2010.

Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.

Yuxin Chen and Andreas Krause. Near-optimal batch mode active learning and adaptive submodular optimization. *International Conference on Machine Learning*, 2013.

David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Ningshan Zhang. Region-based active learning. In *International Conference on Artificial Intelligence and Statistics*, 2019a.

Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Ningshan Zhang. Active learning with disagreement graphs. In *Proceedings of the 26th International Conference on Machine Learning*, 2019b.

Sanjoy Dasgupta and Daniel J. Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 208–215. ACM, 2008.

Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. In *Conference on Learning Theory*, pages 249–263. Springer Berlin Heidelberg, 2005.

Sanjoy Dasgupta, Daniel J. Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems*, pages 353–360, 2008.

Yuhong Guo and Dale Schuurmans. Discriminative batch mode active learning. In *Advances in Neural Information Processing Systems*, 2008.

Steve Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th International Conference on Machine learning*, pages 353–360. ACM, 2007.

Steven Hoi, Rong Jin, and Michael Lyu. Large-scale text categorization by batch mode active learning. In *International Conference on World Wide Web*, 2006a.

Steven Hoi, Rong Jin, Jianke Zhu, and Michael Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006b.

Steven Hoi, Rong Jin, Jianke Zhu, and Michael Lyu. Semi-supervised svm batch mode active learning for image retrieval. In *Computer Vision and Pattern Recognition*, 2008.

Svante Janson. Tail bounds for sums of geometric and exponential variables. *Statistics & Probability Letters*, 135:1–6, 2018.

Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International Conference on Machine Learning*, pages 1453–1461, 2013.

Nozomi Kurihara and Masashi Sugiyama. Improving importance estimation in pool-based batch active learning for approximate linear regression. *Neural Networks*, 36:73–82, 2012.

Andrew McCallum and Kamal Nigam. Employing em in pool-based active learning for text classification. *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python.

Journal of Machine Learning Research, 12:2825–2830, 2011.

Zuobing Xu, Ram Akella, and Yi Zhang. Incorporating diversity and density in active learning for relevance feedback. In *European Conference on Information Retrieval*, 2007.

Chicheng Zhang. Efficient active learning of sparse halfspaces. In *Conference on Learning Theory*, 2018.