Efficient Active Learning

Alina Beygelzimer BEYGEL@US.IBM.COM
IBM Research, NY

Daniel Hsu DJHSU@RCI.RUTGERS.EDU
Rutgers University, NJ and University of Pennsylvania, PA

Nikos Karampatziakis NK@CS.CORNELL.EDU
Cornell University, NY

John Langford JL@YAHOO-INC.COM
Yahoo! Research, NY

Tong Zhang TZHANG@STAT.RUTGERS.EDU
Rutgers University, NJ

Abstract

We present and analyze an active learning algorithm that is theoretically sound in an agnostic setting, empirically effective, and as efficient as standard online learning algorithms. This allows us to soundly and effectively optimize the explore/exploit tradeoff in active learning at a scale of 10^6 examples/second.

The present work is primarily based on (Beygelzimer et al., 2010) and (Karampatziakis & Langford, 2011).

1. Introduction

In active learning, a learner is given access to unlabeled data and is allowed to adaptively choose which ones to label. This learning model is motivated by applications in which the cost of labeling data is high relative to that of collecting the unlabeled data itself. Therefore, the hope is that the active learner only needs to query the labels of a small number of the unlabeled data, and otherwise perform as well as a fully supervised learner. In this work, we are interested in agnostic active learning algorithms for binary classification that are provably consistent, *i.e.* that converge to an optimal hypothesis in a given hypothesis class.

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 only if there is disagreement within this set about how to label the point. The criteria for membership in this candidate set needs to be carefully defined so that an optimal hypothesis is always included, but otherwise this set can be quickly whittled down as more labels are queried. This technique is perhaps most readily understood in the noise-free setting (Cohn et al., 1994; Dasgupta, 2005), and it can be extended to noisy settings by using confidence bounds (Balcan et al., 2006; Dasgupta et al., 2007; Beygelzimer et al., 2009; Hanneke, 2009; Koltchinskii, 2010).

The version space approach unfortunately has its share of significant drawbacks. The first is computational intractability: maintaining a version space and guaranteeing that *only* hypotheses from this set are returned is difficult for linear predictors and appears intractable for interesting nonlinear predictors such as neural nets and decision trees (Cohn et al., 1994). Another drawback of the approach is its brittleness: a single mishap (due to, say, modeling failures or computational approximations) might cause the learner to exclude the best hypothesis from the version space forever; this is an ungraceful failure mode that is not easy to correct. A third drawback is related to sample re-usability: if (labeled) data is collected using a version space-based active learning algorithm, and we later decide to use a different algorithm or hypothesis class, then the earlier data may not be freely re-used because its collection

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

process is inherently biased.

Here, we develop a new strategy addressing all of the above problems given an oracle that returns an empirical risk minimizing (ERM) hypothesis. As this oracle matches our abstraction of many supervised learning algorithms, we believe active learning algorithms built in this way are immediately and widely applicable.

Our approach instantiates the importance weighted active learning framework of (Beygelzimer et al., 2009) using a rejection threshold similar to the algorithm of (Dasgupta et al., 2007) which only accesses hypotheses via a supervised learning oracle. However, the oracle we require is simpler and avoids strict adherence to a candidate set of hypotheses. Moreover, our algorithm creates an importance weighted sample that allows for unbiased risk estimation, even for hypotheses from a class different from the one employed by the active learner. This is in sharp contrast to many previous algorithms (*e.g.*, (Cohn et al., 1994; Balcan et al., 2006; 2007; Dasgupta et al., 2007; Hanneke, 2009; Koltchinskii, 2010)) that create heavily biased data sets. We prove that our algorithm is always consistent and has an improved label complexity over passive learning in cases previously studied in the literature.

We also describe two practical instantiations of our algorithm, where the required ERM oracle is approximated using efficient supervised learners, and report on some experimental results. The first is based on the decision tree learning procedure J48 from Weka v3.6.2 (Hall et al., 2009). The second is based on the online learning software Vowpal Wabbit (VW) (Langford et al., 2007) using the importance weight-aware updates from (Karampatziakis & Langford, 2011). The specialized updates are essential for achieving good performance when using online learning algorithms like VW in the context of importance weighted active learning. This online active learning algorithm runs at the same speed as simple online learning, implying that the explore/exploit tradeoff in active learning can be effectively optimized at rates of 10^6 examples/second.

2. Preliminaries

2.1. Learning Model

Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{X}$ is the input space and $\mathcal{Y} = \{\pm 1\}$ are the labels. Let $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ be a pair of random variables with joint distribution $\mathcal{D}$. An active learner receives a sequence $(X_1, Y_1), (X_2, Y_2), \ldots$ of i.i.d. copies of (X, Y), with the label Y_i hidden unless it is explicitly queried. We use the shorthand $a_{1:k}$ to denote a sequence $(a_1, a_2, \ldots, a_k)$ (so $k = 0$ correspond to the empty sequence).

Let $\mathcal{H}$ be a set of hypotheses mapping from $\mathcal{X}$ to $\mathcal{Y}$. For simplicity, we assume $\mathcal{H}$ is finite but does not completely agree on any single $x \in \mathcal{X}$ (*i.e.*, $\forall x \in \mathcal{X}, \exists h, h' \in \mathcal{H}$ such that $h(x) \neq h'(x)$). This keeps the focus on the relevant aspects of active learning that differ from passive learning. The error of a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ is $\mathrm{err}(h) := \Pr(h(X) \neq Y)$. Let $h^* := \arg\min\{\mathrm{err}(h) : h \in \mathcal{H}\}$ be a hypothesis of minimum error in $\mathcal{H}$. The goal of the active learner is to return a hypothesis $h \in \mathcal{H}$ with error $\mathrm{err}(h)$ not much more than $\mathrm{err}(h^*)$, using as few label queries as possible.

2.2. Importance Weighted Active Learning

In the importance weighted active learning (IWAL) framework of (Beygelzimer et al., 2009), an active learner looks at the unlabeled data $X_1, X_2, \ldots$ one at a time. After each new point X_i, the learner determines a probability $P_i \in [0, 1]$. Then a coin with bias P_i is flipped, and the label Y_i is queried if and only if the coin comes up heads. The query probability P_i can depend on all previous unlabeled examples $X_{1:i-1}$, any previously queried labels, any past coin flips, and the current unlabeled point X_i.

Formally, an IWAL algorithm specifies a *rejection threshold* function $p : (\mathcal{X} \times \mathcal{Y} \times \{0, 1\})^* \times \mathcal{X} \to [0, 1]$ for determining these query probabilities. Let $Q_i \in \{0, 1\}$ be a random variable conditionally independent of the current label Y_i,

$$Q_i \perp\!\!\!\perp Y_i \mid X_{1:i}, Y_{1:i-1}, Q_{1:i-1}$$

and with conditional expectation

$$\mathbb{E}[Q_i | Z_{1:i-1}, X_i] \;=\; P_i \;:=\; p(Z_{1:i-1}, X_i).$$

where $Z_j := (X_j, Y_j, Q_j)$. That is, Q_i indicates if the label Y_i is queried (the outcome of the coin toss). Although the notation does not explicitly suggest this, the query probability $P_i = p(Z_{1:i-1}, X_i)$ is allowed to explicitly depend on a label Y_j ($j < i$) if and only if it has been queried ($Q_j = 1$).

2.3. Importance Weighted Estimators

We first review some standard facts about the importance weighting technique. For a function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, define the *importance weighted estimator* of $\mathbb{E}[f(X, Y)]$ from $Z_{1:n} \in (\mathcal{X} \times \mathcal{Y} \times \{0, 1\})^n$ to be

$$\widehat{f}(Z_{1:n}) \;:=\; \frac{1}{n} \sum_{i=1}^{n} \frac{Q_i}{P_i} \cdot f(X_i, Y_i).$$

Note that this quantity depends on a label Y_i only if it has been queried (*i.e.*, only if $Q_i = 1$; it also depends on X_i only if $Q_i = 1$). Our rejection threshold will be based on a specialization of this estimator, specifically the *importance weighted empirical error* of a hypothesis h

$$\mathrm{err}(h, Z_{1:n}) \;:=\; \frac{1}{n}\sum_{i=1}^{n}\frac{Q_i}{P_i}\cdot \mathbb{1}[h(X_i) \neq Y_i].$$

In the notation of Algorithm 1, this is equivalent to

$$\mathrm{err}(h, S_n) := \frac{1}{n}\sum_{(X_i,Y_i,1/P_i)\in S_n}\frac{1}{P_i}\cdot \mathbb{1}[h(X_i) \neq Y_i] \quad (1)$$

where $S_n \subseteq \mathcal{X}\times\mathcal{Y}\times\mathbb{R}$ is the importance weighted sample collected by the algorithm.

A basic property of these estimators is *unbiasedness*: $\mathbb{E}[\widehat{f}(Z_{1:n})] = (1/n)\sum_{i=1}^{n}\mathbb{E}[\mathbb{E}[(Q_i/P_i)\cdot f(X_i,Y_i) \mid X_{1:i}, Y_{1:i}, Q_{1:i-1}]] = (1/n)\sum_{i=1}^{n}\mathbb{E}[(P_i/P_i)\cdot f(X_i,Y_i)] = \mathbb{E}[f(X,Y)]$. So, for example, the importance weighted empirical error of a hypothesis h is an unbiased estimator of its true error $\mathrm{err}(h)$. This holds for *any* choice of the rejection threshold that guarantees $P_i > 0$.

3. Algorithm

First, we state a deviation bound for the importance weighted error of hypotheses in a finite hypothesis class $\mathcal{H}$ that holds for all $n \geq 1$. The form of the bound motivates certain algorithmic choices to be described below.

Lemma 1. *Pick any $\delta \in (0,1)$. For all $n \geq 1$, let*

$$\varepsilon_n := \frac{16\log(2(3+n\log_2 n)n(n+1)|\mathcal{H}|/\delta)}{n} = O\left(\frac{\log(n|\mathcal{H}|/\delta)}{n}\right). \quad (3)$$

Let $(Z_1, Z_2, \ldots) \in (\mathcal{X}\times\mathcal{Y}\times\{0,1\})^*$ be the sequence of random variables specified in Section 2.2 using a rejection threshold $p : (\mathcal{X}\times\mathcal{Y}\times\{0,1\})^*\times\mathcal{X} \to [0,1]$ that satisfies $p(z_{1:n}, x) \geq 1/n^n$ for all $(z_{1:n}, x) \in (\mathcal{X}\times\mathcal{Y}\times\{0,1\})^n\times\mathcal{X}$ and all $n \geq 1$.

The following holds with probability at least $1-\delta$. For all $n \geq 1$ and all $h \in \mathcal{H}$,

$$|(\mathrm{err}(h, Z_{1:n}) - \mathrm{err}(h^*, Z_{1:n})) - (\mathrm{err}(h) - \mathrm{err}(h^*))| \leq \sqrt{\frac{\varepsilon_n}{P_{min,n}(h)}} + \frac{\varepsilon_n}{P_{min,n}(h)} \quad (4)$$

where $P_{min,n}(h) = \min\{P_i : 1 \leq i \leq n \;\wedge\; h(X_i) \neq h^*(X_i)\} \cup \{1\}$.

```
Algorithm 1
Note: see Eq. (1) for the definition of err (importance weighted error), and Section 3 for the definitions of $C_0$, $c_1$, and $c_2$.
Initialize: $S_0 := \emptyset$.
For $k = 1, 2, \ldots, n$:
    1. Obtain unlabeled data point $X_k$.
    2. Let
          $h_k := \arg\min\{\mathrm{err}(h, S_{k-1}) : h \in \mathcal{H}\}$, and
          $h'_k := \arg\min\{\mathrm{err}(h, S_{k-1}) : h \in \mathcal{H} \wedge h(X_k) \neq h_k(X_k)\}$.
       Let $G_k := \mathrm{err}(h'_k, S_{k-1}) - \mathrm{err}(h_k, S_{k-1})$, and
          $P_k := \begin{cases} 1 & \text{if } G_k \leq \sqrt{\frac{C_0 \log k}{k-1}} + \frac{C_0 \log k}{k-1} \\ s & \text{otherwise} \end{cases}$
       where $s \in (0,1)$ is the positive solution to the equation
          $G_k = \left(\frac{c_1}{\sqrt{s}} - c_1 + 1\right)\cdot\sqrt{\frac{C_0\log k}{k-1}} + \left(\frac{c_2}{s} - c_2 + 1\right)\cdot\frac{C_0\log k}{k-1}.$   (2)
    3. Toss a biased coin with $\Pr(\mathrm{heads}) = P_k$.
          If heads, then query $Y_k$, and let $S_k := S_{k-1} \cup \{(X_k, Y_k, 1/P_k)\}$.
          Else, let $S_k := S_{k-1}$.
Return: $h_{n+1} := \arg\min\{\mathrm{err}(h, S_n) : h \in \mathcal{H}\}$.
```

Figure 1. Algorithm for importance weighted active learning with an error minimization oracle.

We let $C_0 = O(\log(|\mathcal{H}|/\delta)) \geq 2$ be a quantity such that ε_n (as defined in Eq. (3)) is bounded as $\varepsilon_n \leq C_0 \cdot \log(n+1)/n$. The absolute constants $c_1 := 5+2\sqrt{2}$ and $c_2 := 5$ are used in the description of the rejection threshold and its analysis.

Our proposed algorithm is shown in Figure 1. The rejection threshold (Step 2) is based on the deviation bound from Lemma 1. First, the importance weighted error minimizing hypothesis h_k and the "alternative" hypothesis h'_k are found. Note that both optimizations are over the entire hypothesis class $\mathcal{H}$ (with h'_k only being required to disagree with h_k on x_k)—this is a key aspect where our algorithm differs from previous approaches. The difference in importance weighted errors G_k of the two hypotheses is then computed. If $G_k \leq \sqrt{(C_0 \log k)/(k-1)} + (C_0 \log k)/(k-1)$, then the query probability P_k is set to 1. Otherwise, P_k is set to the positive solution s to the quadratic equation in Eq. (2). The functional form of P_k is roughly

$$\min\left\{1,\ O\left(\frac{1}{G_k^2} + \frac{1}{G_k}\right) \cdot \frac{C_0 \log k}{k-1}\right\}.$$

It can be checked that $P_k \in (0,1]$ and that P_k is non-increasing with G_k. It is also useful to note that $(\log k)/(k-1)$ is monotonically decreasing with $k \geq 1$ (we use the convention $\log(1)/0 = \infty$).

In order to apply Lemma 1 with our rejection threshold, we need to establish the (very crude) bound $P_k \geq 1/k^k$ for all k.

Lemma 2. *The rejection threshold of Algorithm 1 satisfies $p(z_{1:n-1}, x) \geq 1/n^n$ for all $n \geq 1$ and all $(z_{1:n-1}, x) \in (\mathcal{X} \times \mathcal{Y} \times \{0,1\})^{n-1} \times \mathcal{X}$.*

Note that this is a worst-case bound; our analysis shows that the probabilities P_k are more like $1/\mathrm{poly}(k)$ in the typical case.

4. Analysis

4.1. Consistency

We first prove a consistency guarantee for Algorithm 1 that bounds the generalization error of the importance weighted empirical error minimizer. The proof actually establishes a lower bound on the query probabilities $P_i \geq 1/2$ for X_i such that $h_n(X_i) \neq h^*(X_i)$. This offers an intuitive characterization of the weighting landscape induced by the importance weights $1/P_i$.

Theorem 1. *The following holds with probability at least $1-\delta$. For any $n \geq 1$,*

$$\mathrm{err}(h_n) \ \leq\ \mathrm{err}(h^*) + \sqrt{\frac{2C_0 \log n}{n-1}} + \frac{2C_0 \log n}{n-1}.$$

Therefore, the final hypothesis returned by Algorithm 1 after seeing n unlabeled data has roughly the same error bound as a hypothesis returned by a standard passive learner with n labeled data.

4.2. Label Complexity Analysis

We now bound the number of labels requested by Algorithm 1 after n iterations. We do so by bounding the probability of querying the label Y_n, which in turn gives a bound on the expected number of labels queried. The key to the proof is in relating empirical error differences and their deviations to the probability of querying a label. This is mediated through the *disagreement coefficient*, a quantity first used by (Hanneke, 2007) for analyzing the label complexity of the A^2 algorithm of (Balcan et al., 2006). The disagreement coefficient $\theta := \theta(h^*, \mathcal{H}, \mathcal{D})$ is defined as

$$\theta(h^*, \mathcal{H}, \mathcal{D}) := \sup\left\{\frac{\Pr(X \in \mathrm{DIS}(h^*, r))}{r} : r > 0\right\}$$

where

$$\begin{aligned}\mathrm{DIS}(h^*, r) := \{&x \in \mathcal{X} : \exists h' \in \mathcal{H} \text{ such that} \\ &\Pr(h^*(X) \neq h'(X)) \leq r \text{ and } h^*(x) \neq h'(x)\}\end{aligned}$$

(the disagreement region around h^* at radius r). This quantity is bounded for many learning problems studied in the literature; see (Hanneke, 2007; 2009; Friedman, 2009; Wang, 2009) for more discussion. Note that the supremum can instead be taken over $r > \epsilon$ if the target excess error is ϵ, which allows for a more detailed analysis.

Theorem 2. *With probability at least $1-\delta$, the expected number of labels queried by Algorithm 1 after n iterations is at most*

$$1+\theta \cdot 2\,\mathrm{err}(h^*) \cdot (n-1) + O\left(\theta \cdot \sqrt{C_0 n \log n} + \theta \cdot C_0 \log^3 n\right).$$

The bound is dominated by a linear term scaled by $\mathrm{err}(h^*)$, plus a sublinear term. The linear term $\mathrm{err}(h^*) \cdot n$ is unavoidable in the worst case, as evident from label complexity lower bounds (Kääriäinen, 2006; Beygelzimer et al., 2009). When $\mathrm{err}(h^*)$ is negligible (*e.g.*, the data is separable) and θ is bounded (as is the case for many problems studied in the literature (Hanneke, 2007)), then the bound represents a polynomial label complexity improvement over supervised learning, similar to that achieved by the version space algorithm from (Beygelzimer et al., 2009).

5. Experiments

Although agnostic learning is typically intractable in the worst case, empirical risk minimization can serve as a useful abstraction for many practical supervised learning algorithms in non-worst case scenarios. With this in mind, we experimentally evaluated two practical instantiations of Algorithm 1.

5.1. Decision tree learning experiments

Our first instantiation uses a popular algorithm for learning decision trees in place of the required ERM oracle. Specifically, we use the J48 algorithm from Weka v3.6.2 (Hall et al., 2009) (with default parameters) to select the hypothesis h_k in each round k; to produce the "alternative" hypothesis h'_k, we just modify the decision tree h_k by changing the label of the node used for predicting on x_k. Both of these procedures are clearly heuristic, but they are similar in spirit to the required optimizations. We set $C_0 = 8$ and $c_1 = c_2 = 1$—these can be regarded as tuning parameters, with C_0 controlling the aggressiveness of the rejection threshold. We did not perform parameter tuning with active learning although the importance weighting approach developed here could potentially be used for that.

5.1.1. Data Sets

We constructed two binary classification tasks using MNIST and KDDCUP99 data sets. For MNIST, we randomly chose 4000 training 3s and 5s for training (using the 3s as the positive class), and used all of the 1902 testing 3s and 5s for testing. For KDDCUP99, we randomly chose 5000 examples for training, and another 5000 for testing. In both cases, we reduced the dimension of the data to 25 using PCA.

To demonstrate the versatility of our algorithm, we also conducted a multi-class classification experiment using the entire MNIST data set (all ten digits, so 60000 training data and 10000 testing data). This required modifying how h'_k is selected: we force $h'_k(x_k) \neq h_k(x_k)$ by changing the label of the prediction node for x_k to the next best label. We used PCA to reduce the dimension to 40.

5.1.2. Results

We examined the test error as a function of (i) the number of unlabeled data seen, and (ii) the number of labels queried. We compared the performance of the active learner described above to a passive learner (one that queries every label, so (i) and (ii) are the same) using J48 with default parameters.

In all three cases, the test errors as a function of the number of unlabeled data were roughly the same for both the active and passive learners. This agrees with

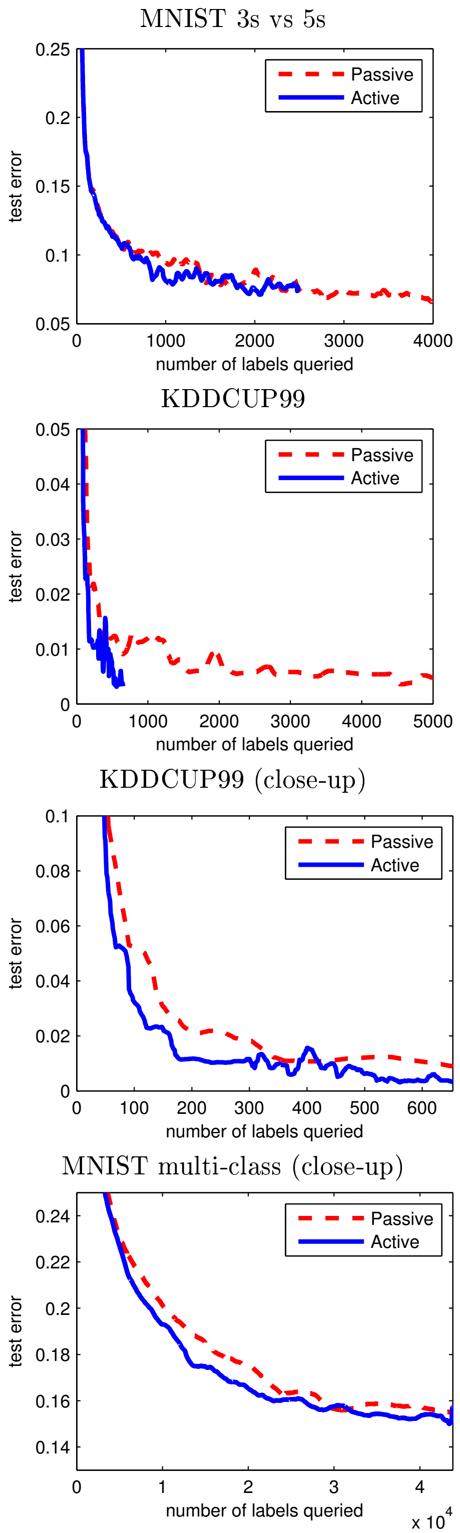

Figure 2. Test errors as a function of the number of labels queried for decision tree learning experiments.

the consistency guarantee from Theorem 1. We note that this is a basic property *not* satisfied by many active learning algorithms (this issue is discussed further in (Dasgupta & Hsu, 2008)).

In terms of test error as a function of the number of labels queried (Figure 2), the active learner had minimal improvement over the passive learner on the binary MNIST task, but a substantial improvement over the passive learner on the KDDCUP99 task (even at small numbers of label queries). For the multi-class MNIST task, the active learner had a moderate improvement over the passive learner. Note that KDDCUP99 is far less noisy (more separable) than MNIST 3s vs 5s task, so the results are in line with the label complexity behavior suggested by Theorem 2, which states that the label complexity improvement may scale with the error of the optimal hypothesis. Also, the results from MNIST tasks suggest that the active learner may require an initial random sampling phase during which it is equivalent to the passive learner, and the advantage manifests itself after this phase. This again is consistent with the analysis (also see (Hanneke, 2007)), as the disagreement coefficient can be large at initial scales, yet much smaller as the number of (unlabeled) data increases and the scale becomes finer.

5.2. Online learning experiments

Our second instantiation combines Algorithm 1 with an online gradient descent algorithm for learning linear predictors, as implemented in Vowpal Wabbit (VW) (Langford et al., 2007). In each iteration, the algorithm computes G_k based on the current weight vector $w_k \in \mathbb{R}^d$ and the new unlabeled data point $x_k \in \mathbb{R}^d$. This determines the query probability P_k, and the weight vector w_k is updated to w_{k+1} if the label is queried (and otherwise $w_{k+1} := w_k$).

Because the importance weights $1/P_k$ may be large, naïve approaches for dealing with importance weights can completely break down. For example, an update that simply multiplies the gradient of the loss with the importance weight and subtract it from the weight vector would create unecessarily large updates that shift the weights far beyond what is necessary to achieve a small loss. Instead, we use updates that take into account the curvature of the loss $\ell(\hat{y}, y)$ and are directly motivated from minimizing importance weighted losses.

In our experiments, we use two such updates. The first are implicit updates which update the weight vector as

$$w_{k+1} := \text{argmin}\, \frac{1}{2}||w - w_k||^2 + \eta_k i_k \ell(w^\top x_k, y_k)$$

where η_k and i_k are, respectively, the learning rate and importance weight at time k. The second are importance invariant updates (Karampatziakis & Langford, 2011):

$$w_{k+1} := w_k - s(w_k^\top x_k, i_k) x_k$$

where $s(p, i)$ is a scaling function that comes from the solution of this ODE

$$\frac{\partial s}{\partial i} = \eta_k \left.\frac{\partial \ell(p, y_k)}{\partial p}\right|_{p = w_k^\top x_k - s(w_k^\top x_k, i)||x_k||^2} \qquad (5)$$

$$s(p, 0) = 0$$

where $\ell(p, y)$ is the loss for predicting p when the label is y. It has a closed form solution for many ordinary loss functions such as squared loss and logistic loss, and coincides with implicit updates for piecewise linear losses such as hinge loss. Importantly, it satisfies an invariance property: updating twice with importance weight i has the same effect as updating once with weight $2i$ (*i.e.*, $s(p, 2i) = s(p, i) + s(p - s(p, i), i)$).

To get a handle on G_k we estimate $k \cdot G_k$ by the importance weight that the example would need to have in order for an update with the alternative hypothesis's prefered label to cause the classification of the example to become the alternative label. Specifically, for binary problems with $\mathcal{Y} = \{-1, 1\}$, let $\hat{y}_k = \text{sign}(w_k^\top x_k)$ and hence the alternative label is $y_a = -\hat{y}_k$. In the case of importance invariant updates, we want an importance weight i such that $(w_k - s(w_k^\top x_k, i) x_k)^\top x_k = 0$ where $s(w_k^\top x_k, i)$ is computed using y_a in place of y_k. Since s satisfies (5), by separating variables, integrating both sides and making use of the initial condition, we get that

$$i = \frac{1}{\eta_k} \int_0^{\frac{w_k^\top x_k}{||x_k||^2}} \frac{dt}{\left.\frac{\partial \ell(p, y_a)}{\partial p}\right|_{p = w_k^\top x_k - t||x_k||^2}}. \qquad (6)$$

Implicit updates also yield simple closed form solutions for the required importance weights. As these estimates and the updates for minimizing an importance weighted loss have simple forms, we obtain a very fast active learning algorithm.

5.2.1. Data Sets

We present empirical results on four text classification datasets: 'rcv1' is a modified version (Langford et al., 2007) of RCV1 (Lewis et al., 2004), 'astro' is from (Joachims, 2006), 'spam' was created from the TREC 2005 spam public corpora, and 'webspam' is from the PASCAL large scale learning challenge. In all experiments, we did a single pass through the training set and optimized squared loss. We report

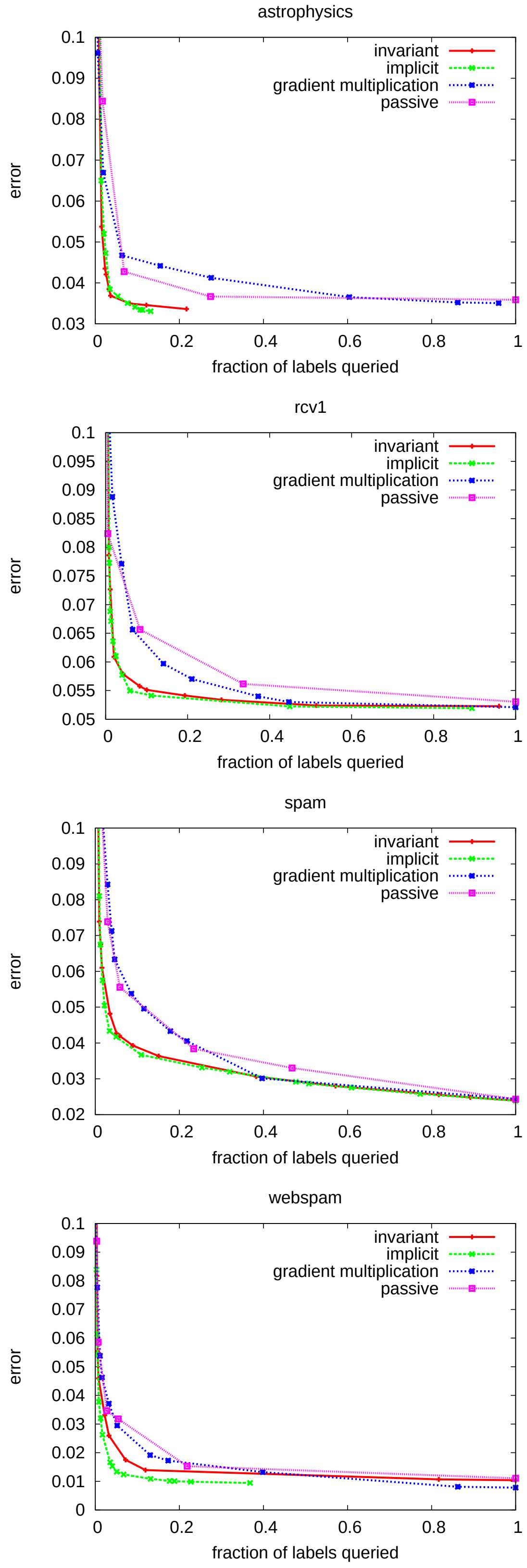

Figure 3. Test errors as a function of the number of labels queried for online learning experiments.

the error on the test set. We search over learning rates of the form $\eta_k = \frac{\mu}{\|x_k\|^2}\left(\frac{\kappa}{k+\kappa}\right)^p$ with $(\mu, \kappa, p) \in \{2^i\}_{i=0}^{10} \times \{10^i\}_{i=0}^{8} \times \{0.5, 1\}$. Since learning rates decay, (6) implies that the importance weights will grow as $\sim \eta_k^{-1}$, so between $\Omega(\sqrt{k})$ and $O(k)$.

5.2.2. Results

In Figure 3 we summarize our results. Each combination of learning rate schedule and setting of the parameter C_0 in Algorithm 1 ($C_0 \in \{10^{-8}, 10^{-7}, \ldots, 10^{1}\}$) is an experiment that can be represented in the graph by a point whose x-coordinate is the fraction of labels queried by the active learning algorithm and whose y-coordinate is the test error of the learned hypothesis. To summarize this set of points, the figures plot part of its convex hull. The points on the convex hull (sometimes called a Pareto frontier) are experiments which represent optimal tradeoffs between generalization and label complexity, for some setting of this tradeoff. When a curve stops sooner than the size of the dataset it means that there were no experiments in which using more queries gave better generalization. We have also included the results from a typical good run of a passive learner. The graphs show very convincingly the value of having an update that handles importance weights correctly. Doing so yields better generalization and lower label complexity, than those attainable by multiplying the gradient with the importance weight. In fact, linearization of the loss can make active learning need more labels than passive learning.

6. Conclusion

This paper provides a new active learning algorithm based on error minimization oracles, a departure from the version space approach adopted by previous works. The algorithm we introduce here motivates computationally tractable and effective methods for active learning with many classifier training algorithms. The overall algorithmic template applies to any training algorithm that (i) operates by approximate error minimization and (ii) for which the cost of switching a class prediction (as measured by example errors) can be estimated. Indeed, we have demonstrated the empirical effectiveness of two instantiations of the template using decision trees and linear predictors. Even when (i) and (ii) only hold in an approximate or heuristic sense, the created active learning algorithm will be "safe" in the sense that it will eventually converge to the same solution as a passive supervised learning algorithm. Consequently, we believe this approach can be widely used to reduce the cost of labeling in situations where labeling is expensive.

References

Balcan, M.-F., Beygelzimer, A., and Langford, J. Agnostic active learning. In *Twenty-Third International Conference on Machine Learning*, 2006.

Balcan, M.-F., Broder, A., and Zhang, T. Margin based active learning. In *Twentieth Annual Conference on Learning Theory*, 2007.

Beygelzimer, A., Dasgupta, S., and Langford, J. Importance weighted active learning. In *Twenty-Sixth International Conference on Machine Learning*, 2009.

Beygelzimer, A., Hsu, D., Langford, J., and Zhang, T. Agnostic active learning without constraints. In *Advances in Neural Information Processing Systems 23*, 2010.

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

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

Dasgupta, S. and Hsu, D. Hierarchical sampling for active learning. In *Twenty-Fifth International Conference on Machine Learning*, 2008.

Dasgupta, S., Hsu, D., and Monteleoni, C. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems 20*, 2007.

Friedman, E. Active learning for smooth problems. In *Twenty-Second Annual Conference on Learning Theory*, 2009.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The WEKA data mining software: An update. *SIGKDD Explorations*, 11 (1):10–18, 2009.

Hanneke, S. A bound on the label complexity of agnostic active learning. In *Twenty-Fourth International Conference on Machine Learning*, 2007.

Hanneke, S. Adaptive rates of convergence in active learning. In *Twenty-Second Annual Conference on Learning Theory*, 2009.

Joachims, T. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006.

Kääriäinen, M. Active learning in the non-realizable case. In *Seventeenth International Conference on Algorithmic Learning Theory*, 2006.

Karampatziakis, N. and Langford, J. Online importance weight aware updates, 2011. arXiv:1011.1576v3.

Koltchinskii, V. Rademacher complexities and bounding the excess risk in active learning. *Journal of Machine Learning Research*, 11:2457–2485, 2010.

Langford, J., Li, L., and Strehl, A. Vowpal Wabbit online learning project, 2007. `http://hunch.net/?p=309`.

Lewis, D.D., Yang, Y., Rose, T.G., and Li, F. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

Wang, L. Sufficient conditions for agnostic active learnable. In *Advances in Neural Information Processing Systems 22*, 2009.

Zhang, T. Data dependent concentration bounds for sequential prediction algorithms. In *Eighteenth Annual Conference on Learning Theory*, 2005.